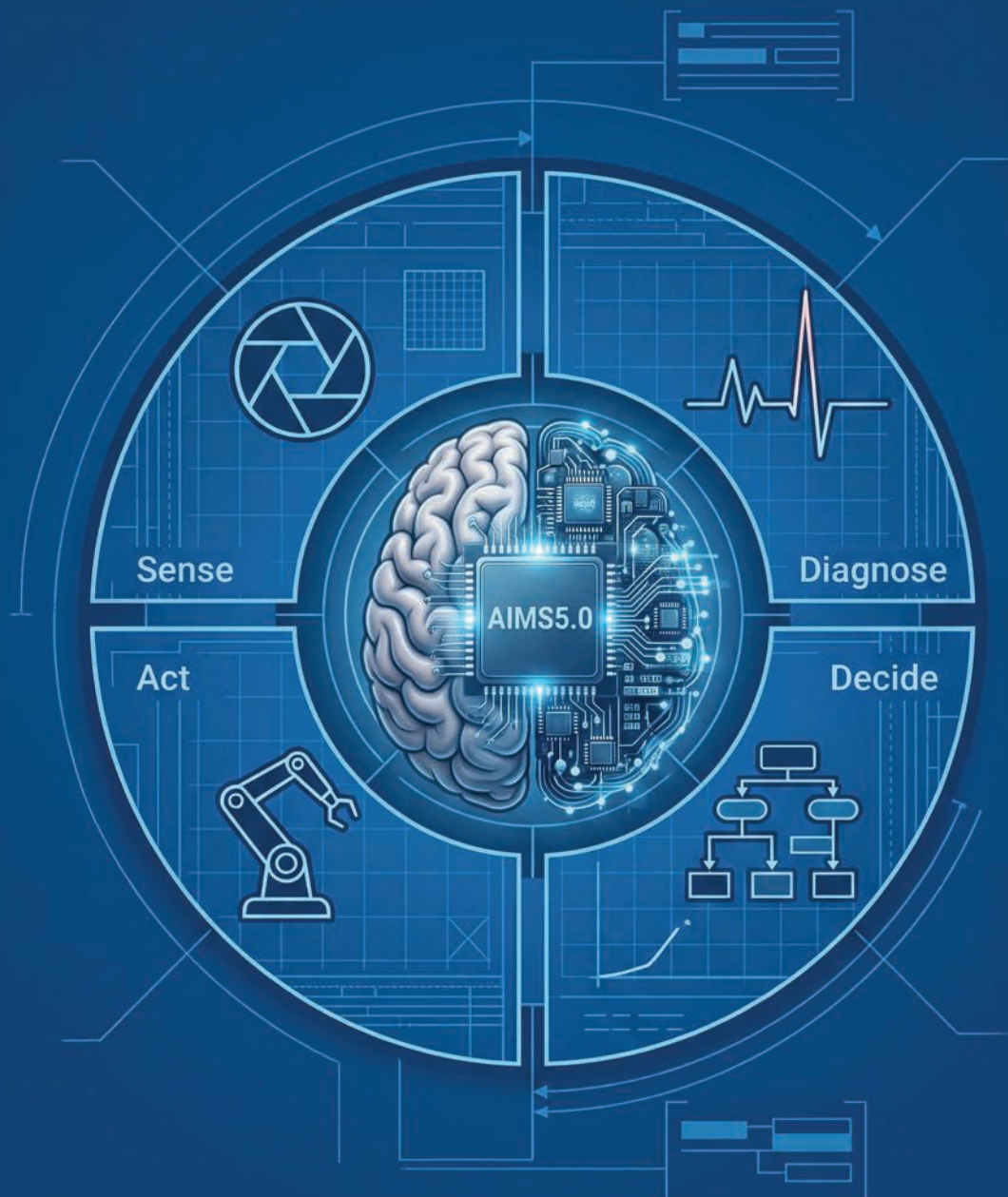




Industrial AI *for* Industry 5.0:

*Methods, Deployments and Lessons
from the AIMS5.0 Project*

EDITORS:

PÁL VARGA • KĀRLIS FREIVALDS
GIAN ANTONIO SUSTO • PÉTER OLASZI

SPECIAL ISSUE OF THE INFOCOMMUNICATION JOURNAL

Industrial AI *for* Industry 5.0:

Methods, Deployments and Lessons from the AIMS5.0 Project

EDITORS:

PÁL VARGA • KĀRLIS FREIVALDS
GIAN ANTONIO SUSTO • PÉTER OLASZI

The research leading to these results was conducted in the AIMS5.0 project, which received funding from the Chips Joint Undertaking under grant agreement No. 101112089 and from the partners' national funding programmes.

2026

SPECIAL ISSUE OF THE INFOCOMMUNICATION JOURNAL

Introduction: Industrial AI for Industry 5.0

Pál Varga, Kārlis Freivalds, Gian Antonio Susto, and Péter Olszsi

I. AIMS5.0 – PROVING GROUND FOR ENGINEERED AI

Industry 4.0 connected machines, sensors and enterprise systems. Industry 5.0 asks what that connectivity is for, and adds three requirements to efficiency: production must be human-centric, sustainable and resilient. Artificial intelligence is the most promising tool for meeting these requirements, and also the easiest to misapply.

The gap between a promising model and a dependable production system is wide. A detector that scores well on a public benchmark may fail when the camera tilts by a few degrees. A scheduling policy trained in simulation may be opaque to the engineers who must trust it. A prognostic model may be accurate yet cost more to run than the downtime it prevents. Closing this gap is an engineering problem as much as a modelling one: it concerns data, integration, tolerances, cost and human acceptance. This book is written for the engineers who must close it. It combines methodological reviews with results from real industrial deployments, and it reports what did not work alongside what did.

AIMS5.0 (Artificial Intelligence in Manufacturing leading to Sustainability and Industry 5.0) is a European research and innovation project supported by the Chips Joint Undertaking and national funding authorities under grant agreement no. 101112089. The target here is to accelerate the transition from Industry 4.0 to Industry 5.0 by integrating AI systematically into industrial processes. Rather than treating algorithm development as an isolated activity, AIMS5.0 addresses the complete adoption lifecycle, from the definition of industrial use cases through data management, algorithm development, deployment, validation and standardisation.

AIMS5.0 (<https://aims50.eu/>) is built around twenty industrial use cases spanning semiconductor manufacturing, machine tools, robotics, logistics and production systems. These use cases provide the empirical foundation of this book and make it possible to examine AI at very different levels of technological maturity. Some cases remain exploratory, with the problem formulation and relevant performance indicators still being refined, whereas others have progressed to production pilots operating on live industrial lines. Taken together, they reveal not only what industrial AI can achieve but also the conditions that determine whether an approach can move beyond experimentation.

Semiconductor manufacturing appears repeatedly throughout the book because it provides an especially demanding environment for industrial AI. Few sectors combine such high degrees of automation with equally strict quality requirements

Pál Varga is with the Department of Telecommunications and Artificial Intelligence, Budapest University of Technology and Economics.

Kārlis Freivalds is with the Institute of Electronics and Computer Science.

Gian Antonio Susto is with the Department of Information Engineering.

Péter Olszsi is with AITIA International Inc.

and such costly consequences of downtime or process deviation. At the same time, the book extends well beyond this domain. Applications in cosmetics production, lighting, battery manufacturing and additive manufacturing demonstrate that many of the underlying engineering challenges are shared across sectors. The resulting lessons are therefore not tied to a single production technology, but reflect broader patterns in the adoption of AI for Industry 5.0.

II. ORGANISATION OF THE BOOK, AND SOME LESSONS LEARNED

The core chapters follow a common structure, moving from the background and state of the art in each field to AIMS5.0 applications and the research gaps that remain. Their sequence also reflects the functional loop of an intelligent industrial system: sensing the physical process, diagnosing its condition, deciding how to respond and acting on the plant. This progression is framed by a methodological foundation at the beginning of the book and by emerging language-model-based capabilities at the end.

Chapter 2, *Machine Learning in Industry 5.0*, provides this foundation. It shows how Industry 5.0 broadens the productivity-oriented perspective of Industry 4.0 by placing greater emphasis on human-centricity, resilience and sustainability. The chapter also explains why established non-neural methods remain important in industrial settings, particularly when interpretability, data efficiency or limited computing resources matter. The AIMS5.0 use cases show how these methods support tasks ranging from condition monitoring and predictive maintenance to production planning. Across them, the same principle recurs: industrial AI is most effective when domain expertise continues to shape both the model and the decisions made from its output.

Chapter 3, *Image Processing: a Review on Industrial Applications*, turns to the sensing stage. It examines industrial vision for quality control, automation and real-time monitoring, where model accuracy must be balanced against processing speed and the consequences of incorrect decisions. Modern vision methods are discussed through AIMS5.0 applications in robotic grasping, automated insertion, semiconductor inspection and assembly support. The chapter also shows how synthetic data and continual learning can reduce annotation effort and help vision systems adapt as products and operating conditions change. These cases present perception not as an isolated recognition problem, but as the first step in a wider decision and control process.

Chapter 4, *Anomaly and Fault Detection and Predictive Maintenance*, moves from sensing to diagnosis. It considers how AI and connected sensing can be used to identify abnormal behaviour, detect faults and anticipate maintenance needs. Particular attention is given to explainability and uncertainty,

since industrial users must understand not only what a model predicts but also how much confidence can be placed in that prediction. The use cases span production processes, machining and edge-based monitoring, including the transition from experimental anomaly detection to production-ready deployment. This transition is important because a diagnostic model becomes useful only when its warning can lead to timely action without overwhelming operators with unreliable or poorly contextualised alerts.

The next stage is decision making. Chapter 5, *Reinforcement Learning, Optimization and Scheduling*, examines planning and scheduling problems in which conventional approaches may struggle with changing conditions or real-time requirements. Reinforcement learning and related optimisation methods are illustrated through semiconductor scheduling, supply-chain planning and robot motion planning. The central message is that a useful industrial decision must do more than optimise an abstract objective: it must fit the production environment, remain understandable and adapt when operating conditions change.

Chapter 6, *Robotics, Automation, and Process Management*, completes the sense–diagnose–decide–act loop by focusing on execution in the physical system. It connects robotics and automation with the broader management of production processes, using AIMS5.0 examples from wafer fabrication, intralogistics, material handling and automated production. These cases show that deployment is not a final step added after model development. The way an AI system will be integrated, operated and maintained influences its (re-)design.

Chapter 7, *Large Language Models in Industry 5.0*, extends the discussion toward knowledge access and human–AI interaction. In engineering environments, relevant knowledge is often distributed across documents, software tools and organisational boundaries. Large language models can help engineers work with this information, but their value depends on reliable grounding, appropriate safeguards and continued human oversight. The AIMS5.0 examples range from engineering copilots and knowledge retrieval to compliance support and natural language interfaces for robot programming, illustrating both the potential and the limits of these emerging systems.

Although the chapters focus on different technological functions, their use cases in AIMS5.0 overlap. These connections reinforce the main argument of the book: industrial AI creates value not through isolated algorithms, but through the careful integration of sensing, reasoning, action and human expertise.

Across the AIMS5.0 use cases, the strongest recurring lesson is that industrial AI depends as much on the structure of the engineering problem as on the learning algorithm itself. Data are often scarce, imbalanced or expensive to label, while maintenance and production knowledge may be fragmented across systems and documents. Improving the data pipeline can therefore matter more than increasing model complexity. At the same time, purely data-driven solutions are rarely sufficient on their own. The most robust approaches tend to combine learning with established engineering knowledge, using physical, geometric or process constraints to guide the model where possible. This also changes how performance should be judged. A higher model score has practical value only when it

improves an engineering outcome, such as detecting a relevant defect, respecting a process constraint or supporting a better maintenance decision. In some cases, the limiting factor lies not in the model but in the sensor arrangement, mechanical setup or surrounding production process. Industrial AI must therefore be evaluated as part of the complete system rather than as an isolated prediction task.

A second set of lessons concerns what happens when a promising model leaves the laboratory. Performance can deteriorate when the product, machine, operating regime or production site changes, which makes generalisation an explicit engineering requirement rather than something that is just assumed. Methods such as transfer learning and domain adaptation can help, but in some applications a deliberately specialised model with a clearly defined scope is the more dependable choice. Human factors are equally important. Operators and engineers need enough transparency to judge when an AI output can be trusted. Finally, deployment itself requires sustained engineering. Models must fit existing manufacturing systems, operate within practical computing constraints and remain maintainable as data and processes evolve. The AIMS5.0 cases show that these considerations cannot be postponed until the end of development. They shape the choice of data, models, interfaces and evaluation criteria from the beginning, and they ultimately determine whether an experimental result can become a reliable in industrial operation.

III. HOW TO READ THIS BOOK

The chapters are written so that each can be read independently, but the book is most informative when the connections between them are kept in view. Read as a whole, the research gaps identified in the individual chapters form a broader agenda for the next stage of industrial AI. Questions of transferability across machines and sites, interpretability for operators, scalability under real-time and edge constraints, data privacy and effective human–AI collaboration appear in different forms throughout the book. Their recurrence suggests that the central challenge of Industry 5.0 is no longer simply to demonstrate that AI can solve isolated industrial tasks. The more demanding objective is to create AI-enabled systems that continue to perform reliably when they encounter real production variability, organisational constraints and human decision makers.

The chapters that follow document what has already been demonstrated under industrial conditions while remaining explicit about the limitations that still prevent broader adoption. In this sense, the book is intended neither as a catalogue of algorithms nor as a collection of success stories. It is an engineering account of how industrial AI is being developed, tested and integrated into the transition toward Industry 5.0, and of the work that still remains before such systems can be considered dependable at scale.

ACKNOWLEDGMENT

The research leading to these results was conducted in the AIMS5.0 project, which received funding from the Chips Joint Undertaking under grant agreement No. 101112089 and from the partners' national funding programmes.

Machine Learning in Industry 5.0

Pál Varga, and Péter Olaszi

Abstract—Industry 5.0 extends the productivity-oriented perspective of Industry 4.0 by placing greater emphasis on human-centricity, resilience, and sustainability in manufacturing. Although deep neural networks dominate much of current AI research, traditional machine learning (ML) methods remain widely used in industrial practice. Their continued relevance stems from their interpretability, data efficiency, and suitability for deployment on resource-constrained hardware. This paper examines their role through the European AIMS5.0 project. More than 40 traditional ML, optimization, and signal-processing methods are mapped to 13 industrial use cases covering machine tools, intralogistics, semiconductor manufacturing, supply chains, and other production settings. We describe the main concepts behind the different method families and summarize the observed selection logic in a decision diagram based on task type, label availability, and the need for operator-facing explanations. Across the use cases, a common human-in-the-loop pattern emerges: expert feedback is used to refine the models, while operator actions connect model outputs back to the physical process. The reported outcomes include an increase in overall equipment effectiveness and reduction in electricity cost, as well. At the same time, the use cases show that strong technical performance does not necessarily translate into scalable deployment. Based on these findings, we derive a research agenda centered on data-efficient learning, controlled adaptation to concept drift, decision-oriented and resource-aware ML, and evaluation criteria that extend beyond predictive accuracy.

Index Terms—Industry 5.0, machine learning, industrial AI, anomaly detection, explainable AI, human-in-the-loop, predictive maintenance, AIMS5.0

I. INTRODUCTION

Artificial Intelligence (AI) and Machine Learning (ML) extend industrial digitalization by enabling systems to infer patterns from data and use them for prediction, classification, anomaly detection, and decision support. These capabilities complement, rather than replace, conventional automation. automation executes predefined logic, whereas ML can derive models from operational data and update decisions when conditions or data distributions change. In Industry 4.0, this is particularly relevant to tasks such as predictive maintenance, data-driven quality assessment, process monitoring, and optimization.

Industry 5.0 broadens the objectives of industrial development beyond productivity and efficiency to include human-centricity, resilience, and sustainability [1]. These objectives are not achieved by AI/ML alone. Instead, AI/ML provides specific analytical and adaptive capabilities within a wider socio-technical system. For example, ML can estimate equipment failures or process deviations before they occur, support

operators with data-driven recommendations, and identify operating parameters associated with lower resource consumption. The resulting actions still depend on automation systems, domain constraints, human decisions, and the surrounding digital infrastructure. This distinction is important when assessing the contribution of AI/ML to Industry 5.0.

This paper is about ML in Industry 4.0 and 5.0, and separates traditional ML solutions from neural-network-based AI approaches. Traditional ML methods rely on structured feature representations and statistical models – such as regression, decision trees, ensemble methods, or support vector machines – to learn predictive relationships from data [2], [3]. These approaches typically require domain-engineered features, are computationally efficient, and offer higher interpretability, making them well suited for many industrial tasks with limited data or strict transparency requirements [4], [5]. In contrast, deep neural network (DNN)-based approaches learn hierarchical feature representations directly from raw data through multilayer architectures, enabling high performance in complex tasks such as image analysis, speech, or time-series modeling. However, DNNs generally demand larger datasets, greater computational resources, and exhibit lower inherent interpretability [6]. Thus, traditional ML and DNN-based methods complement each other: the former excels in structured, explainable industrial contexts, while the latter provides powerful end-to-end modeling for perception and high-dimensional data.

II. A GENERAL OVERVIEW OF MACHINE LEARNING IN INDUSTRY 5.0

Machine Learning (ML) provides a suite of statistical and algorithmic tools that enable data-driven modeling, prediction, and optimization across industrial processes [4], [7]. While deep neural networks dominate much of current AI research, a wide range of non-neural ML methods remains fundamental in industrial environments, owing to their interpretability, modest data and computational demands, and ease of integration with existing control and monitoring systems [5].

This section first outlines the basic distinctions between supervised, unsupervised, semi-supervised, reinforcement, and evolutionary learning. It then explains the concepts behind the method families employed in the AIMS5.0 use cases (Table I): classification and ensemble methods, clustering and dimensionality reduction, anomaly detection, optimization and search, signal processing and time-series analysis, explainability and causal analysis, and statistical monitoring and simulation. Finally, it surveys typical industrial applications – sustainable production, resilient supply chains, collaborative robotics, and digital twins – and discusses how the data regime and the role of the human operator guide the choice among these methods in an Industry 5.0 setting [8].

P. Varga is with the Department of Telecommunications and Artificial Intelligence, Budapest University of Technology and Economics, Műegyetem rkp. 3., H-1111 Budapest, Hungary.

P. Olaszi is with AITIA International Inc., AI and IIoT Division, 7. Záhony street, 1031 Budapest, Hungary.

TABLE I
MACHINE LEARNING ALGORITHMS EMPLOYED IN AIMS5.0 USE CASES

Algorithm / Use case	03	05	07	08	09	11	12	13	15	16	17	19	20
<i>Classification & Ensemble Methods</i>													
Decision Trees	X									X			
Gradient Boosted Decision Trees (GBDT)	X						X						
Extreme Gradient Boosting (XGBoost)							X						
Regression Tree							X						
Random Forest		X										X	
K-Nearest Neighbors (KNN)		X											
Logistic Regression		X											
Support Vector Machines (SVM)		X											
Iterative self-training			X										
Binary classification						X							
<i>Clustering & Dimensionality Reduction</i>													
K-Means Clustering					X							X	
Density-Based Spatial Clustering of Applications with Noise (DBSCAN)								X					X
Euclidean Clustering		X		X									
Principal Component Analysis (PCA)					X								
<i>Anomaly Detection</i>													
Isolation Forest								X	X				X
<i>Optimization & Search</i>													
Genetic Programming					X						X		
Constraint Programming					X								
Two-stage Stochastic Programming (2SP)												X	
Chance-constrained Programming (CCP)												X	
Simulated Annealing						X							
Simultaneous Perturbation Stochastic Approximation (SPSA)						X							
Reinforcement Learning			X										
Incremental learning			X										
<i>Signal Processing & Time Series</i>													
Time Series Analysis		X			X		X	X					
Fast Fourier Transformation (FFT)		X						X					
Harmonics Analysis		X											
Motif Discovery		X											
Change Point Detection		X											
Symbolic Aggregate Approximation (SAX)					X								
Kalman Filtering				X									
<i>Other Methods</i>													
Common Spatial Pattern (CSP)				X									
Linear Regression										X			
Shapley Additive Explanations (SHAP)							X						
Depth-based Isolation Forest									X				
Feature Importance (DIFFI)									X				
Accelerated Model Explanations for Anomaly Detection (AcME-AD)									X				
Balanced Anomaly and Latent Inference Framework (Extended B-ALIF)									X				
Causal Rule Mining (CRM)												X	
Statistical Process Control (SPC)											X		
Discrete-event simulation			X										
Markov decision process			X										
Semi-supervised learning			X										

A. Basic Distinctions of Machine Learning Methods

In this subsection, let us provide a brief overview of the main distinction of machine learning methods – from supervised learning through unsupervised and semi-supervised training to the classical reinforcement learning and evolutionary methods [4], [7].

Traditional supervised algorithms such as *linear and logistic regression*, *decision trees*, *random forests*, and *support vector machines* (SVM) have long been applied to regression and classification problems in manufacturing, logistics, and quality assurance [4], [5]. Regression models enable parameter esti-

mation and demand forecasting where causal relationships are approximately linear. Decision trees and ensemble methods, including random forests and gradient boosting, are particularly effective for fault detection, product quality classification, and process optimization, where mixed numerical and categorical inputs must be interpreted under uncertainty [9]. SVMs, owing to their robustness to high-dimensional feature spaces, are widely used in vibration-based predictive maintenance, anomaly detection, and condition monitoring tasks [10].

Unsupervised methods provide insight when labeled data are scarce or unavailable. *Clustering* algorithms such as *k*-

means, *Density-Based Spatial Clustering of Applications with Noise (DBSCAN)*, or *hierarchical clustering* are widely used for sensor data segmentation, equipment behavior profiling, and energy-consumption pattern analysis [11]. *Principal Component Analysis (PCA)* and other dimensionality reduction techniques (e.g., Independent Component Analysis, t-SNE) support multivariate process monitoring and fault isolation by identifying latent variables that capture the majority of variance in sensor networks [12]. These approaches have been instrumental in process control and statistical quality monitoring, often serving as lightweight, explainable baselines for more complex hybrid models [7].

Industrial datasets often contain abundant unlabeled samples; hence, *semi-supervised* and *active learning* frameworks are gaining traction [13], [14]. Label propagation, co-training, and self-training approaches leverage limited labeled data to expand model generalization, while active learning strategies query human experts for the most informative samples. Such human-in-the-loop paradigms align strongly with the Industry 5.0 principle of human–AI collaboration, ensuring that machine intelligence complements domain expertise rather than replacing it [8].

Classical reinforcement learning methods—such as tabular Q-learning or policy iteration—are applicable to process scheduling, adaptive control, and logistics optimization when the state and action spaces are well defined [15], [16]. Similarly, *genetic algorithms* and *evolutionary strategies* are widely used for parameter tuning, production scheduling, and layout optimization, particularly where the search space is complex but gradient information is unavailable [17], [18].

In the following sections, let us provide an overview of the various ML algorithms with base references. Table I lists a heterogeneous set of methods: classical learners, clustering and anomaly detectors, mathematical optimization techniques, signal-processing tools, and explainability add-ons. The following sections explain the concept behind each group, so that the use cases of Section III can be read without consulting the primary literature. Standard textbooks for the statistical learning methods are [2], [3].

B. Classification and Ensemble Methods

Decision trees split the feature space recursively, choosing at each node the feature and threshold that best separates the target values, e.g., by minimizing Gini impurity for classification or squared error for regression. When the target is continuous, the same procedure yields a *regression tree*; the Classification and Regression Trees (CART) framework covers both [19]. A single tree is easy to read but unstable, which motivates ensembles. A *random forest* trains many trees on bootstrap samples, with a random subset of features considered at each split, and averages their votes; this reduces variance while keeping the method largely parameter-free [20]. *Gradient boosted decision trees (GBDT)* instead build shallow trees sequentially, each one fitted to the residual errors (the negative gradient of the loss) of the current ensemble [21]. *XGBoost* is a scalable GBDT implementation that adds explicit regularization of tree complexity, second-order loss approximation, and sparsity-aware split finding [22].

K-nearest neighbors (KNN) stores the training set and labels a new sample by a majority vote of its k closest neighbors in a chosen distance metric; it needs no training phase, but prediction cost grows with the data size [23]. *Logistic regression* models the probability of a class as a sigmoid function of a linear combination of the features, which makes the fitted coefficients directly interpretable as log-odds contributions. *Support vector machines (SVM)* find the separating hyperplane with maximal margin between classes, and kernel functions extend this to non-linear boundaries [24]. Several of these models are used for *binary classification*, the case where the output is one of two states (e.g., normal/faulty, go/no-go), usually evaluated with precision, recall, and ROC-based metrics.

Iterative self-training is a semi-supervised wrapper around any classifier: a model trained on the small labeled set predicts labels for unlabeled samples, the most confident predictions are added to the training set, and the process repeats [13]. It is attractive when annotation is expensive, as in visual assembly monitoring, but the confidence threshold must be chosen carefully to avoid reinforcing early mistakes.

C. Clustering and Dimensionality Reduction

K-means partitions the data into k groups by alternately assigning each point to the nearest centroid and recomputing the centroids, minimizing the within-cluster sum of squared distances; it is fast but assumes roughly spherical clusters and requires k in advance [11]. *DBSCAN* defines clusters as regions of high point density: a point with at least a minimum number of neighbors within radius ϵ is a core point, clusters grow through connected core points, and isolated points are labeled noise. It therefore finds arbitrarily shaped clusters and outliers without fixing the number of clusters [25]. *Euclidean clustering* is a special case used on 3D point clouds: points closer than a distance threshold are joined into the same object, and a KD-tree index makes the neighbor search efficient [26].

Principal component analysis (PCA) projects multivariate data onto orthogonal directions of maximal variance, obtained from the eigenvectors of the covariance matrix. Keeping the first few components compresses correlated sensor channels into a small number of latent variables, which supports both visualization and multivariate process monitoring [27].

D. Anomaly Detection

Anomaly detection searches for observations that deviate from the expected behavior, often without labeled faults [28]. *Isolation Forest* exploits the fact that anomalies are few and different: it builds random trees by choosing a random feature and a random split value, and anomalies are isolated after fewer splits than normal points. The anomaly score is derived from the average path length over the forest, so no distance or density needs to be computed, and the method scales linearly with the data size [29]. Because an unsupervised score does not always match what domain experts consider anomalous, *Bayesian Active Learning Isolation Forest (B-ALIF)* uses the Isolation Forest output as a prior and updates it with a small number of expert labels, selected by active learning [30].

E. Optimization and Search

Several use cases rely on mathematical optimization or search rather than on learning in the strict sense. *Genetic programming* (GP) evolves computer programs, typically represented as expression trees, through selection, crossover, and mutation, guided by a fitness function [31]. In scheduling, the evolved trees are priority rules (dispatching rules) that combine job attributes such as due date, weight, and processing time. *Constraint programming* (CP) models a problem as variables with finite domains and declarative constraints, and solves it by combining constraint propagation, which prunes infeasible values, with systematic search; it is especially effective for scheduling with complex resource and timing restrictions [32].

When parameters are uncertain, stochastic programming is used. In *two-stage stochastic programming* (2SP), first-stage decisions (e.g., capacity or order quantities) are made before the uncertainty is revealed, and second-stage recourse decisions correct them for each scenario; the objective minimizes first-stage cost plus expected recourse cost [33]. *Chance-constrained programming* (CCP) instead requires that constraints hold with at least a given probability, e.g., that demand is met in 95% of cases [34].

Simulated annealing is a metaheuristic inspired by the cooling of metals: it explores the solution space by random moves, accepting worse solutions with a probability that decreases as a “temperature” parameter is lowered, which helps it escape local optima [35]. *Simultaneous perturbation stochastic approximation* (SPSA) is a gradient-free optimizer for noisy objectives: it perturbs all parameters simultaneously in a random direction and estimates the gradient from only two measurements of the objective, regardless of the number of parameters [36]. This makes it suitable for tuning physical systems, such as antenna configurations, where each evaluation is a measurement.

A *Markov decision process* (MDP) formalizes sequential decision making with states, actions, transition probabilities, and rewards; its solution is a policy that maximizes the expected cumulative reward [37]. *Reinforcement learning* (RL) learns such a policy from interaction or simulation when the transition model is not known explicitly [15]. *Incremental learning* refers to models that are updated sample by sample or batch by batch as new data arrive, without retraining from scratch, which suits non-stationary production environments and resource-constrained devices [38].

F. Signal Processing and Time Series

Time series analysis covers methods that model temporal dependence, trend, and seasonality in sequentially recorded data, such as autoregressive and moving-average models used for forecasting and residual-based monitoring [39]. The *fast Fourier transform* (FFT) computes the discrete Fourier transform in $O(N \log N)$ time, converting a signal from the time domain to the frequency domain [40]. On rotating machinery, *harmonics analysis* then tracks the amplitudes at multiples of the fundamental rotation or tooth-passing frequency, since faults such as tool breakage, imbalance, or gear damage change these harmonic patterns [41].

Motif discovery finds the most similar pairs of subsequences in a long time series, i.e., recurring patterns; the matrix profile computes, for every subsequence, the distance to its nearest neighbor, from which both motifs (repeated behavior) and discords (unusual behavior) can be read [42]. *Change point detection* locates the time instants where the statistical properties of a signal (mean, variance, or distribution) change abruptly, e.g., at the start and end of a machining operation or at a process drift [43]. *Symbolic aggregate approximation* (SAX) first reduces a normalized time series to segment means (piecewise aggregate approximation), then maps each mean to a letter using equiprobable breakpoints of the Gaussian distribution. The resulting strings can be searched and compared with text-based methods, and domain experts can express patterns of interest directly as symbol sequences [44].

The *Kalman filter* recursively estimates the hidden state of a linear dynamic system from noisy measurements. Each step predicts the next state from a motion model and then corrects the prediction with the new measurement, weighting the two by their uncertainties [45]. In tracking, the prediction step bridges short gaps when a sensor loses the target. *Common spatial patterns* (CSP) is a supervised spatial filtering method for multichannel signals, most notably EEG: it finds linear combinations of channels whose variance is maximal for one class and minimal for the other, and the log-variance of the filtered signals serves as a compact feature for motor imagery classification [46], [47].

G. Explainability and Causal Analysis

Explainable AI (XAI) methods make the output of a model understandable to its users, which is a prerequisite for human oversight in Industry 5.0 [6]. *Shapley additive explanations* (SHAP) assign each feature a contribution to an individual prediction based on Shapley values from cooperative game theory, i.e., the average change of the model output when the feature is added to all possible subsets of the other features. The contributions add up to the difference between the prediction and the average prediction [48]. SHAP is model-agnostic, but efficient exact variants exist for tree ensembles such as XGBoost.

Anomaly detectors need explanations tailored to their output. *Depth-based Isolation Forest Feature Importance* (DIFFI) derives global and local feature importance from the Isolation Forest itself: features that isolate anomalous points at shallow depths, and do so more than for normal points, receive higher importance [49]. *Accelerated model explanations for anomaly detection* (AcME-AD) is a model-agnostic alternative. It builds on AcME, which computes importance scores by perturbing each feature along the quantiles of its empirical distribution [50], and adapts this to anomaly scores, providing fast local feature importance together with a what-if analysis tool [51].

Causal rule mining (CRM) goes one step beyond correlation. It first mines association rules (“if condition X holds, failure Y is frequent”), then tests whether each rule is causal by comparing matched groups of records that differ only in the presence of X , in the manner of a retrospective cohort study [52].

H. Statistical Monitoring and Simulation

Linear regression models a continuous output as a weighted sum of input features, fitted by least squares; it remains the default baseline for yield or performance prediction because its coefficients are directly interpretable [2]. *Statistical process control* (SPC) monitors a process with control charts: a statistic such as the sample mean is plotted over time against control limits, typically three standard deviations from the in-control mean, and points outside the limits or non-random patterns signal special causes of variation [53]. Finally, *discrete-event simulation* (DES) models a system as a sequence of events (e.g., job arrival, machine start, machine breakdown) that change its state at discrete time points. It is not a learning method, but it provides the test bed on which schedulers, RL agents, and digital twins are trained and validated before deployment [54].

I. Typical ML applications for Industry 4.0

The classical ML methods of Section II-A support numerous operational tasks: predictive maintenance, statistical process control, supply chain optimization, product defect detection, and energy-efficiency analytics [4], [55]. In the context of Industry 5.0, their roles extend toward *transparent*, *sustainable*, and *human-centric* decision support systems [8], [56]. Because they are interpretable and computationally efficient, non-neural methods remain a critical component of trustworthy and certifiable AI pipelines that must comply with emerging regulatory frameworks such as the EU AI Act [57].

There are various types of applications already in practice. In perception-oriented tasks, ML methods for image recognition and computer vision enable automated quality control, defect detection, and visual inspection, reducing human error and inspection time [58]. Predictive models applied to sensor and maintenance data drive predictive maintenance, allowing early fault detection and optimized service scheduling that minimize downtime [9]. In robotics, learning algorithms support perception, motion planning, and adaptive control, making human-robot collaboration safer and more flexible [59]. For process optimization and control, supervised and reinforcement learning models adapt production parameters in real time to improve yield, precision, and resource utilization [60].

At a systemic level, supply chain optimization leverages ML for forecasting, risk analysis, and dynamic routing under uncertainty [61], [62], while energy optimization models contribute to sustainable operations through load balancing, peak-demand reduction, and efficiency analytics [63].

1) *ML for Sustainable Production*: Sustainability in manufacturing increasingly relies on ML to optimize energy use, minimize waste, and promote circular production models [64]. Classical regression and ensemble-based predictors are often applied to model energy consumption across machines and process stages [63]. This enables adaptive control of production schedules to reduce peak demand. Reinforcement learning and optimization algorithms support dynamic resource allocation, balancing throughput with environmental performance indicators such as carbon intensity and material utilization. In predictive quality analytics, unsupervised learning methods

detect deviations that lead to rework or scrap, contributing directly to waste reduction.

Furthermore, life-cycle data combined with clustering and probabilistic models help forecast maintenance or recycling needs, advancing closed-loop material flows. Through these data-driven insights, ML transforms sustainability from a static compliance target into a continuously optimized operational objective aligned with Industry 5.0's environmental and social pillars.

2) *ML for Resilient and adaptive supply chains*: Modern industrial supply chains must respond to disruptions ranging from component shortages to geopolitical shocks – this requires predictive and adaptive intelligence. ML methods can enhance resilience by enabling probabilistic demand forecasting, anomaly detection in logistics data, and adaptive routing through reinforcement learning [61]. Ensemble models and Bayesian networks quantify uncertainty, allowing planners to make risk-aware decisions rather than deterministic forecasts [62]. Clustering and outlier analysis further identify supplier reliability patterns or hidden dependencies that may threaten robustness. ML-driven supply chains transition from reactive logistics systems toward proactive, adaptive ecosystems – a core requirement of Industry 5.0 resilience.

3) *Collaborative robotics*: Collaborative robots (cobots) are one of the physical embodiments of human-machine cooperation envisioned in Industry 5.0. Machine learning algorithms underpin perception, intent recognition, and adaptive control that allow cobots to safely share workspaces with humans [59]. Supervised and unsupervised learning methods are applied in cobots to process multimodal sensory data—vision, depth, force, and gesture. With these classification of human states and real time prediction of motion trajectories are also possible. Reinforcement learning and imitation learning frameworks optimize task execution policies through experience or human demonstrations while respecting safety and ergonomic constraints [65]. Online learning further enables cobots to adapt to operator preferences or evolving assembly tasks without complete reprogramming. These ML-enabled mechanisms transform robots from deterministic executors into context-aware collaborators, improving flexibility, safety, and worker well-being in hybrid production cells.

4) *ML for Digital Twins in Industry 5.0*: Digital twins (DTs) serve as virtual counterparts of physical assets, integrating multi-source data for monitoring, simulation, and decision support [66]. Machine learning enhances DT fidelity by learning mappings between sensor inputs and system states, estimating unobservable parameters, and forecasting future behaviors [67]. Regression and state-space models support real-time condition estimation, while clustering and anomaly-detection methods reveal deviations between predicted and actual dynamics. Surrogate ML models, such as Gaussian processes [68] or gradient boosting regressors [21], approximate complex physical simulations, enabling faster what-if analyses and control optimization.

When integrated with reinforcement learning, DTs evolve into self-optimizing entities that can test adaptive control strategies before deployment. In Industry 5.0, DTs augmented

with ML not only reflect operational efficiency but can also incorporate human, social, and environmental dimensions.

5) *Selecting Among the Method Families*: It is already clear from the algorithm use vs. use case mapping in Table I that use cases' method choice is driven less by predictive accuracy alone than by the data regime and by the role of the human. Where labeled faults are rare, unsupervised detectors (Isolation Forest, DBSCAN, PCA) provide a starting point, and weakly supervised or active-learning extensions such as B-ALIF and self-training let experts refine them with few labels. Where decisions must be justified to operators or auditors, inherently interpretable models (linear and logistic regression, shallow trees, SPC charts, SAX patterns defined by experts) or post-hoc explainers (SHAP, DIFFI, AcME-AD) are preferred over opaque predictors. Where the task is a decision rather than a prediction, as in scheduling or supply planning, the optimization family (CP, GP, stochastic programming, RL on an MDP) is combined with discrete-event simulation to test policies before deployment. Finally, signal-processing methods (FFT, harmonics analysis, change point detection, Kalman filtering, CSP) mostly act as feature extractors in front of the learners, and their physical interpretability is often what makes the overall pipeline acceptable in a human-centric, Industry 5.0 setting.

Fig. 1 summarizes this reasoning as a decision diagram.

III. AIMS5.0 USE CASES

The AIMS5.0 project (*Artificial Intelligence in Manufacturing leading to Sustainability and Industry5.0*) is a European Innovation Action funded under the Key Digital Technologies Joint Undertaking, running from May 2023 to September 2026 with 53 participants from industry, research organizations, and academia [69]. Its aim is to strengthen European digital sovereignty by adopting, extending, and implementing AI-enabled hardware and software components across the whole industrial value chain. The results are validated in 20 use cases spanning 9 industrial domains, including semiconductor manufacturing, machine tools, lighting, food production, logistics, and supply chain management. The use cases therefore offer a broad, practice-driven view of how AI and ML are deployed in real production environments on the way from Industry 4.0 toward Industry 5.0.

This section presents the 13 use cases in which traditional ML methods play a substantial role; the algorithms they employ are summarized in Table I, and their concepts are explained in Section II. Each use case is described in terms of its industrial problem, the ML methods applied and the reasons for choosing them, and the results or lessons learned. The use cases are ordered by their project numbering, from edge-level condition monitoring (UC03, UC05) through human-centric assembly, robotics, and scheduling (UC07–UC09), to semiconductor fab operations, contamination management, and supply chain and ontology-based traceability (UC12–UC20). Taken together, they illustrate the recurring pattern discussed in Section II: interpretable and computationally lightweight methods are preferred where data are scarce, decisions must be explained to operators, or models must run on constrained

devices. Most of the use cases share the human-in-the-loop pipeline structure shown in Fig. 2.

A. UC03: Servitization and service delivery enabled by Zero administration

In industrial environments – especially within manufacturing, energy, and transportation sectors – there is a growing demand for real-time machine monitoring and predictive maintenance solutions. Traditional approaches that rely on collecting all sensor data for analysis are often infeasible due to bandwidth constraints, latency requirements, and data privacy regulations. This use case addresses the need for autonomous, intelligent analytics capabilities at the edge. The use case made progress in predictive maintenance by analysing data to find correlations and identifying error codes, mainly related to battery issues. Various machine learning models were tested; a time-based dimension was introduced using a *gradient-boosted decision tree (GBDT)* model to predict which lawnmowers might fail and when the failure might occur.

The *Stream Analyse platform* [70] enables running analytics and machine learning models directly on edge devices. The platform supports a variety of lightweight and interpretable algorithms, optimized for constrained environments. Typical algorithms include *decision trees* for real-time classification tasks, and *rule-based models* for threshold monitoring, alerting, and expert system-like decision-making, often tailored to domain-specific knowledge. The solution works even offline or with intermittent connection.

The following best practices are used for deploying machine learning models on the Stream Analyse platform: (1) Machine learning models are selected based on hardware constraints. Decision trees and compact neural networks are preferred for their low resource requirements and high inference speed. (2) Preprocessing (cleaning, normalizing, and transforming sensor data) is implemented directly on the device, reducing the need for external computation. (3) Feedback mechanisms allow operators or connected systems to label and validate predictions, thereby improving model performance over time through supervised learning.

While this approach brings benefits, some challenges must be mitigated. Variations in hardware capabilities across different industrial devices require specific configurations and may limit the uniform deployment of models. Industrial equipment, even of the same type, may exhibit different behaviour under changing operational conditions, making it difficult to deploy a single model across all instances without fine-tuning. Edge devices also often have limited memory, CPU, and power availability, constraining the size and complexity of deployable models. Systems must ensure inference occurs within tight time constraints, especially in real-time or safety-critical environments. Over time, changes in equipment behaviour or environmental conditions can lead to model performance degradation. Updating models without cloud connectivity or significant downtime is a concern. The above methods are primarily applied for predictive maintenance, monitoring asset value, and estimating equipment lifetime in the servitization business model.

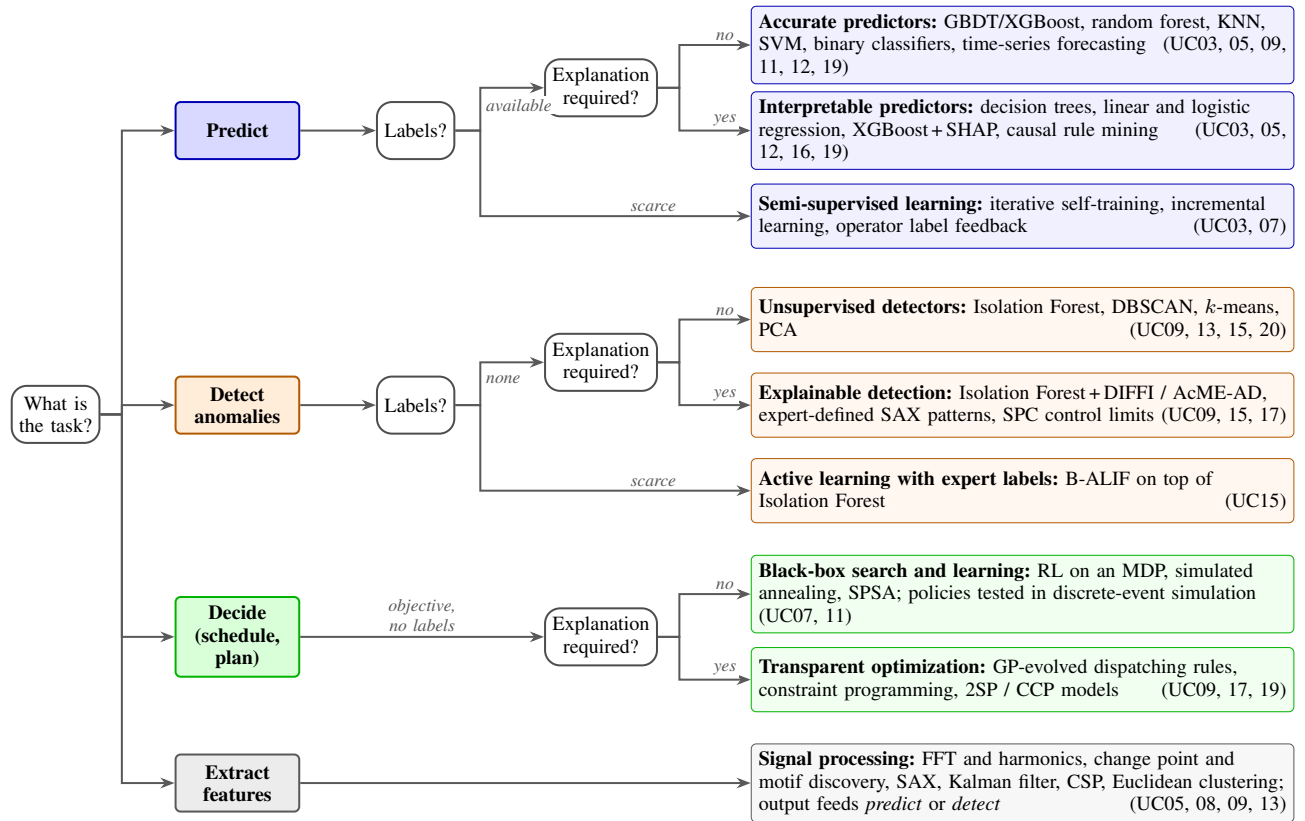


Fig. 1. Decision diagram for selecting among the traditional ML method families used in the AIMS5.0 use cases. The three questions are the task type, the availability of labels, and whether operators need an explanation of the output. Each leaf lists the method families and the use cases (UC) that applied them (cf. Table I).

often executed on edge devices close to the machine (UC03 Stream Analyze, UC05 Arrowhead, UC13)

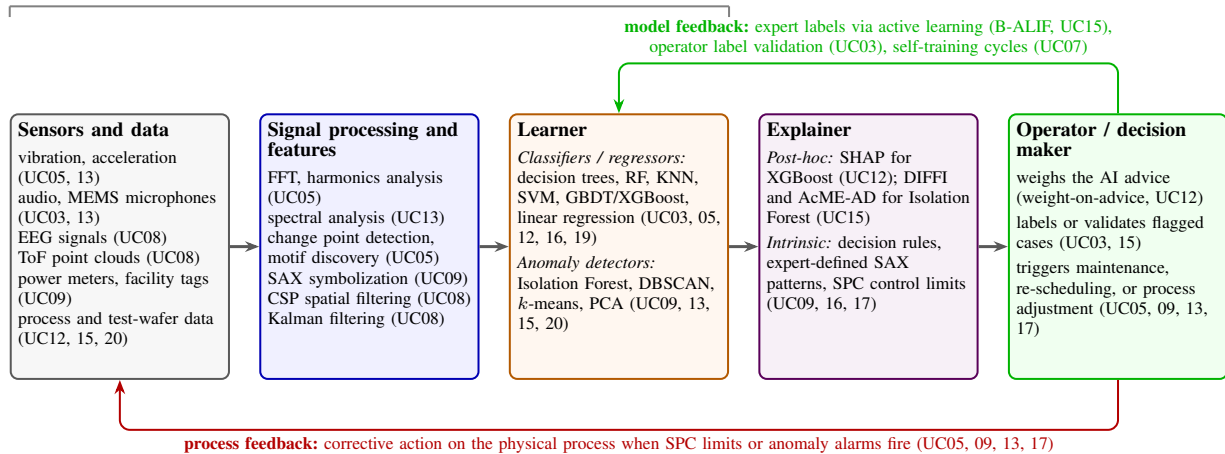


Fig. 2. Typical human-in-the-loop ML pipeline shared by the AIMS5.0 use cases. Sensor data are turned into features by signal processing, a traditional ML learner produces a prediction or anomaly score, and an explainer makes the result understandable to the operator. Two feedback loops close the cycle: expert labels refine the model, and operator actions change the physical process.

The use case also employs TinyML neural networks [71] and incremental learning techniques [38], which fall in the scope of neural networks within the realm of machine learning.

The deployment achieved quantifiable results in the Smart Lawn Service pilot in Germany, managing 500 robotic mow-

ers across 130 contracts. The analytical models deployed in Databricks demonstrated an 82.5% reduction in administrative costs, with one full-time equivalent now capable of managing 200 contracts compared to the baseline of 50 contracts. The AI model development cycle was reduced from

40 working days to less than one week, representing an over 80% time reduction. Multiple analytical models were successfully deployed for real-time monitoring, including Service Level Agreement (SLA) compliance tracking, site-related error scoring, customer churn risk assessment, and preventive maintenance monitoring. Edge AI models for lawn quality scoring and blade sharpness analysis (using audio sensors) were deployed to research mowers equipped with the Stream Analyze Engine. The solution advanced to Technology Readiness Level (TRL) 6-7 through side-by-side validation with actual service delivery operations, demonstrating scalability from 500 to potentially 10,000 products. This real-world validation confirmed the effectiveness of deploying lightweight decision trees and GBDT models on resource-constrained edge devices while maintaining continuous connectivity with cloud-based analytics infrastructure through the Databricks platform.

B. UC05: AI techniques at different layers in Machine Tool Domain

Large part heavy-duty precision machining is a long-term milling processes, when the machine works without operator supervision. Once the machine is set up and programmed with the necessary instructions, it can operate for hours without human intervention. However, errors may arise during the milling process, potentially affecting the quality of the final product. These errors may require parts of the workpiece to be re-machined, or, in the most severe cases, the workpiece needs to be discarded.

One of the most common problems in milling processes is tool breakage. Automatic tool breakage detection in milling operations is a problem of interest in the manufacturing industry, as a tool breakage can significantly affect product quality and increase production costs. To address this issue, implementing a system capable of detecting early tool breakage would be highly beneficial for reducing production costs by preventing defective parts and minimizing wasted time. Machine learning algorithms can analyse data to identify patterns indicative of faults, failures, or abnormal machine behaviour. These techniques have the potential to enhance the efficiency and reliability of milling processes.

To understand the behaviour of the data when the machine operates with a broken tool, experimental tests were conducted on a milling machine in IDEKO's laboratory. These tests involved performing machining passes under predefined cutting conditions using known tools while collecting data from accelerometers. Crucially, the tool's condition (broken or not) was predetermined, providing labelled data for tool breakage – an often challenging dataset to obtain.

The aim of this analysis is to detect tool breakage from vibration signals during the milling machining process. Therefore, an analysis based on the vibration signal has been performed. The first step in processing the vibration signal is to select a part of the signal, the part where the machine is actually machining. Then, the *Fast Fourier Transform* is applied. Applying the FFT decomposes the signal into its individual spectral components and thus provides information about its composition. In the next step, the values of the most

relevant harmonics are extracted. The number of peaks or harmonics is directly related to the number of teeth of the tool. Analysing these features extracted from the FFT signals, the pattern that was observed is that when the tool is broken the harmonic of the step frequency is smaller than that of the rotation or some multiple, and, conversely, when the tool is well, the amplitude of the harmonic of the step frequency is the largest. As the aim is to classify whether the test was carried out with a broken tool or not, several binary classification models have been adjusted to these data. More specifically, binary classification algorithms used to solve this problem have been *Random Forest*, *K-Nearest Neighbours*, *Logistic Regression* and *Support Vector Machines*. After optimising the hyperparameters and fitting the model to data, the accuracy obtained was greater than 0.95 for the four models, concluding that these models are able to correctly predict the class to which each observation belongs.

The next step involves implementing this approach under real production conditions. In a real production environment, raw vibration signals are not continuously available throughout the milling process. However, we have successfully extracted key features from these vibration signals. Consequently, the data available in real production consist of the extracted features rather than the raw signals. A significant challenge in this scenario is the absence of labelled data, as we lack examples of tool breakage events with corresponding labels. To overcome this limitation, the initial step in this approach is to segment the machining process, isolating the time series segments corresponding to each operation performed by the machine. This segmentation enables the comparison of repeated executions of the same process. By analysing these executions, we can detect any anomalies or deviations in machine behaviour during the process, thereby identifying potential issues.

The most challenging tasks included automatic data segmentation in the presence of noise and highly variable conditions. We also had limited labelled data: signals were not annotated with failure information. Ensuring model generalization across different tools, cutting conditions, and part types also proved to be difficult, as was handling multivariate time series from various sensors with potentially missing data, and adapting to different sampling frequencies and data formats in heterogeneous environments. Applying *Motif Discovery* helped automatically segmenting repetitive time series data from machining operations, which is traditionally done manually. The used algorithms include *Change Point Detection* and *Distance-Based Clustering*.

The tool breakage detection model achieved an accuracy exceeding 95% in laboratory experiments, successfully identifying breakages based on vibration-based tool harmonics analysis. The system processes high-frequency data at 2kHz with continuous KPI extraction at 1-second intervals, enabling real-time monitoring of machining operations. This infrastructure, integrated through the Arrowhead framework, allows for both continuous medium-frequency monitoring and on-demand high-frequency analysis. The automated data segmentation approach using the Motif Discovery Toolkit significantly reduces the time required for NC program segmentation

compared to manual methods. The complete solution is currently being validated in GOIMEK's shopfloor environment. It demonstrates the combination of traditional machine learning techniques with edge computing architectures for industrial predictive maintenance applications.

C. UC07: Digital Automated Luminaire Manufacturing Platform

This use case develops an adaptive scheduler for 3D printing hubs at Signify. The goal is to automate and optimize the scheduling process, which is currently done manually. The scheduler aims to anticipate and handle various anomalies, such as disruptions in material availability, and sudden unavailability of workforce. This enables the 3D printing hubs to operate in a more efficient and effective manner.

Signify is collaborating with Eindhoven University of Technology (TU/e) on this project. To validate the performance of the scheduler algorithm, a *discrete-event simulation* model of one of the printing hubs was built using Siemens Plant Simulation. This model allows for testing and evaluating the scheduler's performance in a controlled environment. The simulation model takes into account various factors such as the number of printers, their availability, the set of orders, the work schedule of the operators, and the manufacturing rules. The discrete-event simulation model successfully validated scheduler performance under various disruption scenarios, including material shortages and workforce unavailability.

The collaboration with TU/e involves the exploration of different approaches for the scheduler. The approaches being explored include transformer-based scheduling using sequence-to-sequence models, *Markov decision process with reinforcement learning*, and *genetic algorithms*. The goal is to find the most promising approach that can handle real-time large-scale scheduling problems. In the long term, the collaboration is expected to result in the development of an intelligent scheduler that is adaptable to different requirements and scales of different printing farms. The scheduler will optimize the assignment of print jobs to printers, considering factors such as just-in-time printing, filament reel utilization, and co-production of parts belonging to the same leading order. Overall, the use of AI algorithms and the simulation model developed in collaboration with TU/e will help Signify improve efficiency and productivity in their 3D printing operations.

In addition to scheduling, the use case also addresses Assembly Line Monitoring, which focuses on using Computer Vision (CV) to monitor the execution of manual assembly work. In the context of this assembly monitoring—specifically for object detection and tracking, Automated Data Annotations were explored. This involves *semi-supervised learning*, where small labeled datasets are combined with unlabeled data; pseudo-labeling uses an *iterative self-training* approach where the model generates its own training data; and *incremental learning* gradually improves of the model through multiple training cycles. These methods improve e.g. the *Mean Intersection over Union* (Mean IoU) metric from 0.4 to nearly 0.8 for the automated annotation pipeline, demonstrating the effectiveness of these approaches. However, further analysis

revealed that the effort required to scale this solution – specifically the installation of hardware (cameras, computers) and the extensive time still needed to label and train models for the diverse assembly activities of many product variants across different workstations – did not sufficiently outweigh the benefits. Consequently, full production deployment was halted, exemplifying a "pilot-learn fast-fail fast" approach valuable for industrial AI adoption.

Use Case 7 also uses modern deep learning approaches, which are out of scope for the current chapter: VideoMAE and Florence2 models are both transformer-based deep learning architectures; the semi-supervised learning pipeline uses deep learning models for pseudo-labeling; the reinforcement learning approach for scheduling likely uses neural networks (as most modern RL implementations do); finally, "early exploration" utilizes transformer-based scheduling. In industrial setting it is often necessary to combine them with traditional machine learning techniques.

D. UC08: AI supporting and strengthening human and manufacturing cycles

Use case 8 concentrates on the use of AI to support and strengthen human workers and manufacturing processes through material intralogistics. It demonstrates an Autonomous Mobile Robot (AMR) task and traffic management system that enables safe and energy-efficient intralogistics within a factory environment. The AMR perception system is based on sensor fusion, allowing object detection, human presence identification, and recognition of human gestures for manual AMR control when required. An active exoskeleton supports factory workers with disabilities, assisting those who would otherwise be unable, or would struggle to perform their tasks. Materials delivered to the workstation by AMRs are handled by the human worker using an AI-supported upper-body exoskeleton based on brain-signal analysis.

The use case employs a few traditional machine learning methods. *Kalman filtering* is used in the human detection and tracking system, helping to maintain continuous tracking even when sensor data is temporarily unavailable or unreliable. By predicting positions based on previous movement patterns, it fills tracking gaps and enables autonomous mobile robots (AMRs) to anticipate human movements during brief occlusions, significantly enhancing tracking robustness in dynamic factory environments.

In the exoskeleton control system, *Common Spatial Pattern* (CSP) is applied to EEG signals to extract meaningful features that distinguish different motor imagery patterns – such as imagined movements. This preprocessing step improves the classification accuracy of subsequent algorithms by highlighting the most discriminative aspects of brain activity patterns. This results in more reliable control of the exoskeleton through brain signals.

In the human detection system, *KD-tree-based Euclidean clustering* processes 3D point cloud data from sensors like time-of-flight (ToF) cameras to separate and identify distinct objects in the environment. By grouping points that likely belong to the same physical entity, the algorithm helps

AMRs distinguish between humans, objects, obstacles, and other environmental features. This segmentation is a crucial preprocessing step that transforms raw point cloud data into meaningful object representations that can be further classified and tracked [26].

Before switching to ToF cameras, a Depth-based clustering method was employed for LiDARs, in which the LiDAR 3D point cloud was segmented according to depth differences computed from LiDAR beam triangulation [72]. However, due to the high cost and fragility of LiDAR sensors, ToF cameras are preferred and employed instead.

The use case achieved 80% human presence detection accuracy within 5 meters, 85% gesture recognition accuracy across five gesture classes, and 75% 3D bounding box quality (IoU). The traditional ML techniques help more complex deep learning models (LSTM, BiLSTM, GRU, RNN, TCN) to operate effectively in the factory environment.

E. UC09: Energy-aware Scheduling of Batch Processing Machines in Wafer Fabs

The main goal of UC09 is to understand the energy flows of equipment in the semiconductor industry. A typical semiconductor factory needs as much energy as a small city does, because a lot of electrical power is used to maintain the clean room conditions, and for hundreds of equipment to produce the integrated circuits on the wafer. The teams already installed first sensor systems to measure the power consumption at some points within the factory. The main energy flows are known and first correlations have been found using historical and current weather data. A strong correlation between the power consumption, and the outside heat and humidity is evident from the observations.

Use case 9 employs several traditional machine learning algorithms. *Genetic Programming (GP)* is implemented to automatically discover efficient dispatching rules for energy-aware scheduling of batch processing machines in semiconductor manufacturing. The approach constructs tree structures representing priority indices that determine job scheduling order based on factors like job weights, ready times, and due dates. Initially developed for single tools, the system was progressively extended to handle tool groups and flexible flow shops with parallel identical batch processing machines. The framework incorporates time window decomposition to account for different job ready times and a decision theory heuristic to determine optimal idle time insertion in schedules.

The implementation uses the *GaLib framework* with C++ programming and follows an iterative learning approach with training and validation phases. When tested on real-world problem instances from IFD's wafer fabrication facilities in Dresden, the GP approach demonstrated average reductions of 16% in total weighted tardiness and 11% in total electricity costs compared to benchmark approaches using only manual scheduling processes. Field tests with productive lots validated these improvements in the front-end wafer facility. This scheduling approach is applied on diffusion furnaces and other high-energy-consuming equipment, so as to balance traditional production metrics with sustainability considerations under time-of-use energy tariffs.

Constraint Programming was developed as a complementary approach to the genetic programming methods for energy-aware scheduling in semiconductor manufacturing. It was designed to schedule jobs in parallel machine environments while accounting for grid operator's load shifting requests. This approach represents a key innovation in the project's second year, enabling the semiconductor fabrication facility to respond dynamically to external energy grid conditions and requests.

The implementation forms part of the broader strategy to enhance the digital twin that accounts for energy consumption, demand, renewable generation and grid procurement. This constraint-based scheduling system helps predict the fab's energy consumption patterns and enables automated decision-making in response to load adjustment requests from grid operators. When load thresholds are requested by grid operators during periods of energy scarcity or grid stress, the constraint programming system can rapidly decide whether to follow these requested thresholds based on production constraints and optimization objectives, contributing to both production efficiency and grid stability. This capability represents a critical advancement in enabling semiconductor manufacturing facilities to participate in demand response programs while maintaining production schedules.

K-Means Clustering is deployed for anomaly detection in the complex cooling systems of semiconductor fabrication facilities. The method analyzes approximately four months of time series data collected from cooling equipment sensors, comprising hundreds of machine tags measuring parameters like temperatures and flow rates. For each sensor tag, feature vectors are computed over 5-minute intervals, incorporating statistical measures including mean, minimum, maximum values, variance, and the slope of the regression line, which are then clustered to identify deviations from normal patterns.

Symbolic Aggregate Approximation (SAX) represents an innovative approach to analyze time series data from semiconductor facility management systems. The technique transforms continuous sensor readings like temperature or flow data into symbolic representations by dividing the time series into segments and mapping each segment to a symbol based on its statistical characteristics. This dimensionality reduction technique creates new feature vectors that can be used for efficient clustering and anomaly detection in the complex cooling systems operating within the cleanroom environments.

At IFAT's Villach site, SAX was applied to explore the frequency and timing of specific critical events in cooling system operation, such as temperature increases exceeding 3 degrees within 15-minute intervals. By leveraging domain knowledge from facility management experts, the SAX-based analysis proved particularly effective at detecting human-defined anomalies that might otherwise be missed in the vast datasets generated by the monitoring systems. This application demonstrates how SAX provides a complementary approach to traditional statistical methods, offering new insights for predictive maintenance and optimization of energy-intensive cooling infrastructure essential to semiconductor manufacturing.

Principal Component Analysis (PCA) serves as a dimensionality reduction technique, enabling the visualization and

analysis of highly complex multidimensional data from semiconductor manufacturing environments. Applied specifically to facility management data comprising over 700 tags and approximately 20GB of information collected over 3.5 months from December 2023 to March 2024, PCA transforms this high-dimensional feature space into a comprehensible two-dimensional representation that highlights the most significant variations in the dataset. This visualization revealed that the smallest cluster, containing only 66 observations out of thousands, harbored the most significant anomalies - including instances of actual equipment damage that had occurred. The approach was integrated into an interactive browser-based tool, combining AI-supported anomaly detection with human expertise for optimal facility management. The PCA transformation made it possible to understand relationships between multiple parameters simultaneously, such as cold water valve positions, bypass control valves, and outlet temperatures, providing insights that would be impossible to detect through traditional monitoring approaches. The application demonstrates how targeted machine learning can effectively monitor performance degradation and identify potential failures in critical manufacturing infrastructure without requiring comprehensive sensor coverage.

Time Series Analysis forms the foundation of energy consumption prediction and pattern recognition in Use Case 9, enabling the correlation of energy usage with production parameters, facility loading, and environmental factors. Applied to data from smart power meters installed at strategic locations throughout IFD's 200mm and 300mm wafer fabrication facilities, these methods identified strong relationships between weather conditions and energy demand, revealing that higher humidity and temperature significantly increase energy consumption due to the requirement for dry air in cleanroom environments.

The implementation at IFD includes forecasting models that predict energy consumption 24 hours in advance by combining historical energy patterns with weather data and production schedules. These analyses showed clear weekly patterns and seasonal variations, with box plot visualizations demonstrating the distribution of energy consumption across the weeks of the year using data from 2018-2023. By understanding these temporal patterns, the facility management team can implement peak shaving strategies, adjusting production schedules to reduce energy consumption during high-cost periods or when electrical power shortages might occur. The insights gained from time series analysis directly inform the "golden tool strategy," which aims to optimize energy efficiency using minimal smart metering infrastructure. During high-energy peaks, the system demonstrated potential savings of more than 750 kWh per day through tool scheduling and load shifting strategies.

The use case also employs neural networks for energy forecasting, but traditional ML approaches form the major part of the overall machine learning strategy. The implementation uses Jupyter notebooks, TensorFlow, scikit-learn, Bokeh, Pandas, and HDF5, among others.

F. UC11: AI-supported Industrial IoT for Indoor Food Production

The aim of UC11 is to enhance plant product lifecycle automation. The proposed system utilizes sensors to monitor the conditions of each plant and analyzes the data in real-time with AI-based decision-making algorithms. The framework employs IoT devices, automation, and digital infrastructure in vertical farming systems. The wireless connectivity enhancement component developed by GUT employs *simulated annealing* to improve signal quality in indoor farming environments. This algorithm controls the Electronically Steerable Parasitic Array Radiator (ESPAR) antenna's beam switching capabilities to maximize signal strength and reduce interference in complex indoor environments - filled with dense equipment, plants, and other obstacles. The simulated annealing approach enables adaptive beam switching without requiring environmental information like signal source positions, allowing the antenna to dynamically adjust its radiation pattern from 4,096 possible configurations to maintain reliable connectivity even when in case of interference or potential jamming.

The ML optimization models for the ESPAR antenna's radiation pattern adaptation prioritize *binary classification* with adjustable thresholds, achieving approximately 400% improvement in Signal-to-Noise and Interference Ratio (SNIR) - a 6 dB gain over the omnidirectional baseline in approximately 20 iterations. This approach is implemented on TTTech's Nerve computing unit which serves as the central processing unit for wireless connectivity in the indoor farming setup. The classification model processes signal quality indicators from the wireless communication modules and determines the optimal antenna configuration. This methodology enables the system to quickly identify the best antenna pattern without exhaustively scanning each configuration, providing efficient and reliable wireless connectivity for real-time monitoring and control of farming parameters.

Simultaneous Perturbation Stochastic Approximation (SPSA) algorithms are being investigated as an alternative approach for minimizing loss functions in antenna steering vector optimization for the wireless communication system. These algorithms focus on optimizing the antenna's radiation pattern by approximating gradient information to find optimal configurations without exhaustively testing all possibilities. In the context of the use case UC11, the SPSA approach represents an established methodology that serves as a baseline for comparing the performance of newer ML-based approaches being developed. The team is evaluating whether these traditional algorithms can be enhanced or replaced by more advanced machine learning techniques to improve adaptation speed and reliability in the dynamic indoor farming environment.

G. UC12: Human-Centered AI for the Optimization of Robust and Competitive Semiconductor Manufacturing Networks

Use case 12 aims to optimize semiconductor production and decision-making through a more efficient and sustainable use of resources, transportation, and energy - lowering the CO₂

emissions per product. The focus lies on the holistic visualization and realization of tool and process related machine state scenarios to increase transparency of the processes. The use case achieved a 33.8% increase in Overall Equipment Effectiveness (OEE) from a baseline of 42.9% (December 2023) to 57.4% (May 2025), exceeding the 10% target; potential to reduce cycle time by 23% through 50% reduction in repair variability; and a 8.3% difference in AI acceptance between experiment groups measured through weight-on-advice metrics.

Extreme Gradient Boosting (XGBoost) was implemented for work in progress (WIP) bubble prediction in semiconductor manufacturing. When training on one year of production data (2023) for the oven work center at IFD, XGBoost consistently proved to be the most effective model. The implementation uses machine, process, and product data (51 variables) to predict future WIP situations, helping to identify bottlenecks in advance and contributing directly to the achieved OEE improvements.

While *Shapley Additive Explanations (SHAP)* is not a predictive algorithm itself, it was implemented as an explainable AI approach to interpret the results of the traditional ML models. It was specifically used to analyze the feature importance of XGBoost models, increasing transparency and helping human decision-makers understand which variables most influence the WIP predictions. This implementation supports the human-centered aspect of the use case by making ML model outputs more interpretable. Experiments comparing black box versus transparent AI presentations showed that transparent AI achieved a weight-on-advice score of 0.58 compared to 0.63 for black box representations, demonstrating an 8.3% difference in how decision makers accept and trust AI predictions.

In the use case XGBoost emerged as the primary implemented solution based on its superior performance. Some other traditional ML algorithms have been considered or were in testing stages, but were not fully implemented: *Lasso regression*, *Ridge regression*, and *Elastic Net* regularization regression techniques were included in the planned algorithm testing suite – particularly for managing the high-dimensional feature space of the data –, but were not implemented eventually. *Support Vector regression (SVR)* and *Optimal Decision Tree* were also among the candidate algorithms. *Seasonal Autoregressive Integrated Moving Average with Exogenous Regressors (SARIMAX)* was considered as a comparison benchmark against XGBoost for time series prediction capabilities.

H. UC13: Intelligent Sensors improving Robustness of Automated Wafer Transportation and Storage Systems

Isolation Forest is used for anomaly detection in the audio signals collected from Overhead Transportation (OHT) shuttles in the workshop. The algorithm analyzes sound patterns from Micro-Electro-Mechanical Systems (MEMS) microphones installed along the rail system to detect abnormal acoustic signatures that might indicate mechanical issues before catastrophic failures occur. This approach enables predictive maintenance by identifying shuttles that deviate

from normal sound profiles. The implementation at Infineon Dresden leverages edge devices to process these audio files through feature extraction before applying the Isolation Forest algorithm, which effectively "isolates" anomalous sound patterns by partitioning the data space.

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is applied as part of the unsupervised anomaly detection strategy for monitoring the health status of automated material handling system components. This algorithm groups similar sensor readings together and identifies outliers in the acceleration data collected from the sensor platform housed in Front Opening Unified Pods (FOUPs). The density-based approach is particularly valuable in this use case as it doesn't require predefined knowledge about how many clusters exist in the data, making it suitable for detecting emerging failure patterns in the transportation systems where labeled failure data is scarce.

Traditional *spectral analysis* and *feature extraction* are implemented to process audio signals from the MEMS microphones monitoring the OHT shuttles, which operate on the rail system at Infineon Dresden. These methods transform time-domain sound data into frequency-domain representations that reveal characteristic patterns of both normal operation and potential mechanical failures. This enables the maintenance team to identify specific frequency signatures associated with bearing damage by comparing the frequency spectrum of good and bad bearings.

Traditional *time series analysis* techniques are applied to shuttle acceleration data from the multi-sensor platform (vibration, sound, and distance sensors). These algorithms process the data to detect shocks (peaks) and resonance frequencies that might indicate excessive forces on wafers or mechanical wear in the transportation system components. The implementation includes shock analysis algorithms that evaluate acceleration patterns to identify potentially damaging events during wafer transport, allowing for early intervention before components fail. This approach helped reduce maintenance efforts and saved significant downtime costs.

I. UC15: Productionized Machine Learning models at scale

Use Case 15 addresses the integration and deployment of scalable, interpretable, and production-ready AI/ML models for anomaly detection and predictive maintenance within semiconductor manufacturing processes. The use case is split to five complementary sub-activities for anomaly detection, defect classification, maintenance status classification, OEE prediction, and defect density image classification. Out of these, UC15-subA, Anomaly Detection and Virtual Inspection employs non-neural AI methods.

Isolation Forest is the core algorithm used in UC15-subA for unsupervised anomaly detection in semiconductor manufacturing. LFoundry chose this algorithm because it has been shown to be one of the best-performing anomaly detection approaches across many settings. It's well-suited for identifying anomalies in multivariate time-series sensor data to detect early-stage warnings of faults in sputtering deposition and metal etch processes.

Depth-based Isolation Forest Feature Importance (DIFFI) was implemented as an explainability layer on top of the Isolation Forest algorithm in UC15-subA. Given that operators need to take actions based on anomaly detection results, an approach was needed to make the black-box anomaly detection interpretable. DIFFI provides both local and global feature importance values, helping operators understand which sensor measurements contributed most to detecting an anomaly. This explainability component addresses the "human-in-the-loop" challenge by building trust in the ML solution by making its decisions transparent to users.

Accelerated Model Explanations for Anomaly Detection (AcME-AD) [51] was developed and implemented in UC15-subA as a model-agnostic explainability framework for anomaly detection in tabular data. It provides fast and interpretable insights into detected anomalies through local feature importance scores and a what-if analysis tool. This allows semiconductor manufacturing operators to understand the factors behind each detected anomaly without the computational overhead of more resource-intensive explainability techniques. This framework enabled a 55% reduction in time for monitoring industrial processes across four sputtering machines.

The *Balanced Anomaly and Latent Inference Framework (Extended B-ALIF)* algorithm extends the Isolation Forest model by incrementally incorporating expert feedback based on Bayesian Active Learning principles. It was implemented to improve detection performance in situations where labeled data is scarce, which is common in semiconductor manufacturing where fault examples are rare. The framework refines anomaly identification iteratively by leveraging limited human input, making it valuable in the Decision Support Systems developed for LFoundry, where tagged data for anomalies is limited.

J. UC16: AI enhanced production and performance of RF transceivers for civilian SATCOM and radar applications

The main aim of UC16 is to use AI solutions to improve the manufacturing throughput and sustainability of future radio frequency (RF) components and systems. AI-based methodologies are developed for design and facility optimization from radar applications through avionics to next generation high power satellite communication.

Linear Regression is utilized in the radio frequency (RF) component design and fabrication optimization processes. The teams from THALES and FORTH apply this algorithm to establish relationships between fabrication parameters and device performance characteristics. For example, during the manufacturing of RF Micro-Electro-Mechanical Systems (MEMS) switches and surface acoustic wave (SAW) temperature sensors, linear regression models help predict how changes in fabrication parameters (like dielectric film thickness or air gap spacing) affect performance metrics such as switching time and isolation. This approach helps understanding cause-effect relationships in the manufacturing process, where input features from the ISO quality control measurements can be directly correlated with output performance metrics.

Decision Trees are implemented as part of the supervised learning approach for quality control and fabrication process

optimization. Decision trees are used to classify potential defects or predict whether devices will meet specifications based on measurements taken during the ISO quality control process after each fabrication step. This is useful for identifying decision points in the manufacturing process that might lead to device failure or suboptimal performance. For example, when fabricating the mini-MEMS RF leakage sensors, decision trees help determine which combinations of process parameters are most likely to achieve the target specifications of 1 microsecond switching time and -10 dB isolation. Both algorithms form part of the use case's strategy to "utilize commonly adopted open-access algorithms" rather than developing new ones, with the focus being on proper feature and label selection during the training process.

K. UC17: Intelligent Contamination Management & Prescriptive Control Plans and Process Flows

Use case 17 primarily focuses on sensor networks, data collection infrastructure, and machine learning tools for contamination monitoring in semiconductor manufacturing facilities. The use case employs cost-effective static sensors utilizing meander structures, replacing expensive legacy systems. The sensors feed into a *data lake* architecture that consolidates data from multiple sources.

IFD designed a heuristic scheduling framework for flow shops based on *genetic programming*, applied to scheduling tasks related to Front Opening Unified Pod (FOUP) cleaning and monitoring processes. This helps optimize the sequence and timing of contamination control activities in the cleanroom environment, improving overall efficiency and contamination prevention.

Pfeiffer Vacuum improved its contamination control equipment with edge computing capabilities for real-time *anomaly detection*. The system analyzes equipment data (including measurement data, tool status, and analyzer status) every 5 seconds – a form of statistical anomaly detection that enables maintenance personnel to take immediate corrective action when issues arise, reducing downtime and preventing contamination events.

Fraunhofer IISB connected analytical instruments such as Inductively Coupled Plasma Mass Spectrometer (ICP-MS) with *machine learning models* for prediction and root cause analysis of contamination events. The implementation includes automated data transfer and ML-based heatmap visualization for metal contamination control measurements, reducing the need for time-intensive offline analysis.

While not always categorized as machine learning, *Statistical Process Control (SPC)* is being used in this use case for monitoring Airborne Molecular Contamination data from cleaned FOUPs. The documents show SPC data being visualized through tools like Tableau and Zeppelin. This statistical approach helps maintain control over the cleaning processes by identifying when measurements fall outside acceptable limits, allowing for process adjustments before contamination problems occur.

L. UC19: Extension of Digital Reference Ontology for Smart Sustainable Business Planning, Quality Traceability and Incident Handling

Use Case 19 addresses supply chain planning and quality management. One major driver of the recent semiconductor crisis is the bullwhip effect along OEM supply chains, which can be mitigated through closer connectivity and collaborative planning. IC miniaturization increases failure risks and complicates root cause analysis, wasting resources. This use case contributes to the elaboration of the Digital Reference Ontology (DRO) across three dimensions: traceability, supply chain management, and sustainability.

The *Random Forest* classification model was implemented for quality incident management in the semiconductor manufacturing process – specifically for the wire bonding manufacturing step. This model was used to differentiate between regular and problematic wire bond manufacturing conditions where problematic conditions were created through experiments that mimicked known root causes of failures. The method successfully identified three relevant machine-signal-related features important for classification that could be directly linked to potential IC failures. This approach enhances root cause analysis efficiency and enables proactive detection of failures in one of the most critical assembly steps in semiconductor manufacturing.

Causal Rule Mining (CRM) was applied to identify causal relationships between wire bonding and final test IC failures using *knowledge graph*-based traceability data. By adding cluster membership information from the knowledge graph, the team was able to conduct more sophisticated analysis of complex causal relationships that might otherwise remain hidden. CRM helps understand the cause-effect relationships in the manufacturing process, though it is less reliable with noisy or incomplete datasets.

K-means Clustering was used to validate observations from the machine signal data analysis. For instance, when analyzing the three relevant machine-signal features discovered by the Random Forest model, k-nearest neighbor validation achieved an accuracy of 0.738 (0.734; 0.742), confirming that clusters of different experimental conditions could be distinguished. However, k-means clustering is limited when dealing with high-dimensional features and imbalanced datasets, which are common in quality incident management scenarios.

For supply chain optimization, *two-stage stochastic programming (2SP)* and *chance-constrained programming (CCP)* models were implemented to handle demand uncertainty more robustly than traditional linear programming approaches. These stochastic optimization models showed superior performance under demand variability while balancing trade-offs among service level, inventory position, and planning stability.

M. UC20: Harmonization of Automated Digitized Equipment Control (AuDi-EC)

Use case 20 addresses the lack of a harmonized equipment test strategy for two 300 mm high volume factories, which should work as a "One virtual Fab". Test wafers are used for the permanent check of the tool status to assure the quality

demands of the customer. The ratio of test wafers compared to production wafers is higher and an important cost driver. The main goal is to balance the conflicting goals between quality and costs. Highest priority, especially for health and automotive customers, is to fulfil the quality demands. On the other hand, the processes must be lean and efficient to reach the cost targets. The test processes are very complex even in one of the existing factories. The goal of this use case is to harmonize different processes, tools and software to get a single monitoring strategy for the two different 300 mm production plants in Villach and Dresden.

Isolation Forests are used in the anomaly detection component of the test wafer strategy. In semiconductor manufacturing, detecting anomalies in equipment behavior is crucial for maintaining quality standards. Within Use Case 20, Isolation Forests help identify unusual patterns in test wafer data across both the Villach and Dresden facilities, supporting the harmonization efforts by detecting when equipment monitoring results deviate from normal parameters.

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is implemented for unsupervised anomaly detection in the test wafer monitoring process. In UC20, DBSCAN helps process engineers identify clusters of similar equipment behavior across the two manufacturing sites and detect outliers that might indicate equipment issues requiring intervention. This supports the OneVirtualFab concept by providing a consistent approach to anomaly detection across both facilities, contributing to the harmonization of equipment control processes and optimizing the use of test wafers.

IV. CHALLENGES, RESEARCH GAPS AND FUTURE DIRECTIONS

The use cases presented in Section III demonstrate that traditional machine learning remains highly relevant to industrial applications. It offers practical solutions when computational efficiency, limited data, and interpretability are important, particularly when compatibility with existing industrial systems must also be ensured. At the same time, the use cases make clear that successful industrial ML depends on far more than predictive performance [56]. Persistent challenges include data quality, generalization under changing conditions, and the maintenance of models after deployment. Equally important is the ability to incorporate human expertise and translate model outputs into reliable operational decisions.

Consequently, the transition from Industry 4.0 toward Industry 5.0 [1], [8] calls for a shift from an algorithm-centric view of ML to one that considers the entire system and its lifecycle. The central question is no longer simply whether a model achieves sufficient accuracy on a given dataset. Rather, it is whether that performance can be sustained, understood, validated, and translated into action throughout the lifetime of an industrial system.

A. Data Availability, Quality, and Heterogeneity

Industrial ML often operates in a data regime that differs fundamentally from that of benchmark applications. Data from normal operation are usually plentiful, whereas the events

of greatest interest – such as failures, quality deviations, or contamination events – occur only rarely. As a result, class imbalance and a shortage of labeled fault examples are common challenges.

This issue appears in several AIMS5.0 use cases. In UC05, labeled tool-breakage data could be generated under laboratory conditions, but equivalent labels were not readily available in real production. UC13 relies substantially on unsupervised anomaly detection, while UC15 combines Isolation Forest with expert feedback to progressively refine the identification of relevant anomalies.

The usefulness of an industrial dataset cannot be judged by its size alone. Sensor calibration, missing measurements, and sampling rates may be just as important, as may the relationship between the recorded measurements and the underlying physical events. Data collected from different machines or factories can also vary in format, resolution, and semantics. These differences make it difficult to reuse models directly across production environments.

A major research gap is therefore the development of *data-efficient industrial learning* methods that can exploit unlabeled process data while requiring only limited expert annotation. Active, semi-supervised, and incremental learning are particularly relevant in this context. Future research should also place greater emphasis on semantic interoperability. Ontologies and standardized asset descriptions, such as those of the ISO 23247 digital twin framework [73], can help distinguish genuine changes in the process from differences introduced by sensing and information infrastructures.

B. Generalization, Concept Drift, and Model Adaptation

A model that performs well on historical data or on a single machine cannot be expected to perform equally well indefinitely after deployment. Tools wear, materials and recipes change, and sensors may be replaced or recalibrated. Such changes can alter the distribution of the input data or the relationship between inputs and the target variable, a phenomenon known as concept drift [74].

This limitation is visible in UC05, where generalization across tools, cutting conditions, and part types remains challenging despite high accuracy under laboratory conditions. UC03 highlights a related issue, as differences among edge devices constrain uniform deployment. UC20 addresses the problem at a broader scale by considering how monitoring strategies can be harmonized between two fabrication sites.

Future systems therefore require mechanisms for recognizing when current operating conditions have moved beyond those represented in the training data. Monitoring should track predictive performance whenever labels eventually become available, but it should also examine shifts in feature distributions, anomaly scores, and residuals. Once a meaningful change is detected, the system must determine whether recalibration, incremental updating, or complete retraining is the most appropriate response.

Incremental learning provides a promising mechanism for this purpose [38]. However, unrestricted online adaptation brings its own risks, because a model may learn from erroneous measurements, transient disturbances, or incorrect

labels. An important research direction is therefore *controlled adaptation*, in which drift detection, model updating, validation, and human approval are treated as parts of a single lifecycle. At the same time, every operational decision should remain traceable to the model version and the data on which it was based.

C. From Prototypes to Maintainable and Measurable Systems

Another important gap lies in the transition from successful experiments to sustainable production deployment. Algorithmic performance is only one component of a production ML system. In practice, the surrounding data pipelines, integration mechanisms, and monitoring infrastructure can dominate long-term costs [75]. Machine learning operations (MLOps) practices address this broader lifecycle, but their adaptation to shop-floor and edge environments remains limited [76].

Edge deployments such as UC03 illustrate how memory, processing capacity, and energy consumption can restrict the models that are practical to deploy. UC07 reveals another side of the same challenge. Automated annotation substantially improved the computer-vision pipeline, yet the hardware and labeling effort required across numerous product variants made full deployment economically unattractive. An ML method may therefore be technically effective without being operationally scalable. Such negative results are scientifically valuable and deserve to be reported more systematically.

The use cases also report a diverse range of outcomes, from classification accuracy to energy savings, OEE improvements, and indicators of human acceptance. This diversity makes direct comparison difficult. Future studies should therefore distinguish between three levels of evaluation. *Model-level* metrics describe predictive or detection performance. *System-level* metrics capture factors such as latency, resource consumption, and maintenance effort. Finally, *industrial-impact* metrics measure effects on cost, downtime, quality, and human decision making. Temporal, cross-machine, and cross-site validation should complement random hold-out tests, which primarily demonstrate that a model can interpolate within the distribution represented by its training data.

These considerations support a broader move toward *resource-aware and frugal industrial ML*. In some applications, a slightly less accurate decision tree or linear model may create more industrial value than a complex alternative. If it can be deployed, inspected, updated, and maintained at substantially lower cost, the simpler model may ultimately be the more practical solution.

D. Explainability, Uncertainty, and Human Oversight

Interpretability becomes especially important when ML outputs influence maintenance, production planning, or human work. Traditional models are often comparatively easy to inspect, whereas post-hoc techniques such as SHAP, DIFFI, and AcME-AD can provide explanations for more complex predictors and anomaly detectors [6], [77]. In the pipeline of Fig. 2, the explainer therefore serves as the link between the learner and the operator.

The AIMS5.0 use cases indicate, however, that simply providing an explanation does not guarantee effective human–AI cooperation. In UC12, decision makers relied less on the transparent presentation of the model than on the black-box version, with a weight-on-advice of 0.58 versus 0.63. UC15 illustrates a different situation, in which anomaly explanations become valuable because they help operators identify the sensor variables responsible for an alert and investigate the underlying process.

This highlights an important gap in current XAI evaluation. Explanations are often assessed as properties of the model interface, whereas Industry 5.0 requires their influence on the *human decision process* to be evaluated [78]. Future studies should examine whether an explanation improves diagnostic accuracy, reduces investigation time, or helps users recognize incorrect predictions. The desired outcome is appropriate reliance: operators should follow the model when its recommendation is sound and override it when it is not [79].

Uncertainty should likewise become an explicit component of decision support. A model should be able to indicate when the available data are insufficient or differ substantially from its training experience. Low-confidence cases can then be escalated to human experts rather than forcing an automated decision. Explainability, uncertainty estimation, and abstention should therefore be studied together. From a governance perspective, traceability, human oversight, and robustness are also becoming increasingly important regulatory requirements [57]. The research challenge is to embed these properties technically within ML pipelines instead of treating compliance as documentation added after development.

E. From Prediction to Multi-Objective Decision Support

Many industrial ML applications ultimately revolve around decisions rather than predictions (cf. the *decide* branch of Fig. 1). Predicting a machine failure, energy demand, or WIP level has limited value unless that prediction can be translated into an appropriate operational response.

Several use cases demonstrate that such decisions involve competing objectives. UC09 must balance production performance against electricity cost. UC19 considers service level alongside inventory and planning stability, while UC20 weighs quality assurance against the cost of test wafers. Similar tensions can arise between productivity and other concerns, such as worker safety or equipment lifetime.

Predictive accuracy, however, is typically represented by a single statistical metric. Future work should therefore connect ML more systematically with multi-objective, stochastic, and constraint-based optimization. Predictions should not, in particular, be treated as exact inputs to an optimizer. Their uncertainty should propagate into the decision layer, and models may even be trained directly to improve decision quality rather than to minimize prediction error [80].

F. Hybrid Models and ML-Enhanced Digital Twins

Digital twins provide a controlled environment in which alternative decisions can be evaluated before they are applied to the physical process [66], [67]. ML can complement these

environments in several ways. It can estimate parameters that cannot be measured directly, construct fast surrogate models, and detect discrepancies between expected and observed behavior. Within AIMS5.0, UC07 uses discrete-event simulation to evaluate scheduling strategies, while UC09 integrates energy forecasting and scheduling into a digital representation of fab operations.

A particularly relevant research direction is the development of *hybrid industrial models* that combine first-principles knowledge with data-driven learning [81]. Purely data-driven models may struggle when confronted with abnormal conditions they have never observed, whereas purely physical models can be computationally expensive or incomplete. Hybrid approaches offer a middle ground: they can reduce training-data requirements while preserving a physically meaningful structure. Digital twins can also provide safe test beds for reinforcement learning and adaptive scheduling [82]. This potential, however, depends on continuously validating that the twin remains representative of the physical process it is intended to mirror.

G. Cross-Site Learning and Data Sovereignty

Many industrial organizations operate fleets of similar machines or maintain multiple production sites. Learning jointly from these environments could improve model robustness, particularly when failures are rare. Yet industrial data cannot always be centralized because of intellectual-property concerns, confidentiality requirements, or organizational boundaries. Federated learning offers a way to train models without transferring raw data [83], although its applicability to strongly heterogeneous industrial sites remains an open research question. For traditional ML, simpler approaches to knowledge reuse may also be valuable. Engineered features, anomaly thresholds, model parameters, or domain rules can potentially be transferred without moving complete datasets. UC20's OneVirtualFab concept illustrates the strategic importance of this type of cross-site harmonization.

H. Research Agenda

Based on the challenges discussed above, the following questions define a practical research agenda for machine learning in Industry 5.0:

- 1) **Data-efficient learning:** How can useful models be trained when failures are rare and labels are expensive?
- 2) **Transfer and adaptation:** How can models remain reliable across machines, sites, and changing conditions without complete retraining?
- 3) **Lifecycle-aware ML:** How should drift detection, re-training, and versioning be integrated into maintainable industrial pipelines?
- 4) **Human–ML cooperation:** Which explanations and uncertainty cues help operators recognize when a recommendation should not be followed?
- 5) **Decision-oriented ML:** How can prediction uncertainty be incorporated into scheduling and optimization?

- 6) **Hybrid modeling:** How can physical knowledge and digital twins be combined with data-driven models to improve generalization?
- 7) **Resource-aware ML:** How should model complexity be balanced against latency, energy, and maintainability?
- 8) **Comparable validation:** Which evaluation protocols demonstrate benefits beyond a single dataset or pilot installation?

Overall, the AIMS5.0 use cases suggest that the next stage of industrial ML will depend less on developing ever more complex algorithms and more on building learning systems that are *adaptable, maintainable, interpretable, and operationally measurable*. Traditional ML is well positioned to contribute to this transition, particularly when it is combined with domain knowledge and human expertise. The emphasis therefore shifts from isolated model performance to dependable performance across the full industrial lifecycle.

V. CONCLUSIONS

This paper examined the role of machine learning in the transition from Industry 4.0 to Industry 5.0. Particular attention was given to the continued relevance of traditional ML methods in industrial environments. Industry 4.0 established the foundations for connected, data-driven, and increasingly automated production. Industry 5.0 builds on these foundations, but shifts the focus toward how technology can support people, sustainability, and resilience.

The paper mapped the traditional ML methods used in 13 AIMS5.0 use cases (Table I), explained the concepts behind each method family, and condensed the observed selection logic and human-in-the-loop structure into Figs. 1 and 2. Section ?? derived a research agenda from the recurring difficulties.

The reviewed literature and the AIMS5.0 use cases show that established ML techniques remain highly relevant. These include regression methods, decision trees, random forests, support vector machines, clustering, anomaly detection, and time-series models. Their strengths become especially clear when industrial data are structured, datasets are limited, or computing resources are constrained. They are also attractive when model behaviour needs to be understandable. In such situations, a more complex model is not automatically a better industrial solution. Table II summarized the abbreviations used throughout the paper.

The use cases also show that ML rarely creates value as an isolated predictive component. Its outputs are most useful when they become part of a broader operational workflow. ML can support predictive maintenance, quality monitoring, production planning, energy management, anomaly detection, and process improvement. In practice, these models are often combined with optimization methods, simulation tools, expert knowledge, or edge-computing platforms. This creates a layered decision process in which different technologies play different roles. ML extracts useful information from data. Other components determine how that information is translated into action.

One of the clearest lessons from the investigated cases is that predictive performance alone is not enough. A model

may achieve excellent accuracy and still be difficult to deploy. Industrial environments introduce challenges that are often absent from benchmark datasets. Labels may be scarce, operating conditions may change, and machines may behave differently across sites. Integration and maintenance can also be costly. These factors can determine whether a technically successful prototype survives beyond the pilot stage.

For this reason, industrial ML should be evaluated more broadly. Accuracy, precision, recall, or similar model-level metrics remain important, but they tell only part of the story. A practical ML system must also be robust, scalable, and maintainable. Its computational demands must fit the deployment environment. Most importantly, it should produce measurable operational value.

The AIMS5.0 use cases reinforce this point. Some applications demonstrated strong technical performance but also revealed economic or deployment constraints. Such outcomes are valuable rather than disappointing. They expose the distance between a successful experiment and a sustainable industrial solution. They also show why large-scale adoption cannot be justified by predictive accuracy alone.

Human involvement is equally important. Industry 5.0 does not simply replace human expertise with automated decisions. Instead, ML should strengthen the ability of operators, engineers, and planners to understand what is happening and decide what to do next. Explainability can support this process, but explanation alone is not enough. More transparency does not automatically lead to greater trust, nor does it guarantee better decisions.

A more useful goal is therefore appropriate human reliance. ML systems should help users understand when a recommendation is well supported and when caution is needed. They should also make uncertainty visible where possible. In high-impact settings, experts should be able to question, validate, and, when necessary, override model outputs. This is where human-machine cooperation becomes more than a design principle: it becomes part of system reliability.

Looking ahead, the future of ML in Industry 5.0 is likely to depend less on increasingly complex standalone models and more on dependable learning systems. These systems will need to work with limited data, adapt to changing conditions, and operate across machines and sites. They will also need to fit within existing industrial infrastructures rather than around them.

Several research directions are especially important. Continual and incremental learning can help models adapt over time. Hybrid approaches can combine data-driven learning with physical knowledge and simulation. Resource-aware methods can support efficient deployment at the edge. Cross-site learning can improve knowledge transfer between factories. Closer integration with optimization can also help convert predictions into more robust decisions.

Traditional machine-learning methods are therefore likely to remain an important part of the Industry 5.0 landscape. Their relative simplicity is not a weakness. In many industrial applications, it is an advantage. These methods can be efficient, interpretable, and well suited to structured process data. They

can also be easier to validate and maintain than more complex alternatives.

The key challenge is no longer simply to build models that perform well in controlled experiments. The real test is whether those models continue to create value once they enter the factory floor. For Industry 5.0, successful ML will be defined not only by what it can predict, but by how reliably it can support people, processes, and decisions under real operating conditions.

ACKNOWLEDGMENT

The research leading to these results was conducted in the AIMS5.0 project, which received funding from the Chips Joint Undertaking under grant agreement No. 101112089 and from the partners' national funding programmes.

REFERENCES

- [1] A. Renda, S. Schwaag Serger, D. Tataj, A. Morlet, D. Isaksson, F. Martins, M. Mir Roca, C. Hidalgo, A. Huang, S. Dixon-Declève, P.-A. Bolland, F. Bria, C. Charveriat, K. Dunlop, and E. Giovannini, "Industry 5.0, a transformative vision for Europe: Governing systemic transformations towards a sustainable industry," European Commission, Directorate-General for Research and Innovation, Luxembourg, ESIR Policy Brief 3, 2022, catalogue number KI-BE-21-005-EN-N. [Online]. Available: https://research-and-innovation.ec.europa.eu/knowledge-publications-tools-and-data/publications/all-publications/industry-50-transformative-vision-europe_en
- [2] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. New York, NY, USA: Springer, 2009.
- [3] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.
- [4] T. Wuest, D. Weimer, C. Irgens, and K.-D. Thoben, "Machine learning in manufacturing: advantages, challenges, and applications," *Production & Manufacturing Research*, vol. 4, no. 1, pp. 23–45, 2016.
- [5] A. Presciuttini, A. Cantini, F. Costa, and A. Portioli-Staudacher, "Machine learning applications on iot data in manufacturing operations and their interpretability implications: A systematic literature review," *Journal of Manufacturing Systems*, vol. 74, pp. 477–486, 2024. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0278612524000773>
- [6] P. Linardatos, V. Papastefanopoulos, and S. Kotsiantis, "Explainable ai: A review of machine learning interpretability methods," *Entropy*, vol. 23, no. 1, 2021. [Online]. Available: <https://www.mdpi.com/1099-4300/23/1/18>
- [7] Z. Ge, Z. Song, S. X. Ding, and B. Huang, "Data mining and analytics in the process industry: The role of machine learning," *IEEE Access*, vol. 5, pp. 20 590–20 616, 2017.
- [8] X. Xu, Y. Lu, B. Vogel-Heuser, and L. Wang, "Industry 4.0 and industry 5.0— inception, conception and perception," *Journal of Manufacturing Systems*, vol. 61, pp. 530–535, 2021.
- [9] T. P. Carvalho, F. A. A. M. N. Soares, R. Vita, R. d. P. Francisco, J. P. Basto, and S. G. S. Alcalá, "A systematic literature review of machine learning methods applied to predictive maintenance," *Computers & Industrial Engineering*, vol. 137, p. 106 024, 2019.
- [10] A. Widodo and B.-S. Yang, "Support vector machine in machine condition monitoring and fault diagnosis," *Mechanical Systems and Signal Processing*, vol. 21, no. 6, pp. 2560–2574, 2007.
- [11] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, 2010.
- [12] S. J. Qin, "Survey on data-driven industrial process monitoring and diagnosis," *Annual Reviews in Control*, vol. 36, no. 2, pp. 220–234, 2012.
- [13] J. E. van Engelen and H. H. Hoos, "A survey on semi-supervised learning," *Machine Learning*, vol. 109, no. 2, pp. 373–440, 2020.
- [14] B. Settles, "Active learning literature survey," University of Wisconsin–Madison, Department of Computer Sciences, Computer Sciences Technical Report 1648, 2009.
- [15] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.

TABLE II
ABBREVIATIONS USED THROUGHOUT THE PAPER.

Abbreviation	Description
2SP	Two-Stage Stochastic Programming
AcME-AD	Accelerated Model Explanations for Anomaly Detection
AI	Artificial Intelligence
AIMS5.0	Artificial Intelligence in Manufacturing leading to Sustainability and Industry 5.0
AMR	Autonomous Mobile Robot
AuDi-EC	Automated Digitized Equipment Control
B-ALIF	Bayesian Active Learning Isolation Forest
CART	Classification and Regression Trees
CCP	Chance-Constrained Programming
CP	Constraint Programming
CPU	Central Processing Unit
CRM	Causal Rule Mining
CSP	Common Spatial Pattern
CV	Computer Vision
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DES	Discrete-Event Simulation
DIFFI	Depth-based Isolation Forest Feature Importance
DNN	Deep Neural Network
DRO	Digital Reference Ontology
DT	Digital Twin
EEG	Electroencephalography
ESPAR	Electronically Steerable Parasitic Array Radiator
EU	European Union
FFT	Fast Fourier Transform
FOUP	Front Opening Unified Pod
GBDT	Gradient-Boosted Decision Tree
GP	Genetic Programming
HDF5	Hierarchical Data Format Version 5
IC	Integrated Circuit
ICP-MS	Inductively Coupled Plasma Mass Spectrometry
IoT	Internet of Things
IoU	Intersection over Union
ISO	International Organization for Standardization
KD-tree	k-Dimensional Tree
KNN	K-Nearest Neighbors
KPI	Key Performance Indicator
LiDAR	Light Detection and Ranging
MDP	Markov Decision Process
MEMS	Micro-Electro-Mechanical Systems
ML	Machine Learning
NC	Numerical Control
OEE	Overall Equipment Effectiveness
OEM	Original Equipment Manufacturer
OHT	Overhead Transportation
PCA	Principal Component Analysis
RF	Radio Frequency
RL	Reinforcement Learning
ROC	Receiver Operating Characteristic
SARIMAX	Seasonal Autoregressive Integrated Moving Average with Exogenous Regressors
SATCOM	Satellite Communications
SAW	Surface Acoustic Wave
SAX	Symbolic Aggregate Approximation
SHAP	Shapley Additive Explanations
SLA	Service Level Agreement
SNIR	Signal-to-Noise and Interference Ratio
SPC	Statistical Process Control
SPSA	Simultaneous Perturbation Stochastic Approximation
SVM	Support Vector Machine
SVR	Support Vector Regression
TinyML	Tiny Machine Learning
ToF	Time of Flight
TRL	Technology Readiness Level
UC	Use Case
WIP	Work in Progress
XAI	Explainable Artificial Intelligence
XGBoost	Extreme Gradient Boosting

- [16] M. Panzer and B. Bender, "Deep reinforcement learning in production systems: a systematic literature review," *International Journal of Production Research*, vol. 60, no. 13, pp. 4316–4341, 2022.
- [17] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*, 2nd ed. Berlin, Germany: Springer, 2015.
- [18] J. Branke, S. Nguyen, C. W. Pickardt, and M. Zhang, "Automated design of production scheduling heuristics: A review," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 1, pp. 110–124, 2016.
- [19] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. Belmont, CA, USA: Wadsworth, 1984.
- [20] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [21] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *The Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.
- [22] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD)*, 2016, pp. 785–794.
- [23] T. M. Cover and P. E. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.
- [24] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [25] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. Knowledge Discovery and Data Mining (KDD)*, 1996, pp. 226–231.
- [26] R. B. Rusu and S. Cousins, "3d is here: Point cloud library (pcl)," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 1–4.
- [27] I. T. Jolliffe and J. Cadima, "Principal component analysis: a review and recent developments," *Philosophical Transactions of the Royal Society A*, vol. 374, no. 2065, p. 20 150 202, 2016.
- [28] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys*, vol. 41, no. 3, pp. 15:1–15:58, 2009.
- [29] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proc. 8th IEEE Int. Conf. Data Mining (ICDM)*, 2008, pp. 413–422.
- [30] D. Sartor, T. Barbariol, and G. A. Susto, "Bayesian active learning isolation forest (B-ALIF): A weakly supervised strategy for anomaly detection," *Engineering Applications of Artificial Intelligence*, vol. 130, p. 107 671, 2024.
- [31] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA, USA: MIT Press, 1992.
- [32] F. Rossi, P. van Beek, and T. Walsh, Eds., *Handbook of Constraint Programming*, ser. Foundations of Artificial Intelligence. Amsterdam, The Netherlands: Elsevier, 2006.
- [33] J. R. Birge and F. Louveaux, *Introduction to Stochastic Programming*, 2nd ed. New York, NY, USA: Springer, 2011.
- [34] A. Charnes and W. W. Cooper, "Chance-constrained programming," *Management Science*, vol. 6, no. 1, pp. 73–79, 1959.
- [35] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
- [36] J. C. Spall, "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation," *IEEE Transactions on Automatic Control*, vol. 37, no. 3, pp. 332–341, 1992.
- [37] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. New York, NY, USA: Wiley, 1994.
- [38] V. Losing, B. Hammer, and H. Wersing, "Incremental on-line learning: A review and comparison of state of the art algorithms," *Neurocomputing*, vol. 275, pp. 1261–1274, 2018.
- [39] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time Series Analysis: Forecasting and Control*, 5th ed. Hoboken, NJ, USA: Wiley, 2015.
- [40] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Mathematics of Computation*, vol. 19, no. 90, pp. 297–301, 1965.
- [41] R. B. Randall, *Vibration-based Condition Monitoring: Industrial, Aerospace and Automotive Applications*. Chichester, UK: Wiley, 2011.
- [42] C.-C. M. Yeh, Y. Zhu, L. Ulanova, N. Begum, Y. Ding, H. A. Dau, D. F. Silva, A. Mueen, and E. Keogh, "Matrix profile I: All pairs similarity joins for time series: A unifying view that includes motifs, discords and shapelets," in *Proc. 16th IEEE Int. Conf. Data Mining (ICDM)*, 2016, pp. 1317–1322.
- [43] C. Truong, L. Oudre, and N. Vayatis, "Selective review of offline change point detection methods," *Signal Processing*, vol. 167, p. 107 299, 2020.
- [44] J. Lin, E. Keogh, L. Wei, and S. Lonardi, "Experiencing SAX: a novel symbolic representation of time series," *Data Mining and Knowledge Discovery*, vol. 15, no. 2, pp. 107–144, 2007.
- [45] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.
- [46] H. Ramoser, J. Müller-Gerking, and G. Pfurtscheller, "Optimal spatial filtering of single trial EEG during imagined hand movement," *IEEE Transactions on Rehabilitation Engineering*, vol. 8, no. 4, pp. 441–446, 2000.
- [47] B. Blankertz, R. Tomioka, S. Lemm, M. Kawanabe, and K.-R. Müller, "Optimizing spatial filters for robust EEG single-trial analysis," *IEEE Signal Processing Magazine*, vol. 25, no. 1, pp. 41–56, 2008.
- [48] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems 30 (NeurIPS)*, 2017, pp. 4765–4774.
- [49] M. Carletti, M. Terzi, and G. A. Susto, "Interpretable anomaly detection with DIFFI: Depth-based feature importance of isolation forest," *Engineering Applications of Artificial Intelligence*, vol. 119, p. 105 730, 2023.
- [50] D. Dandolo, C. Masiero, M. Carletti, D. Dalle Pezze, and G. A. Susto, "AcME—accelerated model-agnostic explanations: Fast whitening of the machine-learning black box," *Expert Systems with Applications*, vol. 214, p. 119 115, 2023.
- [51] V. Zaccaria, D. Dandolo, C. Masiero, and G. A. Susto, "Acmead: Accelerated model explanations for anomaly detection," 2024. [Online]. Available: <https://arxiv.org/abs/2403.01245>
- [52] J. Li, T. D. Le, L. Liu, J. Liu, Z. Jin, B. Sun, and S. Ma, "From observational studies to causal rule mining," *ACM Transactions on Intelligent Systems and Technology*, vol. 7, no. 2, pp. 14:1–14:27, 2015.
- [53] D. C. Montgomery, *Introduction to Statistical Quality Control*, 8th ed. Hoboken, NJ, USA: Wiley, 2019.
- [54] A. M. Law, *Simulation Modeling and Analysis*, 5th ed. New York, NY, USA: McGraw-Hill, 2015.
- [55] O. Myapati, A. Mukherjee, D. Mishra, S. K. Pal, P. P. Chakrabarti, and A. Pal, "A critical review on applications of artificial intelligence in manufacturing," *Artificial Intelligence Review*, vol. 56, no. 1, pp. 661–768, Oct 2023. [Online]. Available: <https://doi.org/10.1007/s10462-023-10535-y>
- [56] J. Leng, X. Zhu, Z. Huang, X. Li, P. Zheng, X. Zhou, D. Mourtzis, B. Wang, Q. Qi, H. Shao, J. Wan, X. Chen, L. Wang, and Q. Liu, "Unlocking the power of industrial artificial intelligence towards industry 5.0: Insights, pathways, and challenges," *Journal of Manufacturing Systems*, vol. 73, pp. 349–363, 2024. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0278612524000335>
- [57] European Data Protection Supervisor, *AI Act Regulation (EU) 2024/1689 – Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence and amending Regulations (EC) No 300/2008, (EU) No 167/2013, (EU) No 168/2013, (EU) 2018/858, (EU) 2018/1139 and (EU) 2019/2144 and Directives 2014/90/EU, (EU) 2016/797 and (EU) 2020/1828 (Artificial Intelligence Act) (Text with EEA relevance)*. Publications Office of the European Union, 2025.
- [58] T. Czimmermann, G. Ciuti, M. Milazzo, M. Chiurazzi, S. Roccella, C. M. Oddo, and P. Dario, "Visual-based defect detection and classification approaches for industrial applications—a SURVEY," *Sensors*, vol. 20, no. 5, p. 1459, 2020.
- [59] F. Semeraro, A. Griffiths, and A. Cangelosi, "Human–robot collaboration and machine learning: A systematic review of recent research," *Robotics and Computer-Integrated Manufacturing*, vol. 79, p. 102 432, 2023.
- [60] D. Weichert, P. Link, A. Stoll, S. Rüping, S. Ihlenfeldt, and S. Wrobel, "A review of machine learning for the optimization of production processes," *The International Journal of Advanced Manufacturing Technology*, vol. 104, pp. 1889–1902, 2019.
- [61] D. Ni, Z. Xiao, and M. K. Lim, "A systematic review of the research trends of machine learning in supply chain management," *International Journal of Machine Learning and Cybernetics*, vol. 11, pp. 1463–1482, 2020.
- [62] G. Baryannis, S. Validi, S. Dani, and G. Antoniou, "Supply chain risk management and artificial intelligence: state of the art and future research directions," *International Journal of Production Research*, vol. 57, no. 7, pp. 2179–2202, 2019.

- [63] D. A. C. Narciso and F. G. Martins, "Application of machine learning tools for energy efficiency in industry: A review," *Energy Reports*, vol. 6, pp. 1181–1199, 2020.
- [64] A. Jamwal, R. Agrawal, and M. Sharma, "Machine learning applications for sustainable manufacturing: a bibliometric-based review for future research," *Journal of Enterprise Information Management*, vol. 35, no. 2, pp. 566–596, 2022.
- [65] H. Ravichandar, A. S. Polydoros, S. Chernova, and A. Billard, "Recent advances in robot learning from demonstration," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, pp. 297–330, 2020.
- [66] F. Tao, H. Zhang, A. Liu, and A. Y. C. Nee, "Digital twin in industry: State-of-the-art," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2405–2415, 2019.
- [67] M. M. Rathore, S. A. Shah, D. Shukla, E. Bentafat, and S. Bakiras, "The role of AI, machine learning, and big data in digital twinning: A systematic literature review, challenges, and opportunities," *IEEE Access*, vol. 9, pp. 32 030–32 052, 2021.
- [68] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press, 2006.
- [69] European Commission. (2023) Artificial intelligence in manufacturing leading to sustainability and Industry5.0 (AIMS5.0), grant agreement no. 101112089. EU Research Results. HORIZON-KDT-JU-2022-1-IA. Accessed: Sep. 29, 2026. [Online]. Available: <https://aims50.eu/>
- [70] Stream Analyze AB. (2026) Stream analyze platform: The end-to-end edge AI platform. Stream Analyze AB. Uppsala, Sweden. Accessed: Aug. 29, 2026. [Online]. Available: <https://www.streamanalyze.com/>
- [71] P. P. Ray, "A review on TinyML: State-of-the-art and prospects," *Journal of King Saud University – Computer and Information Sciences*, vol. 34, no. 4, pp. 1595–1623, 2022.
- [72] L. Nunes, R. Marcuzzi, J. Behley, and C. Stachniss, "Towards generating realistic 3d semantic training data for autonomous driving," 2026. [Online]. Available: <https://arxiv.org/abs/2503.21449>
- [73] G. Shao et al., "An analysis of the new ISO 23247 series of standards on digital twin framework for manufacturing," *NIST, Tech. Rep.*, 2023. [Online]. Available: <https://www.nist.gov/publications/analysis-new-iso-23247-series-standards-digital-twin-framework-manufacturing>
- [74] J. Gama, I. Žliobait'e, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Computing Surveys*, vol. 46, no. 4, pp. 44:1–44:37, 2014.
- [75] D. Sculley, G. Holt, D. Golovin, E. Davydov, T. Phillips, D. Ebner, V. Chaudhary, M. Young, J.-F. Crespo, and D. Dennison, "Hidden technical debt in machine learning systems," in *Advances in Neural Information Processing Systems 28 (NIPS)*, 2015, pp. 2503–2511.
- [76] D. Kreuzberger, N. Kühl, and S. Hirschl, "Machine learning operations (MLOps): Overview, definition, and architecture," *IEEE Access*, vol. 11, pp. 31 866–31 879, 2023.
- [77] S. Moosavi, M. Zanjani, R. Razavi-Far, V. Palade, and M. Saif, "Explainable ai in manufacturing and industrial cyber–physical systems: A survey," *Electronics*, vol. 13, p. 3497, 09 2024.
- [78] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," arXiv:1702.08608, 2017.
- [79] J. D. Lee and K. A. See, "Trust in automation: Designing for appropriate reliance," *Human Factors*, vol. 46, no. 1, pp. 50–80, 2004.
- [80] A. N. Elmachoub and P. Grigas, "Smart "predict, then optimize"," *Management Science*, vol. 68, no. 1, pp. 9–26, 2022.
- [81] L. von Rueden, S. Mayer, K. Beckh, B. Georgiev, S. Giesselbach, R. Heese, B. Kirsch, J. Pfrommer, A. Pick, R. Ramamurthy, M. Walczak, J. Garcke, C. Bauckhage, and J. Schuecker, "Informed machine learning – a taxonomy and survey of integrating prior knowledge into learning systems," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 1, pp. 614–633, 2023.
- [82] H. Cao, H. Söderlund, Q. Fang, S. Chen, L. Erdal, A. Gubartalla, P. V. Lopes, G. Shao, P. Lönnehed, H. Putto, A. Ahmed, S. Ekered, and B. Johansson, "Towards ai-based sustainable and xr-based human-centric manufacturing: Implementation of iso 23247 for digital twins of production systems," 2025. [Online]. Available: <https://arxiv.org/abs/2508.14580>
- [83] P. Kairouz, H. B. McMahan et al., "Advances and open problems in federated learning," *Foundations and Trends in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

Image Processing: a Review on Industrial Applications

Michał Affek¹, Manuel Barusco², Diāna Dupļevska³, Kārlis Freivalds³, Gergely Hollósi⁴, Corinna Kofler⁵,
Michał Kopczyński¹, Ruba Almahasneh⁴, Philip Newton⁶, Guntis V. Strazds^{3,7}, Gian Antonio Susto⁸,
Gábor Szűcs⁴, and Marek S. Tatara¹

Abstract—Image processing has become a key technology in industrial automation, supporting quality control, robotic manipulation, and process monitoring. However, deploying deep learning-based vision systems on the factory floor remains challenging. Industrial datasets are often small, imbalanced, and expensive to annotate, while production environments differ substantially from the everyday imagery used to pretrain most vision models. This paper reviews recent advances in industrial image processing and summarizes practical experience from several use cases developed within the AIMS5.0 project. We examine methods for synthetic data generation, multi-camera tracking, and continual learning, with particular attention to their role in data-scarce and evolving production settings. The presented applications cover robotic manipulation, quality inspection, assembly monitoring, and data annotation. For robotic grasping, a hybrid pose estimation pipeline combines AI-based detection and segmentation with Iterative Closest Point fitting. It achieves accuracy comparable to state-of-the-art approaches while requiring only an object point cloud instead of a textured CAD model and offers substantially faster execution. A segmentation-based method for bottle socket detection also provides promising localization accuracy, although further improvements are needed to meet strict mechanical tolerances. For industrial inspection, we show that class balancing, moderate augmentation, and transfer learning improve semiconductor defect classification under severe class imbalance. We further introduce a CycleGAN-based domain adaptation approach that is effective when source and target domains differ in their label distributions. In aluminium corrosion detection, supervised segmentation clearly outperforms unsupervised anomaly detection. Experiments on luminaire assembly show that video action recognition performs well on familiar products but generalizes poorly to visually different variants, illustrating the limitations of vision-only approaches in high-mix, low-volume production. Finally, an iterative pseudo-labeling pipeline substantially improves object detection from a small initial set of annotated images. Based on these results, we identify several open challenges in this area, including data requirements, extreme class imbalance, reliable pseudolabel confidence estimation, automatic image-quality filtering, and scalable human-in-the-loop annotation.

Index Terms—Image processing, deep learning, industry 5.0, data annotation

¹ Faculty of Faculty of Electronics, Telecommunications and Informatics, Gdańsk University of Technology, Poland.

² Department of Information Engineering, University of Padova, Italy.

³ Institute of Electronics and Computer Science, Riga, Latvia.

⁴ Department of Telecommunications and Artificial Intelligence, Budapest University of Technology and Economics, Hungary.

⁵ KAI Kompetenzzentrum Automobil- und Industrieelektronik GmbH, Villach, Austria.

⁶ Signify, Eindhoven, The Netherlands.

⁷ University of Latvia, Latvia.

⁸ Department of Information Engineering, University of Padova, Italy.

I. INTRODUCTION

Image processing plays a crucial role in various industrial applications, enhancing efficiency, quality, and automation across different sectors. Categorizing the image processing tasks based on specific criteria helps optimize performance and decision-making. The speed of image processing is critical in industries such as manufacturing and autonomous vehicles, where real-time error detection or object recognition ensures operational efficiency and safety [1]. Real-time processing is particularly important in applications such as surveillance and autonomous driving, where delays can compromise results, while non-real-time processing is sufficient for tasks such as offline medical diagnostics [2] or historical data analysis. Also, in terms of risk [3], there are multiple image processing systems, where an important aim is to minimize the likelihood of wrong decisions. Finally, the magnitude of the impact of decisions varies across industries; for example, in medical imaging, a misclassification can have life-threatening consequences, while minor errors in e-commerce product labeling can only affect the user experience. Together, these factors influence the choice of algorithms, infrastructure, and execution strategies for industrial image processing systems [4].

The integration of image processing into real-time monitoring systems has transformed manufacturing where algorithms capable of edge detection and morphological operations enable immediate feedback on surface quality and defect detection, facilitating timely interventions like tool replacements. This capability not only improves the quality of produced items but also optimizes operational efficiency. One of the proven non-contact methods of Tool Condition Monitoring (TCM) is measuring systems based on image processing technologies that allow assessing the current state of the machined surface and the quantitative indicators of tool wear [5].

In the medical field, deep learning techniques have revolutionized image segmentation processes. These methods allow for precise analysis of medical images, significantly aiding in diagnostics and treatment planning [6].

Image processing helps many researchers around the globe to enhance their models in the agriculture domain. One of the possible tasks is plant classification where the aim is to classify images of plants to species, which is solved by deep learning, e.g. with a pretrained convolutional neural network and transfer learning [7].

Remote sensing is a critical field that involves detecting and

tracking the physical properties of an environment (typically from satellite or aircraft) by the calculation of reflected and emitted radiation from a radius. Many researchers around the world use image data augmentation to improve their models in the remote sensing domain [8].

Hyperspectral imaging poses major challenges for supervised classification methods due to the high dimensionality of the data; and there is an additional challenge to discover and distinguish novel categories for hyperspectral images in scenarios where some classes are unknown [9].

Object pose estimation is a key task in image processing, especially for robotic systems that need to perceive and interact with their environment. It involves determining an object's six degrees of freedom (6-DoF) pose – its 3D position and orientation – from visual input. Precise pose estimation is critical in industrial settings, enabling accurate manipulation and automation. While state-of-the-art approaches often focus on model-free methods requiring little or no prior object knowledge, this doesn't align well with industrial reality. In many industrial applications, high-quality Computer-Aided Design (CAD) models are readily available or easily obtained, and depth sensing is standard alongside color imaging. These practical advantages should be leveraged to develop more efficient, robust, and simpler pose estimation algorithms tailored for industrial use.

Automated image processing systems utilize deep learning algorithms to identify defects with high accuracy in semiconductor manufacturing. This area is characterized by high data complexity and data scarcity and neural networks can be pre-trained on industry-specific data, significantly reducing the amount of sample images required for effective training. This capability allows for real-time quality assurance that rivals human inspectors in accuracy while maintaining industrial speed standards. The state of the art in industrial image processing is characterized by sophisticated AI-driven solutions that enhance quality control, optimize production processes, and support predictive maintenance strategies, all contributing to more efficient manufacturing environments [10, 11].

The landscape of deep learning, particularly in computer vision, is rapidly evolving with the interplay between Convolutional Neural Networks (CNNs) and Transformers. Transformers have emerged as a dominant architecture, especially in Natural Language Processing (NLP) and increasingly in computer vision tasks. Their ability to handle global relationships through self-attention mechanisms has led to significant advancements, outperforming CNNs in various benchmarks for tasks like object detection and segmentation.

Vision transformers have become popular as a possible substitute to CNNs, with their ability to focus on global relationships in images, offer large learning capacity; however, they may suffer from limited generalization. Recently, in vision transformers hybridization of both the convolution operation and self-attention mechanism has emerged, to exploit both the local and global image representations. These hybrid vision transformers, also referred to as CNN-Transformer architectures, have demonstrated remarkable results. While transformers are making significant strides in various domains, the integration of CNNs into hybrid models represents a

promising avenue for enhancing performance in computer vision tasks [12].

Thus, image processing is integral to Industry 5.0, supporting smart manufacturing through automation and real-time data analysis. Combining image processing with robotics and IoT can further enhance industrial automation. In the next sections, applications of image processing in the industrial automation will be presented; these applications continue to expand with advancements in technology and integration with other industrial systems.

A. Synthetic Datasets for Vision

Synthetic datasets have become an increasingly valuable resource in machine learning, particularly in scenarios where collecting and annotating large-scale real-world data is costly or impractical. By leveraging simulation environments or generative models, researchers can produce diverse, labeled datasets with fine-grained control over variables such as lighting, pose, or background complexity.

Recent progress in generative AI has significantly broadened the toolkit available for synthetic dataset creation. Diffusion models, such as Stable Diffusion [13] and Imagen [14], allow for the generating high-resolution, photorealistic images from textual prompts via iterative denoising processes. These models are particularly effective for producing diverse samples. Text-to-image models, like DALL-E [15], leverage transformer-based architectures to generate coherent visuals directly from textual descriptions. These models are widely used to prototype visual scenarios and create labeled data by combining image synthesis with automated captioning or labeling pipelines. Image-to-image translation models, such as Pix2Pix [16] or ControlNet [17], enable the generation of synthetic variants based on structural or semantic constraints in input images. These are particularly useful in domains like medical imaging or autonomous driving, where realism and controlled variation are critical. For instance, diffusion models have recently been applied to generate synthetic organ-on-a-chip cell images, enhancing dataset diversity for biomedical classification tasks [18, 19]. In parallel, Vision-Language Models (VLMs) such as Flamingo [20] and GPT-4V [21] enable coordinated processing of visual and textual data, supporting both generation and interpretation tasks. These models can be used not only to synthesize data but also to provide annotations, questions, or captions-essentially automating the label generation process. Together, these tools enable scalable, diverse, and cost-effective synthetic data generation. However, they also introduce challenges related to domain shift, realism, and potential bias transfer from its training dataset. One way to reduce this domain shift is to combine visual domain randomization in simulation with a limited number of real images, which was shown to improve the robustness of the learned representations in sim-to-real transfer [22]. Despite these issues, synthetic datasets are increasingly essential in pre-training pipelines, domain adaptation, and robustness testing in computer vision.

In the Table 1, we present a selection of representative techniques and models used for synthetic dataset generation,

categorized by their underlying generation strategy. While this overview includes widely adopted tools across multiple domains, it is by no means exhaustive. The field is rapidly evolving, and many domain-specific, hybrid, or proprietary models continue to emerge, especially in medical imaging, robotics, and industrial inspection.

Evaluating the quality of synthetic datasets is essential to ensure their effectiveness and reliability in various applications. This evaluation encompasses three primary dimensions: fidelity, utility, and privacy. Fidelity assesses how closely synthetic data resembles real data; utility measures the usefulness of synthetic data for specific tasks; and privacy evaluates the risk of sensitive information leakage. Table II summarizes key metrics used to assess these dimensions, providing a comprehensive overview for practitioners and researchers.

B. Multi-Camera Solutions of Computer Vision in Industry

Multi-Camera Multi-Target (MCMT) tracking systems [51] [52] have a wide range of applications across various sectors [53], including general mobility, logistics [54] and industry [55–57]. For vehicle and people mobility, MCMT systems reconstruct complete trajectories, enhancing safety, sustainability, and efficiency. In industrial environments, MCMT enhances logistic operations, streamlines manufacturing processes, and improves worker safety. For the last aim, MCMT systems provide continuous surveillance of worker movements and activities across multiple camera feeds. This capability allows for immediate detection of unsafe behaviors or situations, enabling prompt intervention to prevent accidents in the real-time monitoring environments.

MCMT systems can identify potential hazards in the work environment, such as unsafe machinery operation or workers entering restricted areas [58]. This proactive monitoring helps mitigate risks before they escalate into incidents. In environments where workers and machinery operate in close proximity, MCMT systems can track both human and machine movements to prevent collisions. Alerts can be generated if a worker is detected in a dangerous zone, enhancing situational awareness and safety. The data collected by MCMT systems can be analyzed to identify patterns in worker behavior and safety incidents. This analysis enables organizations to implement targeted training programs and safety protocols based on real-world observations. MCMT systems can be integrated with other safety technologies, such as Internet of Things (IoT) devices that monitor environmental conditions (e.g., air quality, noise levels) and wearable sensors that track vital signs. The implementation of MCMT tracking systems in industrial settings not only enhances real-time safety monitoring but also fosters a culture of proactive risk management, ultimately leading to safer working environments for employees [59].

MCMT systems can be used in Intelligent Transportation Systems (ITS) [60] as well, where it utilizes data and technology to improve transportation networks, with applications that enhance traffic management, safety, and sustainability. MCMT tracking systems can be effectively utilized in logistics to optimize operations and enhance efficiency. As the latest results from the MCMT, a comprehensive vehicle tracking

system is developed that performs multiple tasks simultaneously, including tracking vehicles using a single-phase object detector, re-identifying vehicles, and classifying their attributes such as color and type. While re-identification networks and classification models operate independently, ID collisions may occur across cameras. To address this, MCT-MAC [61], a novel multi-camera trajectory matching algorithm was introduced based on hierarchical agglomerative clustering with time and space constraints, allowing trajectory aggregation across multiple cameras. This approach is flexible, supporting various trackers (ByteTrack and DeepSORT) as well.

MCMT systems can track the movement of goods within warehouses, providing real-time data on inventory levels and locations [54]. By monitoring the flow of both goods and personnel, MCMT can identify bottlenecks in the logistics process. This data allows for the optimization of workflows, ensuring that resources are utilized efficiently and that processes are streamlined. MCMT enables the tracking of vehicles transporting goods across multiple camera views, providing a comprehensive overview of shipment locations. This capability is crucial for maintaining supply chain transparency and improving delivery reliability. The analytics provided by MCMT systems allow logistics managers to make informed decisions based on real-time data regarding movement patterns, resource allocation, and operational efficiency. In automated warehouses, MCMT can work alongside robotic systems to monitor their movements and interactions with human workers, ensuring safe and efficient operations. Overall, the implementation of MCMT systems in industry not only enhances operational efficiency but also contributes to improved safety.

C. Continual Machine Learning

Processes enriched with image processing capabilities and decision-making schemes are frequently deployed in an industrial environment. Despite the automated process's ideation, implementation, and integration stages, there is also a step responsible for long-term management and process service. For Machine Learning (ML) models used in these processes, we can identify methods for partial retraining or updating the model as the dataset grows, which fall under the umbrella term Continual Machine Learning (CML).

The main challenge in building an ML solution in an industrial environment is that the solution does not scale up and thus is not easily reconfigurable without a significant amount of resources [62]. Reconfigurability can relate to the changes between machines, tools, processes, or the changes in the distribution of collected data. Concept drift is a data process that influences many implemented models over time. The continued learning methods were introduced to maintain or enhance the efficiency and reliability of solutions, and simultaneously spend fewer resources on retraining the model with the whole dataset. In principle, the methods aim to treat the data classes as equal without the occurrence of catastrophic forgetting. The mentioned phenomenon occurs when the adaptation to the new data without retraining on the whole dataset tends to lose the ability to capture older classes.

Handling catastrophic forgetting requires careful adoption of the CML techniques. Authors in [63] proposed image

TABLE I
OVERVIEW OF GENERATIVE APPROACHES FOR SYNTHETIC DATA CREATION WITH EXAMPLE MODELS.

Technique	Description	Selected Models / Tools
Text-to-Image	Generates realistic images from natural language prompts. Offers fine control over content, style, and scene composition, making it ideal for augmenting datasets or generating domain-specific samples.	DALL-E [15], Imagen [14], Parti [23], Midjourney [24]
Diffusion Models	Use iterative denoising to produce high-fidelity images from noise. These models are currently state-of-the-art in generative realism and are used both for unconditional and conditional generation tasks.	Stable Diffusion [13], DeepFloyd IF [25], GLIDE [26]
Image-to-Image Translation	Translates an input image from one domain to another (e.g., sketches to photos, semantic maps to realistic scenes). Supports controlled generation in highly structured domains.	Pix2Pix [16], CycleGAN [27], ControlNet [17]
Vision-Language Models (VLMs)	Process and relate visual and textual modalities. Useful for tasks like automatic captioning, label generation, question answering, and scene understanding in multimodal contexts.	Flamingo [20], GPT-4V [21], BLIP-2 [28]
Simulation Engines	Use physically accurate 3D environments to render synthetic scenes with automatic ground-truth labels. Widely used in robotics, automotive, and AR/VR settings for controlled dataset generation.	Unity Perception [29], CARLA [30], Blender [31], NVIDIA Omniverse [32]
3D-to-2D Rendering	Projects 3D assets or scans into annotated 2D images with varied poses, lighting, and camera perspectives. Enables generation of multi-view labeled datasets with consistent geometry.	BlenderProc [33], Meshcapade [34], SURREAL [35]
Semantic-to-Image	Synthesizes realistic images from segmentation masks or semantic layouts. Allows control over scene layout while enabling diversity in style and appearance.	SPADE [36], GauGAN [37]
Prompt-Based Annotation	Uses multimodal models to generate captions, tags, or region descriptions given a prompt. Reduces reliance on manual annotations in early-stage dataset development.	CLIP [38], GPT-4V [21], LLaVA [39]
Automatic Mask Generation	Produces high-quality segmentation masks or bounding boxes using pre-trained models. Useful for initializing label sets in detection and segmentation pipelines.	SAM [40], Grounding DINO [41]
Video Synthesis	Generates temporally coherent video sequences either from text, images, or other conditioning. Supports tasks like motion synthesis, animation, and synthetic video pretraining.	Gen-1 [42], Tune-A-Video [43], Video Diffusion [44]

processing methods such as alpha blending and histogram equalization to mitigate this issue. On the other hand, in the cross-modal image-text retrieval domain, authors in [64] proposed the CML method that utilizes the knowledge decomposition and replay, which imitates the knowledge decomposition and replay mechanisms in humans that are put in challenging environments. Besides the methods, researchers also work on more generalized frameworks. For example, [65] presents a differential geometry framework for functionally invariant paths, enabling flexible adaptation of neural networks for secondary tasks. Simultaneously, authors of the mentioned paper claim that the networks resulting from the application of their framework have increased sparsification of the network and can be portrayed as more robust to adversarial attacks. The authors present a representative example of the usage of CML in industrial applications in [66]. The paper describes the development of the framework for visual classification of the defects and demonstrates the approach to new data inspection and labeling with simultaneous aggregation of incoming data for efficient model updates.

II. POSE ESTIMATION OF OBJECTS FOR ROBOTIC SYSTEMS

A. Overview

Reconfigurable AI-based robotic systems require the overall tasks to be decomposed into smaller basic tasks. These basic

tasks are then the building blocks to more complex tasks that can be implemented.

One of the most basic topics in robotic systems is grasping capability, particularly when a robotic arm interacts with and grasps objects. Grasping is performed based on RGB-D images, which provide both color (RGB) and depth (D) information at the pixel level, thereby integrating visual and three-dimensional spatial data. From these images, estimation of the object pose in six degrees of freedom (6-DoF) is required to enable accurate and reliable manipulation.

Pose estimation of objects are performed basically in two different ways: by geometric fit, or using AI-based methods. Geometric fit uses the point cloud of the target object to fit to the reconstructed 3D point cloud of the scene, resulting in 6 degrees of freedom transformation. However, the fit can be problematic, since iterative algorithms developed for the problem can have many local minima resulting in a completely wrong fit. Modern AI-based methods try to enhance the classic methods by using neural network-based solutions. Mostly, these are based on networks trained on images having objects with ground-truth poses. However, ground-truth poses are hard to label, so commonly rendered images are used instead, which on the other hand causes problems in generalization. Common examples of AI-based solutions are PoseCNN [67], FoundationPose [68] or GDR-Net [69].

TABLE II
EVALUATION METRICS FOR SYNTHETIC DATASET QUALITY ACROSS FIDELITY, UTILITY, AND PRIVACY DIMENSIONS.

Dimension	Metric	Description
Fidelity	Fr�chet Inception Distance (FID)	Measures the distance between feature distributions of real and synthetic images using a pre-trained Inception network. Lower FID indicates higher similarity [45].
	Structural Similarity Index Measure (SSIM)	Assesses the similarity between two images based on luminance, contrast, and structure [46].
	Multi-Scale SSIM (MS-SSIM)	Extends SSIM by evaluating image similarity at multiple scales, capturing more perceptual information [47].
Utility	Precision and Recall	Evaluate the accuracy of object detection or segmentation tasks performed on synthetic data compared to real data [48].
	F1 Score	Harmonic mean of precision and recall, providing a balance between the two for classification tasks [48].
Privacy	Exact Match Score	Calculates the proportion of synthetic data records that exactly match real data records, indicating potential privacy risks [49].
	Row Novelty	Assesses the uniqueness of synthetic data records compared to real data, with higher novelty indicating better privacy preservation [49].
	Differential Privacy Metrics	Quantify the privacy guarantees provided by the data generation process, ensuring that individual data points cannot be re-identified [50].

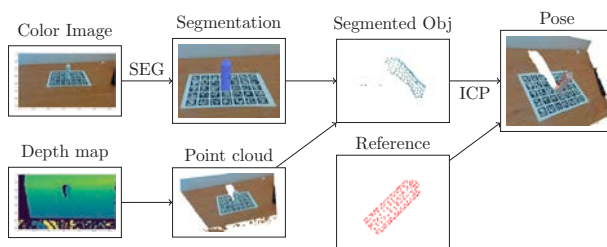


Fig. 1. The pipeline of the hybrid pose estimation algorithm. In the first step the color image is segmented using AI based image processing, then the Depth Map is clipped by the segments. The segmented Point Cloud is fitted using ICP to the reference object.

B. Methods

One particular disadvantage of existing pose estimation methods is their reliance on a highly detailed, textured 3D model of the object. This requirement can be a significant limitation, as the accuracy of the pose estimation process is highly dependent on the quality and completeness of the reference model. While this dependency may not pose a challenge in many industrial applications – where precise CAD models are readily available – there are numerous scenarios where obtaining or constructing such high-fidelity CAD models is impractical, time-consuming, or even infeasible. For instance, in applications involving objects that are irregular, subject to frequent changes, or lack pre-existing digital representations, generating detailed 3D models manually can be an unnecessary burden.

To overcome this issue, RGBD images provide an alternative approach by enabling geometry-based methods that align point clouds of objects with reconstructed point clouds of the surrounding environment. One of the most widely used techniques for this task is the Iterative Closest Point (ICP) algorithm ([70]), which iteratively refines the alignment between two point clouds by minimizing the distance between corresponding points. Although ICP has proven effective in

various applications, it suffers from several well-documented limitations that hinder its robustness and reliability, particularly for small or featureless objects. One major drawback is its susceptibility to local minimum, meaning that the algorithm is not guaranteed to converge to the optimal alignment, especially in the presence of noise, occlusions, or insufficient geometric features. Furthermore, the method struggles when significant portions of an object are occluded or outside the field of view – such as back-facing surfaces that are not captured in the depth image –further complicating the alignment process and reducing accuracy. As a result, ICP alone is often insufficient for precise (or even coarse) pose estimation, necessitating the development of more robust hybrid approaches or alternative algorithms that can compensate for these weaknesses.

Building on these findings, we propose a hybrid approach to pose estimation that enhances stability and feasibility. The method begins by identifying potential objects using well-established AI-based object detection and segmentation algorithms. Once the relevant objects are selected, ICP is applied exclusively to the corresponding 3D reconstructed points, significantly improving robustness and reliability. This approach eliminates the need for highly detailed CAD models, as only the point clouds of the objects are required, which are much easier to obtain. The complete pipeline is illustrated in Figure 1. Our method does not depend on a specific object detection and segmentation model – these components can be replaced with more suitable alternatives depending on the application.

For addressing a specific bottle pose estimation use case, the result of the first two steps of the pose estimation pipeline is depicted in Figure 2. In this example, object detection utilizes a RT-DETR model [71] and the segmentation applies a SAM model [40]. After having selected the proper object classes (e.g. in this case, bottles), based on the segmentation, the reconstructed 3D point cloud of the objects are cut from the point cloud of the environment, as shown in Figure 3. After fitting the model, the pose (location and orientation) of the



(a) Object detection results for the bottle objects (RT-DETR)



(b) Object segmentation result for the bottle objects (SAM)

Fig. 2. The first two steps of the pose estimation pipeline: object detection and segmentation



Fig. 3. The result of the point cloud fitting using ICP algorithm for one of the bottles

object is available from the ICP algorithm.

It is important to note that using the method, the poses of multiple objects can be done simultaneously – this can be considered as an advantage e.g., in situations, where robotic grasping requires collision objects to estimate.

Our solution is a hybrid method: it uses AI for segmenting the point cloud based on the RGB image, and after that, it uses iterative point cloud matching, resulting in good generalization, multiple object fit, and high performance.

C. Results

The proposed solution was compared to the state-of-the-art FoundationPose algorithm [68]. The measurement method is based on an Aruco marker-based ground-truth pose estimation to which both the proposed algorithm and the FoundationPose are compared (see Figure 4) [72].

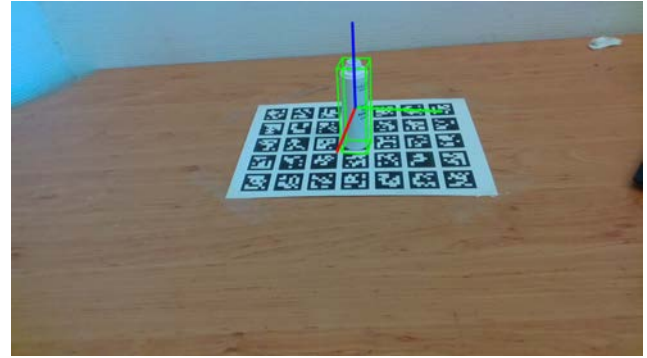


Fig. 4. Evaluating pose estimation utilizing Aruco markers to infer ground-truth poses.

Two separate measurements were conducted, each utilizing a sequence of 300 frames per algorithm. For evaluation, we calculated the translation error and the angle error, where the angle error is the angular difference between the Y axes, since the object was rotational symmetric. The first measurement uses a smooth and slow camera motion (scenario 1) while the second has much more harsh movements with some blurred images (scenario 2). The results can be seen in Figure 5 and Figure 6. Comparing the two algorithms, the FoundationPose has lost tracking in case of harsh movements, but the two methods provide similar accuracy in both angle and location estimation for the pose. However, the proposed solution does not require a textured CAD model but the point cloud for the object. Also, the one-shot speed is much faster, i.e. 100 milliseconds compared to the 2 seconds for the FoundationPose (registration state).

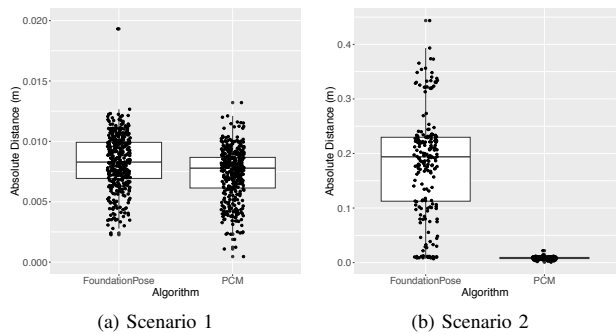


Fig. 5. Comparison of FoundationPose and the proposed solution in estimating the location part of the pose.

III. COMPUTER VISION FOR SOCKET DETECTION

In industrial environments, object detection models play a critical role in automating quality control, assembly verification, and robotic manipulation tasks. These models enable machines to detect and localize specific items, components, or conditions within visual data, significantly reducing the reliance on manual inspection and increasing throughput and consistency. The integration of computer vision systems into production lines helps streamline workflows, reduce human

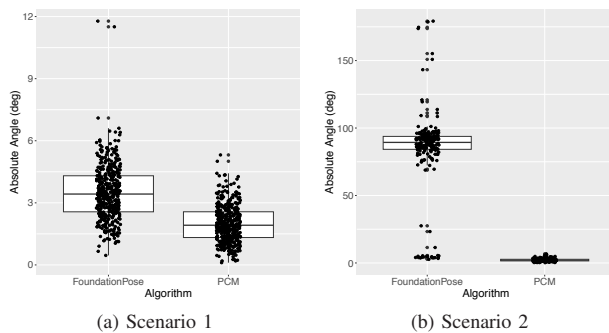


Fig. 6. Comparison of FoundationPose and the proposed solution in estimating the angle part of the pose.

error, and enable real-time decision-making, ultimately improving efficiency and product quality [73].

Modern object detection techniques leverage deep learning architectures trained on large datasets to recognize and localize objects within images or video frames. Among the most prominent object detectors is the YOLO (You Only Look Once) family of models, known for their speed and accuracy. YOLO divides the input image into a grid and simultaneously predicts bounding boxes and class probabilities for each region. This single-shot approach enables real-time performance, making it especially valuable for time-sensitive applications in manufacturing and robotics [74, 75]. The same model family is used for real-time detection in video outside manufacturing as well, for instance in surveillance [76].

In addition to YOLO, other well-known object detection models include SSD (Single Shot MultiBox Detector) and Faster R-CNN. SSD offers a balance between speed and accuracy, making it suitable for mobile or embedded devices [77], while Faster R-CNN provides higher precision at the cost of computational speed, often used in applications where accuracy is critical [78]. These models are typically trained on large-scale annotated datasets, such as COCO [79] or Pascal VOC [80], which contain thousands of labeled images across diverse object categories. However, in industrial scenarios, domain-specific datasets are often required due to the unique appearance of components and the structured nature of the environment - such environments typically involve repetitive patterns, consistent camera angles, and specific lighting conditions, which are not well represented in general-purpose datasets.

In this study, semantic segmentation was applied to a bottle socket detection task within a production context. Given the circular shape of the sockets, manual segmentation was used to create precise pixel-level labels distinguishing between empty and filled sockets. These high-quality annotations make it possible to train both segmentation and object detection models. However, the present results focus on the first segmentation-based experiment. The motivation for choosing semantic segmentation at this stage was to enable accurate estimation of the socket center - a key requirement for subsequent robotic tasks such as inserting a bottle into the socket.

Accurate localization of the socket center is especially critical

in industrial applications involving robotic manipulation. In bottle socket insertion tasks, even small positional errors may result in failed placement due to tight mechanical tolerances. Therefore, spatial accuracy is a key performance requirement in this study, and segmentation was chosen to achieve pixel-level precision in identifying the socket region and its center.

A. Methods

In this work, semantic segmentation was applied to distinguish between two classes: empty sockets and sockets containing bottles. This method involves assigning a class label to each individual pixel in the image, allowing for a highly detailed representation of object boundaries. As a result, it enabled the creation of precisely annotated training data, which were then used to train the YOLOv8m neural network. The model effectively learned to differentiate filled sockets (labeled as bottles) from empty ones (labeled as sockets).

To capture the necessary video data to train and evaluate the bottle socket detection system, an Intel RealSense D435 camera was utilized. The video footage was recorded at a resolution of 1280×720 pixels with a frame rate of 30 FPS. The camera was positioned approximately 8 cm above the conveyor belt and inclined at an angle of 10 to 15 degrees. Variations in this angle were observed across different recordings, indicating potential inconsistencies in camera positioning and highlighting the importance of improved stabilization methods.

A training dataset consisting of 252 images resized to 320×320 pixels was created from these videos. Only the region directly beneath the camera, offering the most perpendicular viewpoint, was included in the dataset to minimize perspective distortions. Manual segmentation was employed to distinguish between two classes: empty sockets (cyan) and sockets containing bottles (blue) (Figure 7). This manual labeling was particularly important due to the visual similarity between the white conveyor belt and the white bottles. The YOLOv8m neural network was trained using this dataset to enhance the detection accuracy of empty sockets.

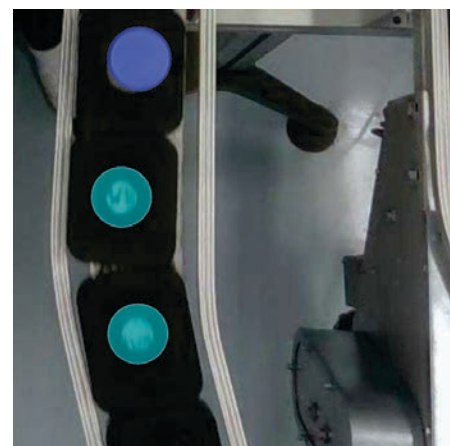


Fig. 7. Example of image from socket detection dataset

During model training, both validation and test datasets were derived from footage captured with the Intel RealSense

D435 camera. These datasets were used to calculate standard performance metrics such as mean Average Precision (mAP).

Separately, we conducted an additional test using a Logitech Brio 4K camera mounted on an adjustable tripod. This camera provided higher resolution and greater control over capture parameters. The goal of this test was to evaluate how well the trained model performs when applied to data from a different camera. To support this evaluation, an ArUco marker was used for camera calibration and precise localization of the bottle socket.

The detection accuracy evaluation involved comparing the predicted socket center with the manually determined true center, identified as 4 cm horizontally from the center of the ArUco marker. The actual center was visually annotated with a pink cross using OpenCV, serving as a visual reference only.

B. Results

The detection accuracy of the model was quantified using the Root Mean Square Error (RMSE) under various experimental conditions, altering the camera resolution, distance, and angle. Thirty images were analyzed for each experimental scenario. The summarized results are presented in Table III.

TABLE III
ACCURACY RESULTS UNDER VARYING EXPERIMENTAL CONDITIONS

Resolution (px)	Distance (cm)	Angle (°)	RMSE (mm)
2560×1440	60	0	6.6
1280×720	80	0	4.2
1280×720	80	10	3.3

The results indicate that slightly increasing the camera angle improved detection accuracy, whereas higher resolution did not necessarily enhance performance.

To further assess the spatial accuracy of the empty socket detection model, an XY deviation analysis was performed, as illustrated in Figure 8. This analysis compares the predicted socket centers to the ground truth positions under consistent experimental conditions: a resolution of 1280×720 pixels, a camera height of 80 cm, and a tilt angle of 10°. Eight test scenarios were examined, each involving systematic rotation of the sockets to evaluate robustness across orientations. The observed positional deviations ranged from a minimum of 2.1 mm to a maximum of 3.9 mm, demonstrating the model's stable performance within a narrow margin of error.

In conclusion, the achieved RMSE of 3.3 mm in detecting the socket center initially suggests a relatively high model accuracy. However, considering the test socket diameter of 36 mm and bottle diameter of 32 mm, the available clearance is only 2 mm on each side. Since the average deviation exceeds this margin, it is likely that bottles will frequently fail to align properly with the socket, especially in the absence of guiding elements or corrective mechanics. This indicates that the current level of positioning precision is insufficient for reliable insertion under these conditions. To ensure dependable performance, improvements are necessary — either by increasing detection accuracy (e.g., through enhanced calibration or data refinement) or by modifying the mechanical setup, such as enlarging the socket diameter or adding alignment aids.

Pixel-wise Deviation of Detected Socket Center from True Center

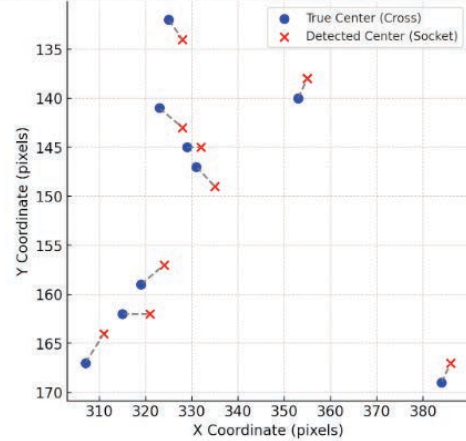


Fig. 8. Socket center comparison

Without such adjustments, the system may not consistently meet industrial requirements for automated insertion tasks.

IV. AI SOLUTIONS FOR SEMICONDUCTOR QUALITY CONTROL

A. Overview

In semiconductor production, ensuring the delivery of high-quality chips and products is of utmost importance due to the critical role semiconductors play in modern technology. To achieve this, extensive inspections and quality checks are conducted throughout the production process, many of them involve the analysis of images. Currently, these defect images are manually analyzed or classified by experts, a process that is time-consuming, labor-intensive, and prone to subjective biases. Automating these inspections through advanced methods such as defect image classification has become a central area of innovation to improve objectivity, efficiency, and scalability. Additionally, given the diverse domains within semiconductor production, which arise from variations in materials, processes, and equipment, it is essential to explore methods for domain adaptation. These techniques enable us to effectively transfer a model trained in one domain to another, addressing the challenge of domain-specific variations without requiring extensive retraining or additional labeled data. Furthermore, in cases where defects are rare or data labeling and collection is expensive, anomaly detection methods are applied to identify unusual patterns. By integrating these approaches, semiconductor manufacturers can enhance product quality, streamline operations, and contribute to sustainability by reducing waste and resource consumption, ultimately creating a more robust and efficient production pipeline. The following subsections will provide a more detailed description of the methods: defect image classification, domain adaptation, and anomaly detection, explaining their contributions to the semiconductor production process.

B. Defect Image Classification

Defect image classification may seem straightforward at first glance, given that the field has been around for quite some time. However, it is far from simple in practice. A review of the literature might suggest that the task can be easily accomplished by using one of the many pre-trained networks on public datasets, applying transfer learning to semiconductor production images, and evaluating the model's performance based on an overall metric, such as the F1-score. While this sounds efficient, the reality is much more complex. Even if we assume that the time-intensive tasks of data collection and labeling have already been completed, significant challenges remain. A major challenge is the highly imbalanced nature of datasets collected over a longer production period, as some defect classes occur less frequently than others. Such imbalances can bias the training process, leading to models that perform poorly on less frequent but critical defect types. Another key difficulty lies in the evaluation criteria: relying solely on overall metrics, such as accuracy or F1-score averaged across all classes, is insufficient for production settings. For critical defect classes, we aim to achieve an F1-score of at least 0.8, as these defects have a disproportionate impact on product quality. Additionally, the evaluation requires a detailed analysis, assessing performance not only at defect class level but also across defect images associated with different base materials. This level of granularity is essential since the dataset includes images originating from three distinct base materials. These additional requirements add layers of complexity to both model development and evaluation. In the following sections, we will describe the approaches we have taken to address these challenges.

Figure 9 illustrates the distribution of the defect image dataset. The y-axis represents the number of images, while the x-axis denotes the different defect classes, sorted in descending order by image count. The different colors in the figure indicate the base materials. Differentiating base materials is important, as defects from base materials 1 and 2 are very similar, while defects from base material 3 differ significantly. Additionally, certain defect classes are exclusive to base material 1 and 2, while others only occur on base material 3.

A closer examination reveals that the dataset is imbalanced in several ways. Most images come from base material 1, followed by base material 3, with very few from base material 2. However, since defects in base material 2 are similar to those in base material 1, the smaller number of images from base material 2 can be disregarded in this context. In contrast, the imbalance between base material 3 and the other two is significant, highlighting the need for more defect images from base material 3 to balance the dataset. Additionally, the dataset is imbalanced across defect classes, as some defects occur far less frequently during production than others.

To address these imbalances and ensure that the model does not become biased toward well-represented defect classes, we apply data balancing and data augmentation strategies to the dataset in the next step. Importantly, these data balancing and augmentation techniques are applied exclusively to the training

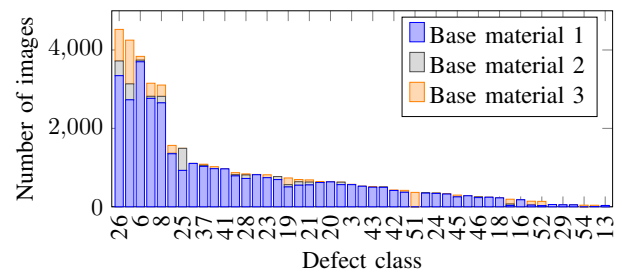


Fig. 9. Dataset distribution per base material

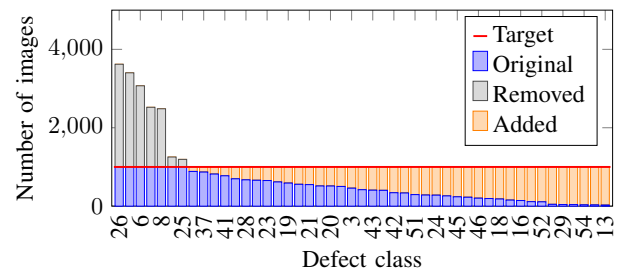


Fig. 10. Data-level balancing

images. The dataset is first split into separate training and validation sets. While the training set undergoes data balancing and augmentation, the validation set remains untouched. This ensures that the validation set reflects the original dataset distribution, allowing for an unbiased evaluation of the model's performance. By maintaining this separation, we can ensure that the results on the validation set accurately represent the model's real-world generalization ability.

These preprocessing steps, data balancing and data augmentation, are designed to improve the robustness of the model and enhance its ability to effectively learn from underrepresented defect classes, as discussed in publications [81–84]. Specifically, when working with datasets from semiconductor production similar to the one in this work, we found that applying data-level balancing in combination with moderate data augmentation yields the best results [85]. Figure 10 visualizes the resulting distribution of the defect image training dataset after resampling. The colors indicate the resampling techniques applied. To address class imbalance, we used a two-fold strategy: for overrepresented classes, excess images were randomly removed, and for underrepresented classes, the dataset was supplemented with replicated images. The blue bars represent the original images, gray bars show removed images from overrepresented classes, and orange bars indicate replicated images added for rare defect classes. The red line at 1,000 images marks the resampling target, determined experimentally as the optimal balance for this dataset.

Figure 11 illustrates examples of the augmentation techniques applied. These techniques are used to diversify the dataset and are dynamically applied to each image before being fed into the model for training. The augmentation methods include shearing, rotating, flipping, shifting, and zooming, applied within carefully defined moderate ranges to

avoid distorting defect characteristics, such as zooming in too much or shifting defects out of the image. During each training iteration, images are augmented with a random combination of transformations and settings within these ranges, ensuring improved variability during training.

For model training, we utilized transfer learning, leveraging the EfficientNetV2S architecture pre-trained on the ImageNet dataset [86]. The balanced and augmented dataset served as the input for fine-tuning the model. By combining resampling, augmentation, and transfer learning, we aimed to maximize the model's ability to generalize effectively while mitigating the challenges posed by the inherent dataset imbalances.

Figure 12 shows the model's performance for each defect class, grouped by base material. The y-axis represents the F1-score, while the x-axis lists defect classes sorted in descending F1-score order. Colors correspond to the base materials, following the same scheme as Figure 9. Transparent circles indicate defect classes with fewer than 20 validation samples, which are excluded from the analysis to ensure reliable evaluation. These classes require additional defect images for better representation. The red line at an F1-score of 0.8 marks the target performance for critical defect classes.

A detailed analysis of the model's performance reveals that the F1-score for certain defect classes falls below expectations, particularly for images associated with base material 3. This issue is primarily attributed to the limited availability of training images for this base material, which significantly affects the model's ability to generalize effectively. To address this, we have made the strategic decision to split the model into two separate ones: one dedicated to base materials 1 and 2, and another one for base material 3. This approach allows us to already use the model for base materials 1 and 2 in production, while iteratively collecting additional images and retraining the model for base material 3 in subsequent steps.

Over time, we will collect more defect images to fill up the underrepresented defect classes and further improve the performance of the models. However, the frequency of updates for the base material 1 and 2 model will be lower compared to the one of base material 3 model. With each base material 1 and 2 model update, we must ensure that the new version maintains or exceeds the performance of the previous version across all critical defect classes, as consistent reliability is crucial for productive usage.

These efforts aim to enhance automation in semiconductor manufacturing, improving efficiency, consistency, and scalability. Advancing these technologies helps reduce resource waste, improve production quality, and meet the growing demand for reliable, high-performance semiconductor products.

C. Domain Adaptation

Domain adaptation is a machine learning technique that transfers a model trained on one dataset (the source domain) to perform well on a different but related dataset (the target domain) where the data distribution is different because of a distribution shift. Instead of needing large amounts of new labeled data, domain adaptation allows models to adapt using

little or no labeled data from the target domain. This makes it especially useful in fields like defect classification, where collecting and labeling new data can be expensive and time-consuming. The two main domain adaptation scenarios are:

- Unsupervised Domain Adaptation (UDA): no target domain labels are available.
- Semi-Supervised Domain Adaptation (SSDA): some target domain samples are labelled and others are not.

The distribution shift in semiconductor manufacturing can occur, and Scanning Electron Microscope (SEM) images can vary significantly across different products, acquisition processes, or acquisition stages.

Domain adaptation methods have shown considerable promise in enabling scalable, robust defect classification across these variations, ensuring that models remain accurate without exhaustive retraining. Despite their potential, domain adaptation approaches have seen limited exploration and adoption compared to traditional defect detection and classification models. Moreover, this gap highlights a critical opportunity: leveraging domain adaptation can significantly mitigate the data scarcity issue, a persistent barrier in deploying machine learning solutions in manufacturing environments.

To tackle the distribution shift challenge in SEM images for defect classification, we developed a CycleGAN-based method named DBACS (DANN-Based Alignment with Cyclic Supervision) [87], which can work in both the previous domain adaptation scenarios.

The closed-source dataset used to develop the model comprises three data domains representing different products, with 2 defect classes: particle defects and point defects. Table IV summarizes the dataset distribution across the three considered domains. While Domain 0 and 1 exhibit a relatively balanced class distribution, Domain 2 presents a significant imbalance, with point defects being more than twice as frequent as particle defects.

TABLE IV
SAMPLE DISTRIBUTION PER CLASS AND DOMAIN IN THE CONSIDERED
CLOSED SOURCE DATASET

Domain	Particle Defects	Point Defects	Total
Domain 0	1706	1516	3222
Domain 1	639	503	1142
Domain 2	577	1196	1773
Total	2922	3215	6137

Examples of the SEM images are presented in Fig. 13

The purpose of the Cycle GAN is to transform an image from a given domain to another without altering the defects present in the image. The model has to change the defect's background while maintaining it unaltered. This is ensured by incorporating the Cyclic Supervision [88] mechanism of the Cycle GANs and additional losses that control the image transformation. As a result, a classifier trained on one domain can perform effectively in another by leveraging the image transformation performed by the Cycle GAN. We focus our experiments on the UDA and SSDA scenarios. Our method, DBACS, was compared with AdaMatch [89], a state-of-the-art domain adaptation method which can work in the UDA and SSDA scenarios.

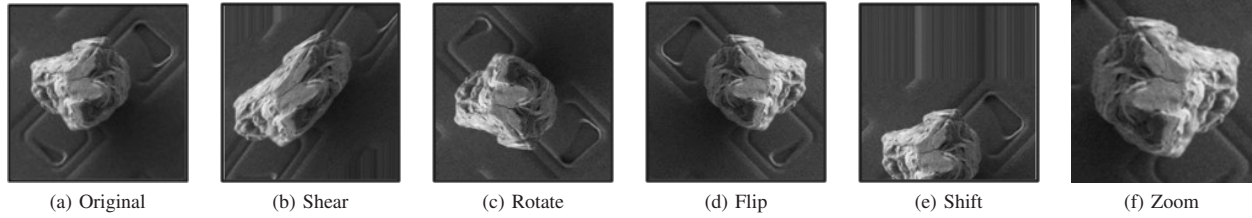


Fig. 11. Applied augmentation techniques

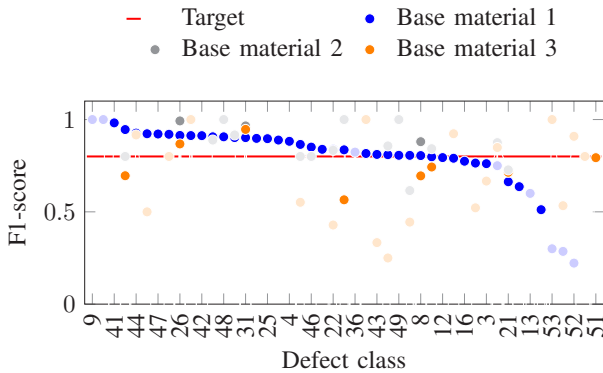


Fig. 12. Model evaluation based on the F1-score for each defect class in the validation dataset.

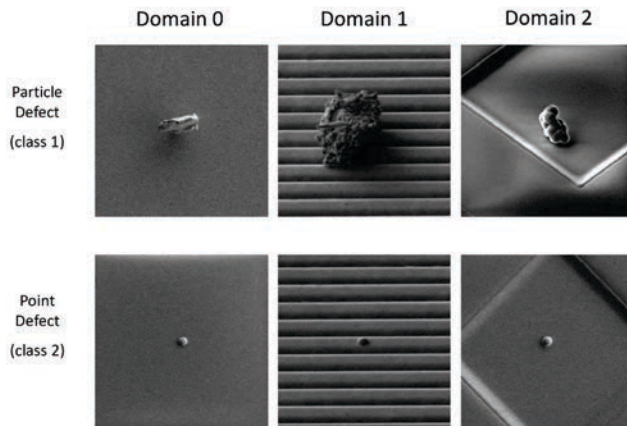


Fig. 13. We address the domain adaptation problem in image classification in the context of semiconductor manufacturing, considering two defect classes: particle defect and point defect. Each column represents a distinct domain.

The experiment results for the UDA scenario are shown in Tables V and VI. The results for the SSDA scenario are shown in Table VII. For every scenario, we tested three different networks: a MobileNet [90], a Resnet50 and a Resnet101 [91]. The three models are pre-trained on ImageNet and fine-tuned on the source domain. After that, the domain adaptation technique is applied, and the model is tested on the target domain test set. In every scenario, the Oracle is a model trained on the labelled target domain and is an upper bound for the performance. The reported values are the classification

accuracies on the test sets. The results show that AdaMatch performs well in all the considered scenarios and outperforms DBACS. However, in the UDA scenario with source domain 1, our method DBACS outperforms AdaMatch. This is caused by the entirely different label distributions between the considered source and target domains. AdaMatch, during its processes, considers this distribution equal and performs poorly in this case.

D. Anomaly detection

In semiconductor manufacturing, defect detection and classification remain challenging tasks due to the high complexity of wafer images and the inherent rarity of defects, which leads to significant data imbalance in training datasets. Anomalies such as aluminium corrosion or surface defects are often visually subtle, difficult to identify manually, and typically require domain expertise for accurate classification. Several state-of-the-art unsupervised anomaly detection methods, including PatchCore [92] and CFA [93], have been explored for this purpose. However, these approaches tend to fall short in high-complexity scenarios where defects are insidious and hard to distinguish from normal patterns.

A challenging problem in this context is the recognition of aluminium corrosion. Developing deep learning models capable of reliably detecting this type of defect is highly difficult. The datasets available for training are typically small, noisy, and feature complex images, exacerbated by a limited number of samples and severe class imbalance: common characteristics in semiconductor defect detection, where anomalies are rare, difficult to collect, and require expert labeling. Furthermore, the considered images are not SEM images but optical images with a very low resolution. In this setting, unsupervised anomaly detection approaches failed to achieve satisfactory results, demonstrating the limitations of unsupervised methods for complex defect types. As a result, a shift toward supervised learning techniques becomes necessary. A similar observation was made for anomaly detection on image representations of radio spectrum data, where autoencoder features gave the best results when combined with a supervised classifier [94].

In the broader field of computer vision, U-Net-based architectures [95] have proven highly effective for segmentation tasks across domains, including medical imaging and beyond. When adapted to anomaly detection in wafer images, the U-Net model can accurately detect and segment defects, pro-

TABLE V
UDA MODELS ACCURACY - SOURCE DOMAIN 0.

Source → Target	0 → 1			0 → 2		
Model	MobileNet	ResNet50	ResNet101	MobileNet	ResNet50	ResNet101
DBACS	79,3	72,4	80,2	64,53	61	65,64
AdaMatch	81,89	84,15	85,3	79	81,23	84,32
Oracle	97,3	85,8	98,2	95,9	81,1	97,1

TABLE VI
UDA MODELS ACCURACY - SOURCE DOMAIN 1.

Source → Target	1 → 0			1 → 2		
Model	MobileNet	ResNet50	ResNet101	MobileNet	ResNet50	ResNet101
DBACS	67,84	69,53	78,75	55,18	63,24	65,27
AdaMatch	56,65	60,78	65,74	52,86	54,33	60,24
Oracle	97,6	92,1	98,1	95,9	81,1	97,1

TABLE VII
SSDA (5% OF LABELED DATA IN THE TARGET DOMAIN) MODELS ACCURACY.

Source → Target	0 → 1			0 → 2		
Model	MobileNet	ResNet50	ResNet101	MobileNet	ResNet50	ResNet101
DBACS	81,67	82,74	82,86	70,03	77,87	79,83
AdaMatch	88,1	89,11	90,12	85,25	86,75	87,13
Oracle	97,3	85,8	98,2	95,9	81,1	97,1

TABLE VIII
DISTRIBUTION OF CLASSES IN THE DATASET

Class	Train	Validation	Test
Corrosion	273	35	34
No Corrosion	1391	175	172
Total	1664	210	206

TABLE IX
DISTRIBUTION OF LABELS IN THE TRAINING SET AFTER DATA AUGMENTATION

Corrosion	No Corrosion	Total
1365	1391	2756

viding enhanced model interpretability through explicit defect localization.

The approach involves training the model in a supervised way, using ground truth mask annotations that focus learning on defect patterns. Furthermore, a carefully designed data augmentation pipeline is applied during training to counteract class imbalance and improve robustness. This strategy significantly enhances performance on the aluminium corrosion recognition task, demonstrating that a supervised segmentation approach can outperform unsupervised methods in complex, data-scarce manufacturing environments.

Tables VIII and IX show the distribution of classes in the considered dataset and the distribution of labels in the training set after data augmentation. The data augmentation was done with vertical flip, horizontal flip, and transposition on both diagonals to images with corrosion defects. This ensures a balanced distribution in the dataset.

In Table X, a comparison of classification results (the Bal-

anced Accuracy Score is reported) and the number of trainable parameters of the different networks trained in a supervised way on the aluminium corrosion detection task is reported. The reported Shallow-CNN comprises 6 convolutional layers and a fully connected network on top, this model was trained from scratch on the considered dataset. The EfficientNetV2S considered was pretrained on the ImageNet dataset and then fine-tuned on our dataset. The ResNet101 models considered were pretrained on the ImageNet dataset or on the PCAM dataset and then fine-tuned on our dataset. Only the U-Net model can do segmentation; the other models were employed for simple classification. As can be seen from the results, the best model is the U-Net model, which outperforms the other classification-based methods and can also segment the input images, allowing model interpretability.

V. LUMINAIRE ASSEMBLY

One of the challenges in applying AI based computer vision in assembly is the cost/benefit tradeoff. Signify can manufacture and sell small volumes customized to specific requirements of a lighting project with short lead times. The variation in the products we manufacture, and the low volumes is at odds with traditional supervised machine learning approaches where AI algorithms learn from human feedback. In supervised machine learning the algorithm is trained to recognize parts on large datasets of human annotated data. Several innovations, such as fine tuning can reduce the amount of training data needed to achieve a high accuracy. But creating a dataset of several hundred to thousands of images costs time and money to create, especially if the trained algorithm does not generalize well to recognize assembly of functionally similar but different looking luminaires. One example we encountered recently was a trail in one of our factories where

TABLE X
COMPARISON OF CLASSIFICATION RESULTS AND NUMBER OF TRAINABLE PARAMETERS

	BACC	Precision	Recall	F1-Score	No. Parameters (in mln)
Shallow-CNN	50.0%	0.0%	0.0%	0.0%	20.8
EfficientNetV2S	93.0%	77.5%	91.2%	83.8%	20.3
ResNet101 (ImageNet)	89.8%	66.7%	88.2%	76.0%	42.7
ResNet101 (PCam)	95.9%	78.6%	97.1%	86.8%	42.7
U-Net	97.1%	86.8%	97.1%	91.7%	31.0

we aimed to recognize the timing of several steps in the assembly of an outdoor luminaire.

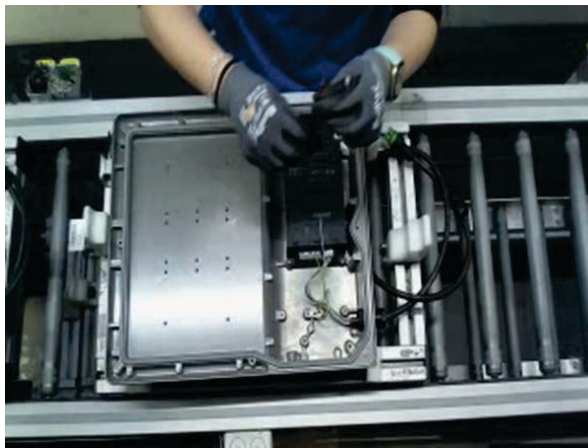


Fig. 14. Assembly process.

Here we mounted a camera above a workstation where the driver is installed in an outdoor luminaire. Fig. 14 shows an image from the camera. The goal is to determine the timings of each step in the assembly process. There are seven steps involved in mounting the driver and connecting the wires. A recording of approximately 45 minutes was taken and annotated using an open-source video annotation tool [96]. Fig. 15 shows an example. Once the entire video was annotated, we obtained 380 clips over seven classes.

To determine whether we could learn an action recognition model to identify the seven classes we fine-tuned a VideoMAE model. The model was already pre-trained on the Kinetics 400 dataset. The dataset contains 400 human action classes, with at least 400 video clips for each action. The clips are taken from different Youtube videos and cover a broad range of classes, such as playing instruments, shaking hands, dancing or doing yoga.

A challenge with using open-source models is that often they are trained on datasets outside of an industrial setting, this makes that fine-tuning may perform less well in practice when applied on datasets from industry vs academia. In academic papers the datasets used are more based on common everyday tasks, which aligns much more with what's present in the original dataset. Another challenge is the availability of well-structured data. Metadata on products is partially manually entered into a system. The reliability of this data depends on the human operator and humans are prone to making mistakes especially when trying to meet tight deadlines.

We finetuned the pre-trained model on our own dataset. The model was tested against 10% of the clips left out of the training set. The model performed quite well on this test set (82%). Later we collected more videos on the same line, where different sizes of the luminaire were being assembled with the same operations: mounting and connecting the driver. On those luminaires that had a different size, the model performed poorly. Visually the luminaires look different but to a human, the operations, are clearly similar. This is where AI vision-only models fail, as they rely on visual cues only and have no real semantic understanding of what is going on in the video. An approach that may prove more successful is a staged approach, where the parts and the movements of the operator are identified by several AI models, for example objection detection and hand tracking, and this is then fed to a third model to determine the assembly steps. But this would prove costly as it requires a more extensive dataset, for object detection and for training the assembly step model. We experimented with some newly released advanced multi-modal models to see whether these could play a role to reduce the cost associated with creating datasets. Zero-shot object detectors and multi-modal AI may play a role as they advance further and are trained on industrial datasets but for now these didn't work sufficiently well on our data to play a role.

More experimentation with an external company is ongoing but for cost benefit reasons it's important that any trained model generalizes well to other luminaires. A theoretical 20% percent improvement on efficiency of the line (which is very high) would be needed to come close to the costs associated with a commercial solution.

VI. SEMI-SUPERVISED APPROACH FOR INDUSTRIAL DATA ANNOTATION

A. Introduction

Manual data annotation remains a critical bottleneck in supervised machine learning, particularly in applications that require large-scale, domain-specific, or expert-curated labels. While high-quality annotations are essential for training performant models, the cost and time associated with manual labeling can exceed the foreseen return on investment. Automated annotation methods aim to alleviate this burden by leveraging machine learning models to generate labels with minimal human intervention. These techniques often combine elements of supervised, weakly supervised, and semi-supervised learning to bootstrap model performance and iteratively enhance label quality.

Semi-supervised learning (SSL) has gained considerable attention as a promising strategy for making use of vast



Fig. 15. Image annotation process.

amounts of unlabeled data. In typical SSL pipelines, a model is first trained on a small labeled dataset and then used to generate pseudo-labels for the remaining unlabeled data. These pseudo-labels, usually filtered based on confidence, are iteratively included in the training set to reinforce the model's learning. Methods such as consistency regularization [97], entropy minimization [98], and teacher-student paradigms [99] have shown remarkable success in reducing the reliance on labeled data while maintaining high accuracy. A comprehensive overview of semi-supervised techniques can be found in recent surveys [100, 101].

B. Processing Pipeline

The proposed annotation pipeline consists of four key stages. The first stage, *Initial Training*, involves training a model on a small set of clean, manually annotated data to establish a solid foundation for inference. In the second stage, *Semi-Supervised Learning*, the model is used to generate pseudo-labels for new, unlabeled data. These pseudo-labeled samples are then incorporated into the training process, allowing the model to learn from both the initial ground truth and its own confident predictions. This approach gradually improves the model's generalization capability as it is exposed to a broader data distribution.

Once performance has sufficiently improved, the pipeline proceeds to the *Re-Inference* phase, where the updated model re-labels the dataset. If the new labels are of higher quality, they are used for additional retraining, further reinforcing the model performance. This iterative process, core to the semi-supervised paradigm-enables the model to refine its own outputs over time, making increasingly accurate predictions without additional human annotation. Finally, in the *Results Acquisition* stage, the final labeled dataset is aggregated and

analyzed to evaluate label quality and overall model performance.

To quantify the effectiveness of the pipeline, we compare the mean Intersection over Union (IoU) scores [102] from the initial training stage and the end of the pipeline. As shown in Figure 17, the model's performance improves substantially, with the final mean IoU nearly doubling. This highlights the advantage of iterative pseudo-labeling and retraining in semi-supervised learning setups. The model used for this task was RT-DETR [103]. The dataset was an internal collection of images acquired from a battery production facility, with the objective of detecting batteries. It consisted of 100 images, all of which were manually annotated by our team.

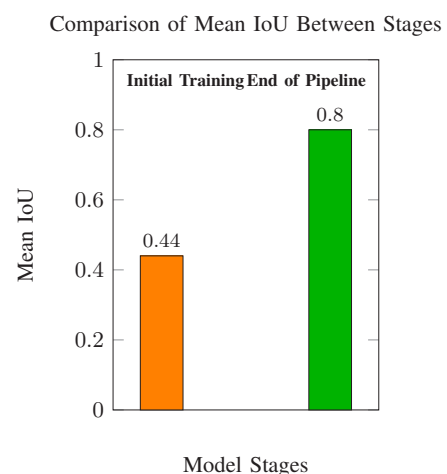


Fig. 17. Comparison of mean IoU scores between the initial training and final model stage.

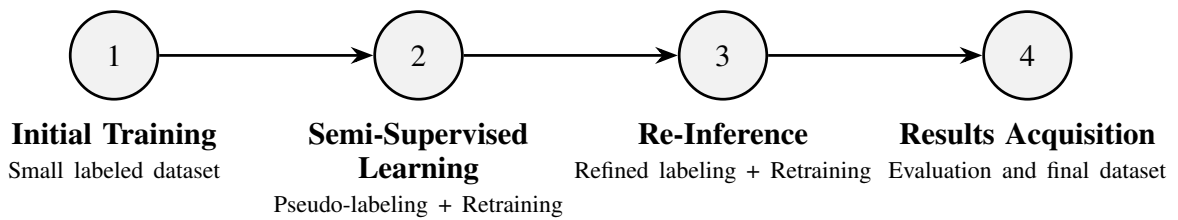


Fig. 16. High-level overview of the proposed semi-supervised annotation pipeline.

C. Summary

Semi-supervised annotation pipelines offer a scalable and cost-effective alternative to fully manual labeling by leveraging a small annotated seed dataset to iteratively train models and expand the labeled set using pseudo-labels. This method is particularly well-suited for domains where manual labeling is expensive or time-consuming. The effectiveness of such pipelines depends heavily on the quality of initial labels, pseudo-label confidence filtering, and the robustness of the underlying model. Modern approaches incorporating consistency regularization, self-training, and teacher-student frameworks continue to push the boundaries of what can be achieved with minimal supervision, opening new possibilities for fully automated or human-in-the-loop annotation systems.

VII. IDENTIFICATION OF THE RESEARCH GAPS FOR INDUSTRY4.0/5.0

A large part of adopting AI in industry revolves around data – collecting sufficient, domain-relevant samples; labeling and preprocessing them; ensuring long-term storage and maintenance; and managing large datasets over extended periods.

In defect image classification, an unresolved challenge is the effective handling of highly imbalanced datasets. Utilizing pretrained models, such as those developed on the ImageNet dataset, and applying transfer learning to specific domains like semiconductor images allows for the effective use of a limited number of images. However, handling classes with extremely few examples continues to pose a challenge. While techniques such as resampling and data augmentation can enhance the dataset, Generative Adversarial Networks (GANs) can also be employed to artificially generate new images. Nonetheless, these approaches require a baseline amount of existing data, which leads to the critical question of how much data is necessary per class to train a robust GAN or classification model. This question often necessitates empirical testing, as there is currently no established methodology or algorithm that clearly determines the optimal number of images required.

Despite recent advances in semi-supervised and active learning, automated annotation still faces significant obstacles. First, pseudo-labeling often propagates errors, especially in the early stages of training, leading to the reinforcement of incorrect annotations. Second, domain shift between labeled and unlabeled datasets can significantly degrade model performance if not properly addressed. Moreover, balancing model confidence with annotation quality remains an open issue—thresholds for pseudo-label acceptance are typically heuristic and task-dependent. Another challenge is the efficient selection

of informative samples for retraining, especially when data arrives in batches or from different sources. Also, integrating human-in-the-loop feedback in a scalable and seamless manner is still underexplored, especially in industrial or real-time contexts. These problems highlight the need for more robust, adaptive, and generalizable annotation strategies that can operate reliably across varying data distributions and tasks.

In addition to the quantity of data, the quality of the images is another vital consideration. It is essential to identify which images may be too noisy for inclusion in the training dataset. Furthermore, accurately labeling images can be challenging; borderline cases often result in different experts categorizing the same image into different classes, similar to how a dog might be confused for a cat. Therefore, it would be beneficial to develop methods that can automatically identify and remove low-quality images from such datasets, thereby improving the overall training process and model performance.

Looking ahead, several improvements appear promising. These include the development of more reliable confidence estimation mechanisms, dynamic thresholding strategies for pseudo-label acceptance, and domain adaptation techniques that can reduce the impact of dataset shift. We also aim to explore intelligent sampling methods for more effective retraining and tighter integration of human feedback loops, supported by intuitive interfaces and lightweight annotation tools. Ultimately, our goal is to make the annotation pipeline more autonomous, yet flexible enough to adapt to specific use cases and data modalities.

The future of deep learning in computer vision is likely to move toward more integrated approaches that combine CNNs and Transformers. Hybrid architectures will continue to be explored, alongside growing interest in model interpretability, real-time performance, and expanded applications in fields such as agriculture, autonomous vehicles, and environmental monitoring.

ACKNOWLEDGMENT

The research leading to these results was conducted in the AIMS5.0 project, which received funding from the Chips Joint Undertaking under grant agreement No. 101112089 and from the partners' national funding programmes.

REFERENCES

- [1] K. Han, H. Pei, R. Huai, T. Huang, and S. Qin, "Design of high-speed image processing system for weakdim target based on fpga," in *2022 7th International Conference on Image, Vision and Computing (ICIVC)*. IEEE, 2022, pp. 783–789.

Image Processing: a Review on Industrial Applications

- [2] R. Wang, T. Lei, R. Cui, B. Zhang, H. Meng, and A. K. Nandi, "Medical image segmentation using deep learning: A survey," *IET image processing*, vol. 16, no. 5, pp. 1243–1267, 2022.
- [3] S. Hussain, I. Mubeen, N. Ullah, S. S. U. D. Shah, B. A. Khan, M. Zahoor, R. Ullah, F. A. Khan, and M. A. Sultan, "Modern diagnostic imaging technique applications and risk factors in the medical field: a review," *BioMed research international*, vol. 2022, no. 1, p. 5 164 970, 2022.
- [4] Y. Li, "Research and application of deep learning in image recognition," in *2022 IEEE 2nd international conference on power, electronics and computer applications (ICPECA)*. IEEE, 2022, pp. 994–999.
- [5] D. Y. Pimenov, L. R. da Silva, A. Ercetin, O. Der, T. Mikolajczyk, and K. Giasin, "State-of-the-art review of applications of image processing techniques for tool condition monitoring on conventional machining processes," *The International Journal of Advanced Manufacturing Technology*, vol. 130, no. 1, pp. 57–85, 2024.
- [6] M. E. Rayed, S. S. Islam, S. I. Niha, J. R. Jim, M. M. Kabir, and M. Mridha, "Deep learning for medical image segmentation: State-of-the-art advancements and challenges," *Informatics in Medicine Unlocked*, p. 101 504, 2024.
- [7] B. P. Gyires-Tóth, M. Osváth, D. Papp, and G. Szűcs, "Deep learning for plant classification and contentbased image retrieval," *Cybern. Inf. Technol.*, vol. 19, no. 1, pp. 88–100, 2019.
- [8] N. E. Khalifa, M. Loey, and S. Mirjalili, "A comprehensive survey of recent trends in deep learning for digital images augmentation," *Artificial Intelligence Review*, vol. 55, no. 3, pp. 2351–2377, 2022.
- [9] R. Mao, M. Al-Shouha, and G. Szűcs, "Spectral generalized category discovery by training on combined labels," in *2024 IEEE 22nd World Symposium on Applied Machine Intelligence and Informatics (SAMI)*. IEEE, 2024, pp. 000 397–000 402.
- [10] B. Tang, L. Chen, W. Sun, and Z.-k. Lin, "Review of surface defect detection of steel products based on machine vision," *IET Image Processing*, vol. 17, no. 2, pp. 303–322, 2023.
- [11] R. Archana and P. E. Jeevaraj, "Deep learning models for digital image processing: a review," *Artificial Intelligence Review*, vol. 57, no. 1, p. 11, 2024.
- [12] A. Khan, Z. Rauf, A. Sohail, A. R. Khan, H. Asif, A. Asif, and U. Farooq, "A survey of the vision transformers and their cnn-transformer based variants," *Artificial Intelligence Review*, vol. 56, no. Suppl 3, pp. 2917–2970, 2023.
- [13] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," *CVPR*, 2022.
- [14] C. Saharia *et al.*, "Photorealistic text-to-image diffusion models with deep language understanding," *NeurIPS*, 2022.
- [15] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, "Zero-shot text-to-image generation," *ICML*, 2021.
- [16] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *CVPR*, 2017.
- [17] L. Zhang, L. S. Manevitz *et al.*, "Adding conditional control to text-to-image diffusion models," *arXiv preprint arXiv:2302.05543*, 2023.
- [18] M. Ivanovs, L. Leja, K. Zviedris, R. Rimša, K. Narbutė, V. Movčana, F. Rūmnieks, A. Strods, K. Gillois, G. Mozolevskis *et al.*, "Synthetic image generation with a fine-tuned latent diffusion model for organ on chip cell image classification," in *2023 Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA)*. IEEE, 2023, pp. 148–153.
- [19] V. Movčana, A. Strods, K. Narbutė, F. Rūmnieks, R. Rimša, G. Mozolevskis, M. Ivanovs, R. Kadiķis, K. G. Zviedris, L. Leja *et al.*, "Organ-on-a-chip (ooc) image dataset for machine learning and tissue model evaluation," *Data*, vol. 9, no. 2, p. 28, 2024.
- [20] J.-B. Alayrac *et al.*, "Flamingo: a visual language model for few-shot learning," *arXiv preprint arXiv:2204.14198*, 2022.
- [21] OpenAI, "Gpt-4 with vision (gpt-4v)," <https://openai.com/research/gpt-4v-system-card>, 2023.
- [22] A. Béres and B. Gyires-Tóth, "Enhancing visual domain randomization with real images for sim-to-real transfer," *Infocommunications Journal*, vol. XV, no. 1, pp. 15–25, 2023.
- [23] J. Yu *et al.*, "Scaling autoregressive models for content-rich text-to-image generation," *arXiv preprint arXiv:2206.10789*, 2022.
- [24] Midjourney, "Midjourney ai," 2022, <https://www.midjourney.com>.
- [25] DeepFloyd Team, "Deepfloyd if: A modular text-to-image diffusion model," 2023, <https://deepfloyd.github.io/IF>.
- [26] A. Nichol and P. Dhariwal, "Glide: Towards photorealistic image generation and editing with text-guided diffusion models," *arXiv preprint arXiv:2112.10741*, 2021.
- [27] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," *ICCV*, 2017.
- [28] J. Li *et al.*, "Blip-2: Bootstrapping language-image pretraining with frozen image encoders and large language models," *arXiv preprint arXiv:2301.12597*, 2023.
- [29] Unity Technologies, "Unity perception package," 2021, <https://github.com/Unity-Technologies/com.unity.perception>.
- [30] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," *CoRL*, 2017.
- [31] Blender Foundation, "Blender - open source 3d creation suite," 2022, <https://www.blender.org>.
- [32] NVIDIA, "Nvidia omniverse," 2023, <https://developer.nvidia.com/nvidia-omniverse>.
- [33] M. Denninger *et al.*, "Blenderproc: Procedural blender pipeline for photorealistic scene generation," 2019, <https://github.com/DLR-RM/BlenderProc>.
- [34] Meshcapade GmbH, "Meshcapade - realistic human models," 2023, <https://meshcapade.com>.
- [35] G. Varol, J. Romero, X. Martin, N. Mahmood, M. J. Black, I. Laptev, and C. Schmid, "Learning from synthetic humans," *CVPR*, 2017.
- [36] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," *CVPR*, 2019.
- [37] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional gans," *CVPR*, 2018.

- [38] A. Radford *et al.*, "Learning transferable visual models from natural language supervision," *ICML*, 2021.
- [39] H. Liu *et al.*, "Visual instruction tuning," *arXiv preprint arXiv:2304.08485*, 2023.
- [40] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," 2023. [Online]. Available: <https://arxiv.org/abs/2304.02643>
- [41] S. Liu *et al.*, "Grounding dino: Marrying dino with grounded pre-training for open-set object detection," *CVPR*, 2023.
- [42] Runway, "Gen-1: Video generation from a single image," 2023, <https://research.runwayml.com/gen1>.
- [43] J. Z. Wu *et al.*, "Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation," *arXiv preprint arXiv:2212.11565*, 2022.
- [44] Anonymous, "Video diffusion models," 2023, <https://video-diffusion.github.io>.
- [45] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *Advances in Neural Information Processing Systems*, vol. 30, 2017. [Online]. Available: <https://arxiv.org/abs/1706.08500>
- [46] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [47] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *The 37th Asilomar Conference on Signals, Systems & Computers*, vol. 2, 2003, pp. 1398–1402.
- [48] D. M. W. Powers, "Evaluation: From precision, recall and f-measure to roc, informedness, markedness and correlation," *arXiv preprint arXiv:2010.16061*, 2020. [Online]. Available: <https://arxiv.org/abs/2010.16061>
- [49] A. M. Alaa, B. van Breugel, E. Saveliev, and M. van der Schaar, "How faithful is your synthetic data? samplelevel metrics for evaluating and auditing generative models," *arXiv preprint arXiv:2102.08921*, 2021. [Online]. Available: <https://arxiv.org/abs/2102.08921>
- [50] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography Conference*. Springer, 2006, pp. 265–284.
- [51] T. I. Amosa, P. Sebastian, L. I. Izhar, O. Ibrahim, L. S. Ayinla, A. A. Bahashwan, A. Bala, and Y. A. Samaila, "Multi-camera multi-object tracking: A review of current trends and future advances," *Neurocomputing*, vol. 552, p. 126 558, 2023.
- [52] X. Yang, J. Ye, J. Lu, C. Gong, M. Jiang, X. Lin, W. Zhang, X. Tan, Y. Li, X. Ye *et al.*, "Box-grained reranking matching for multi-camera multi-target tracking," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 3096–3106.
- [53] D. Papp, D. Lovas, and G. Szűcs, "Object detection, classification, tracking and individual recognition for sea images and videos." in *CLEF (Working Notes)*, 2016, pp. 525–533.
- [54] J. Rutinowski, H. Youssef, A. Gouda, C. Reining, and M. Roidl, "The potential of deep learning based computer vision in warehousing logistics," *Logistics Journal: Proceedings*, vol. 2022, no. 18, 2022.
- [55] D. Chiaravalli, G. Palli, R. Monica, J. Aleotti, and D. L. Rizzini, "Integration of a multi-camera vision system and admittance control for robotic industrial depalletizing," in *2020 25th IEEE international conference on emerging technologies and factory automation (ETFA)*, vol. 1. IEEE, 2020, pp. 667–674.
- [56] S. Wang, H. Sheng, Y. Zhang, D. Yang, J. Shen, and R. Chen, "Blockchain-empowered distributed multicamera multitarget tracking in edge computing," *IEEE Transactions on Industrial Informatics*, vol. 20, no. 1, pp. 369–379, 2023.
- [57] X. Gong, X. Tan, and Y. Xiang, "Contrastive mean teacher for intra-camera supervised person reidentification," *IEEE Transactions on Circuits and Systems for Video Technology*, 2024.
- [58] Y. Jeon, A. S. Kulinan, T. Kim, M. Park, S. Park *et al.*, "Vision-based motion prediction for construction workers safety in real-time multi-camera system," *Advanced Engineering Informatics*, vol. 62, p. 102 898, 2024.
- [59] J. Kim, W. Shin, H. Park, and J. Baek, "Addressing the occlusion problem in multi-camera people tracking with human pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 5463–5469.
- [60] L. Fei and B. Han, "Multi-object multi-camera tracking based on deep learning for intelligent transportation: A review," *Sensors*, vol. 23, no. 8, p. 3852, 2023.
- [61] G. Szűcs, R. Borsodi, and D. Papp, "Multi-camera trajectory matching based on hierarchical clustering and constraints," *Multimedia Tools and Applications*, vol. 83, no. 15, pp. 44 879–44 902, 2024.
- [62] L. Wang, X. Zhang, H. Su, and J. Zhu, "A comprehensive survey of continual learning: Theory, method and application," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 8, pp. 5362–5383, 2024.
- [63] J. Orgil, S. Karungaru, and K. Terada, "Continual learning: linear layer classifier concatenation using image processing transform functions," in *Fifteenth International Conference on Quality Control by Artificial Vision*, vol. 11794. SPIE, 2021, pp. 338–343.
- [64] R. Yang, S. Wang, H. Zhang, S. Xu, Y. Guo, X. Ye, B. Hou, and L. Jiao, "Knowledge decomposition and replay: A novel cross-modal image-text retrieval continual learning method," in *Proceedings of the 31st ACM International Conference on Multimedia*, 2023, pp. 6510–6519.
- [65] G. Raghavan, B. Tharwat, S. N. Hari, D. Satani, R. Liu, and M. Thomson, "Engineering flexible machine learning systems by traversing functionally invariant paths," *Nature Machine Intelligence*, vol. 6, no. 10, pp. 1179–1196, 2024.
- [66] W. Sun, R. Al Kontar, J. Jin, and T.-S. Chang, "A continual learning framework for adaptive defect classification and inspection," *Journal of Quality Technology*, vol. 55, no. 5, pp. 598–614, 2023.
- [67] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes," 2018. [Online]. Available: <https://arxiv.org/abs/1711.00199>

- [68] B. Wen, W. Yang, J. Kautz, and S. Birchfield, "Foundationpose: Unified 6d pose estimation and tracking of novel objects," 2024. [Online]. Available: <https://arxiv.org/abs/2312.08344>
- [69] G. Wang, F. Manhardt, F. Tombari, and X. Ji, "Gdrnet: Geometry-guided direct regression network for monocular 6d object pose estimation," 2021. [Online]. Available: <https://arxiv.org/abs/2102.12145>
- [70] P. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.
- [71] Y. Zhao, W. Lv, S. Xu, J. Wei, G. Wang, Q. Dang, Y. Liu, and J. Chen, "Detrs beat yolos on realtime object detection," 2024. [Online]. Available: <https://arxiv.org/abs/2304.08069>
- [72] S. Garrido-Jurado, R. Muñoz-Salinas, F. Madrid-Cuevas, and M. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, no. 6, pp. 2280–2292, 2014. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0031320314000235>
- [73] Z.-Q. Zhao, P. Zheng, S. tao Xu, and X. Wu, "Object detection with deep learning: A review," 2019. [Online]. Available: <https://arxiv.org/abs/1807.05511>
- [74] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," 2016. [Online]. Available: <https://arxiv.org/abs/1506.02640>
- [75] G. Jocher, A. Chaurasia, J. Qiu *et al.*, "Yolov8: Cuttingedge object detection models," 2023, <https://github.com/ultralytics/ultralytics>.
- [76] R. Mustafovski, T. Shuminoski, and A. Risteski, "YOLOv8-based firearm detection for real-time video surveillance application," *Infocommunications Journal*, vol. XVIII, no. 2, pp. 72–78, 2026.
- [77] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Computer Vision – ECCV 2016*. Springer International Publishing, 2016, p. 21–37. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-46448-0_2
- [78] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," 2016. [Online]. Available: <https://arxiv.org/abs/1506.01497>
- [79] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft coco: Common objects in context," 2015. [Online]. Available: <https://arxiv.org/abs/1405.0312>
- [80] M. Everingham, L. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010. [Online]. Available: <http://dblp.uni-trier.de/db/journals/ijcv/ijcv88.html#EveringhamGWWZ10>
- [81] R. Barandela, R. M. Valdovinos, J. S. Sánchez, and F. J. Ferri, "The imbalanced training sample problem: Under or over sampling?" in *Joint IAPR international workshops on statistical techniques in pattern recognition (SPR) and structural and syntactic pattern recognition (SSPR)*. Springer, 2004, pp. 806–814.
- [82] A. Ali, S. M. Shamsuddin, and A. L. Ralescu, "Classification with class imbalance problem," *Int. J. Advance Soft Compu. Appl.*, vol. 5, no. 3, pp. 176–204, 2013.
- [83] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.
- [84] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of big data*, vol. 6, no. 1, pp. 1–48, 2019.
- [85] C. Kofler, C. A. Dohr, J. Dohr, and A. Zernig, "Datacentric model development to improve the cnn classification of defect density sem images," in *IECON 2022–48th Annual Conference of the IEEE Industrial Electronics Society*. IEEE, 2022, pp. 1–6.
- [86] M. Tan and Q. Le, "Efficientnetv2: Smaller models and faster training," in *International conference on machine learning*. PMLR, 2021, pp. 10 096–10 106.
- [87] A. Poniatowski, N. Gentner, M. Barusco, D. D. Pezze, S. Salti, and G. A. Susto, "Domain adaptation for image classification of defects in semiconductor manufacturing," *IEEE Transactions on Automation Science and Engineering*, vol. 23, pp. 3818–3828, 2026.
- [88] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2223–2232.
- [89] D. Berthelot, R. Roelofs, K. Sohn, N. Carlini, and A. Kurakin, "Adamatch: A unified approach to semisupervised learning and domain adaptation," 2022. [Online]. Available: <https://arxiv.org/abs/2106.04732>
- [90] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.
- [91] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [92] K. Roth, L. Pemula, J. Zepeda, B. Schölkopf, T. Brox, and P. Gehler, "Towards total recall in industrial anomaly detection," 2022. [Online]. Available: <https://arxiv.org/abs/2106.08265>
- [93] S. Lee, S. Lee, and B. C. Song, "Cfa: Coupled-hypersphere-based feature adaptation for target-oriented anomaly localization," 2022. [Online]. Available: <https://arxiv.org/abs/2206.04325>
- [94] S. L. Takács and K. Kajdy, "Convolutional and variational autoencoders with supervised classifiers for RF spectrum anomaly detection," *Infocommunications Journal*, vol. XVIII, no. 2, pp. 55–60, 2026.
- [95] O. Ronneberger, P. Fischer, and T. Brox, "Unet: Convolutional networks for biomedical image segmentation," 2015. [Online]. Available: <https://arxiv.org/abs/1505.04597>
- [96] A. Dutta and A. Zisserman, "The VIA annotation software for images, audio and video," in *Proceedings of the 27th ACM International Conference on Multimedia*, ser. MM '19. New York, NY, USA: ACM, 2019. [Online]. Available: <https://doi.org/10.1145/3343031.3350535>
- [97] K. Sohn, D. Berthelot, C.-L. Li, Z. Zhang, N. Carlini, E. D. Cubuk, A. Kurakin, and H. Zhang, "Fixmatch: Simplifying semi-supervised learning with consistency and confidence," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

- [98] Y. Grandvalet and Y. Bengio, "Semi-supervised learning by entropy minimization," *Advances in neural information processing systems*, vol. 17, 2005.
- [99] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," *Advances in neural information processing systems*, vol. 30, 2017.
- [100] J. E. Van Engelen and H. H. Hoos, "A survey on semisupervised learning," *Machine Learning*, vol. 109, pp. 373–440, 2020.
- [101] A. Oliver, A. Odena, C. Raffel, E. D. Cubuk, and I. Goodfellow, "Realistic evaluation of deep semisupervised learning algorithms," in *Advances in neural information processing systems*, vol. 31, 2018.
- [102] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," in *International Journal of Computer Vision (IJCV)*, vol. 88, no. 2. Springer, 2010, pp. 303–338.
- [103] Y. Zhao *et al.*, "Rt-detr: Real-time detection transformer," *arXiv preprint arXiv:2304.08069*, 2023.

Anomaly and Fault Detection and Predictive Maintenance: a Review on Industrial Applications

Davide Frizzo¹, Valentina Zaccaria¹, David Petrovic¹, Francesco Borsatti¹, Davide Sartor¹, Felice Russo²,
Genoveffa Marcario², and Gian Antonio Susto¹

Abstract—Diagnostics and prognostics are crucial components of Smart Factories. Their continuous development is driven by the Industry 4.0 and 5.0 revolution, which relies on Artificial Intelligence (AI), Internet of Things (IoT) and Industrial Internet of Things (IIoT). This survey provides a comprehensive review of AI-enhanced Anomaly Detection (AD), Fault Detection (FD), and Predictive Maintenance (PdM) frameworks, considering both state-of-the-art methodologies and practical applications across diverse industrial domains. Key aspects covered include benchmark datasets, the integration of IoT/IIoT infrastructures, and the economic implications of diagnostics and prognostics in smart manufacturing. Moreover, the emphasis on the human-centric vision of Industry 5.0, the significance of Explainable Artificial Intelligence (XAI) and Uncertainty Quantification (UQ) for fostering transparent and trustworthy industrial AI systems is highlighted. Finally, research gaps and future research directions are considered, notably in model transferability, computational efficiency, and human-centric AI development for real-world deployment.

Index Terms—Anomaly Detection, Fault Detection, Predictive Maintenance, Internet of Things, Industrial Internet of Things, Industry 4.0, Industry 5.0, Machine Learning, Deep Learning

I. INTRODUCTION

In recent years the phenomena of Industry 4.0 and 5.0 have been continuously gaining popularity. In particular Industry 4.0 [1], [2] driven by the complementarity of various emerging technologies (i.e. nanotechnology, biotechnology, AI, and others) gave birth to the concept of *Smart Factory*. This refers to an industrial system that learns as it works, undergoing a process of continuous adaptation and optimization. The Fourth Industrial revolution became possible due to multiple technological advancements with the most important certainly being the development of IoT and IIoT based technologies. These innovations are revolutionizing the connectivity between hardware and software by building Cyber Physical Systems (CPS), an example of which is Digital Twin (DT) [3].

As an example the 2021 report by Deloitte [4] considers IoT as the most impactful Industry 4.0 technology, followed by AI and Cloud Computing. The result has been made possible by the development of cheaper and more reliable sensors. Additionally, the availability of cloud infrastructures enables the storage, processing and sharing of the data collected by these sensors in real time without the need of a centralized data center, further automatizing production and maintenance processes.

The proliferation of Big Data and IoT has revolutionized industrial processes by facilitating the collection and processing of vast amounts of data. The data-rich environment arising from the application of these emerging technologies provides a fertile ground for integrating AI algorithms, enabling the automation of crucial manufacturing tasks such as diagnostics and prognostics [5].

As a consequence, this technology is being widely adopted in the industrial domain for the deployment of diagnostics and prognostics methodologies in order to provide an efficient and cost-effective way to reactively and proactively monitor the health of industrial systems. IIoT infrastructures are essentially the main source of data used to produce the diagnostic and prognostic decisions with the main goals of identifying anomalous behavior, faults and their underlying causes, reducing downtime, maintenance costs and extending the lifespan of the industrial assets.

This study focuses on providing practitioners and researchers a comprehensive review on the state-of-the art and applications of AI-enhanced diagnostic and prognostic frameworks which are fastly becoming the standard in smart manufacturing: AD, FD and PdM.

AD and FD, which will be introduced in Sections II and III, are diagnostics approaches. Their aim is to first identify the presence of anomalous behaviors or faults in the manufacturing process (i.e. AD and Fault Detection). Successively it will be analyzed if the type, root cause and location of such malfunctioning can be identified (i.e. Fault Classification).

On the other hand, PdM, presented in IV, is a prognostic strategy. It comes later in the production pipeline and its focus is on forecasting the future state, the Remaining Useful Life (RUL) or the likelihood of failure of an equipment where a fault or a malfunctioning process has been diagnosed. Even though AD, FD and PdM are defined as three separated tasks, they are closely interconnected: diagnostic results produced by AD or FD are the inputs of a prognostic system, such as one that performs RUL estimation.

The main characteristics of the three diagnostic and prognostic frameworks surveyed in this chapter are presented in Table I.

While the goals of Industry 4.0 focus on economical sustainability and operational efficiency, Industry 5.0 shifts the vision towards the central role of humans and societal needs [7], [8], [9], [10]. The focus has shifted toward to the development of new technologies that prioritize human well-being and interaction within industrial processes. To address this paradigm shift, particularly in the context of AD, FD,

¹Department of Information Engineering, University of Padova, Via Giovanni Gradenigo, 6b, Padova, Italy.

²LFoundry SRL, Via Antonio Pacinotti 7, Avezzano, Italy.

TABLE I
MAIN CHARACTERISTICS OF AD, FD AND PDM FRAMEWORKS

Framework	Field	Paradigm	Task	Data modalities
Anomaly Detection	Diagnostic	Un- and Semi-supervised	Classification	Tabular, Image
Fault Detection	Diagnostic	Supervised and Semi-supervised	Multi-Class Classification	Tabular, Multivariate Time Series
Predictive Maintenance	Prognostic	Supervised and Semi-supervised	Regression, Classification [6]	Multivariate Time Series

and PdM, this paper also surveys XAI (Section IV-C) and UQ (Section IV-B2) approaches, which are crucial for fostering human-centric and transparent industrial AI systems.

A. Benchmark Datasets

One of the main challenges when it comes to the integration of AD, FD and PdM in the industrial domain is the availability of high quality datasets. These are needed to train and validate the AI models used to perform diagnostics or prognostics tasks. In industrial settings, in fact, historical data of faults or machine breakages are usually rare since companies are willing to avoid the high-costly and catastrophic situations generated by such events. Moreover also privacy reasons have to be considered. Furthermore, in the past, humans were primarily responsible for constructing maintenance and fault reports on machines. This often resulted in data that is not well-structured or standardized, making it challenging for AI models to effectively process and learn from. Consequently, there is a significant need to build comprehensive benchmark datasets to enable researchers to compare and validate different proposed methods on a common and consistent ground.

Because of the data collection challenges described above these datasets are usually obtained from simulated data, aiming to mimic the degradation process which leads to the machine failure or specific faults.

Even though AD, FD and PdM are quite different tasks they are naturally connected in the manufacturing pipeline and thus usually these datasets can be easily manipulated to be adapted to any of these problems.

In particular:

- For the AD labels contained in the dataset can be ignored because of the unsupervised nature of the task.
- In FD problems usually labels characterizing different faults are available.
- In PdM workflows a degradation model, such as the ones presented in IV-B2, or an Health Index (HI) can be exploited to construct the labels.

Here is a list of the benchmarks datasets most commonly used in the literature:

- **Commercial Modular Aero-Propulsion System Simulation:** This is probably the most common and well documented benchmark dataset in the literature for the tasks of FD and PdM. It is produced by the Commercial Modular Aero-Propulsion System Simulation [11]. It contains sensor measurements from turbofan engines used in passenger aircraft vehicles. In the simulations each

engine starts from random health states which, after a short period of time, start to degrade towards engine failure. The dataset is already divided into a training a training set (sensor measurements available until the end of life (i.e. RUL = 0)) and test set (sensor measurements stop some time before failure). Each dataset is formed by 24 features, 3 being the operations conditions and the remaining 21 being sensor measurements. The dataset is further divided into four sub datasets (i.e. FD001, FD002, FD003, FD004) with different numbers of faults and operating conditions making the RUL prediction task more challenging. Additional challenges posed by this dataset are the heterogeneous lengths of the run to failure life cycles of different engines and the different characteristics of the engines due to manufacturing variations. Moreover multiple degradation modes can develop on different engines even if starting from the same RUL value making it extremely challenging for data-driven models to produce accurate predictions [12].

- **New Commercial Modular Aero-Propulsion System Simulation: The New Commercial Modular Aero-Propulsion System Simulation (N-CMAPSS)** [13], [14] (also referred to as C-MAPSS-2) is an improved version of the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) dataset, introduced in [14]. The improvements include the fact that several common maneuvers performed by commercial flights are now recorded in the data, the equipment degradation onset is related to the operational history and the dataset now also include labels regarding faults. As a consequence, this dataset can be used to benchmark models designed for the FD task.
- **PHM18:** This dataset was introduced for the PHM Data Challenge 2018 [15] and it contains data about Ion Milling etching tools and the aim of the challenge was the prediction of faults on these kind of equipments. Ion Milling is a crucial step in the process of manufacturing integrated circuits, in particular in the production of wafers. This operation, refer to [15] for a detailed description, is quite complex and multiple failures can occur. It is thus beneficial to predict when and where the failures occur to strategically schedule equipment's downtime to intervene with maintenance operations. Thus this dataset can be employed both for the FD and the PdM tasks. The dataset contains time series data regarding the execution of 20 ion milling machines described with 3 failure causes and 22 features. Labels for the three failure

types are provided while the time corresponding to the machine's end of life (needed for the RUL estimation task) is provided indirectly as the time when the machine is switched off by the operator to perform a maintenance intervention. This poses a challenge since the actual time when the failure start may occur much earlier than the failure time and it is not provided. Moreover failure times are not available in case the equipment did not fail within a specified time horizon. As for C-MAPSS the data are splitted into a training and a test set and a specific score function is designed to have an objective evaluation of the solutions proposed by the competition's participants.

- **Lithium-ion batteries:** Another common field of application for PdM methodologies is the one of lithium-ion batteries (i.e. Section IV-D1), used to power electric vehicles. A common benchmark dataset used in this field is contained in the NASA Prognostics Center of Excellence [16] which is provided by National Aeronautics and Space Administration (NASA) and contains charge and discharge data of lithium-ion batteries. Three batteries are considered, all at a capacity of 2 Ah. Charging and discharging experiments are performed on the batteries. Experiments stopped when the battery capacity dropped below a predefined threshold (Acceptable Performance Threshold (APT) of 70%). The battery capacity is recorded at the end of each discharge cycle.
- **Rolling element bearing:** Rolling element bearings constitute one of the main causes of rotating machines failures. There are several benchmark dataset which can be used to benchmark PdM methodologies in this field. One of the most common is the IEEE PHM Challenge 2012 bearing dataset produced through the PRONOSTIA [17] experimental platform, whose aim is to provide real experimental data on rolling bearings degradation. This platform provides degradation data for all possible type of defects and can produce bearing degradation data under different operating conditions. A rotating speed sensor and force sensor are used to monitor the operating conditions while health monitoring is conducted through the collection of temperature and vibration signals through accelerometers. The sampling frequency of these sensors is set in order to capture the entirety of the frequency spectrum the bearing during the whole degradation period. Another benchmark dataset commonly used in the application of FD and PdM approaches on bearings is the Intelligence Maintenance Systems (IMS) dataset [18]. The experimental setting consists of four bearings installed on a shaft. Each bearing is equipped with two Sensitivity Quartz ICP accelerometers measuring the x and y coordinates of the acceleration to which each bearing is subjected. Three datasets are available: both accelerometer data are considered in dataset 1, while only one accelerometer is considered for datasets 2 and 3. This dataset is characterized by a very high sampling frequency of 20 kHz. The three provided datasets correspond to three different experiments which ended with different defects on one or multiple bearings which are used to

construct the failure labels. Other popular benchmark dataset for rolling element bearings are Case Western Reserve University (CWRU) and [19].

- **Tennessee Eastman Process (TEP)**
Dataset: The TEP [20] dataset is typically used as a benchmark for FD. It collects data from Tennessee Eastman (TE) [21], a crucial process for the production of fibers, chemicals and advanced materials for every day purpose. This is a synthetic dataset producing normal and faulty runs. TEP also comes with ground truth knowledge of the main root causes of failures, making it a commonly used benchmark dataset for XAI models [22], [23].

For a more comprehensive list of available benchmark datasets refer to [24].

B. Scopes and Topics of interest

The research fields of AD, Fault Detection and Diagnosis (FDD) and PdM are very wide, containing applications using multiple learning frameworks based on different data modalities. In order to provide a general overview of the field and its application on the industrial world we decided to focus on a portion of topics which we think are more related and interesting for Industry 4.0 and 5.0.

Before the emergence of data-driven approaches, several methods were developed for diagnostics and prognostics, typically relying on domain expertise or mathematical models representing the underlying physical processes. While these methods can provide more than acceptable results they require to communicate with domain expert and to develop overly complex physical models which dramatically increase the computational requirements of the final system. Data-driven solutions are more efficient since, being based just on the relationships between the input and output variables, do not require the design of complex representations of the equipment.

For the above mentioned reasons, we decide to focus mainly on data-driven approaches (i.e. Machine Learning (ML) and Deep Learning (DL) based) since we believe in their superiority in terms of efficiency and predictive accuracy with the respect to less recent methodologies based on domain expertise and non data-driven AI solutions which are leaving the floor for learning-based approaches as the new standard for diagnostics and prognostics.

For what concerns the data modalities for all the three frameworks we restrict our study to tabular or sequential data excluding domains such as images, video, and other high-dimensional perceptual data, we refer the interested readers to [1] [2], which have been published jointly with this review.

The survey is organized as follows: Sections II, III, IV are dedicated to the three frameworks: AD, FDD and PdM, respectively. Each one of these sections provides an initial theoretical background on the specific task (i.e. Sections II-A1, III-A1, IV-A), followed by an in depth description of the state of the art methods (i.e. Sections II-A2, III-A2, IV-B) used to solve them. In Sections II-B, III-B, IV-D an overview of the applications in

industrial environments is outlined. Remaining in the realm of industrial applications, Section V describes some use cases in the context of the Artificial Intelligence in Manufacturing leading to Sustainability and Industry 5.0 (AIMS5.0) project providing a real world example of challenges in integrating the three presented frameworks in a sustainable industrial process. Section VI is dedicated to the identification of research gaps. Finally, the study concludes in Section VII.

II. ANOMALY DETECTION

In this section theoretical foundations and state of the art methods for the AD framework will be presented.

A. AD Methods and Theoretical Foundations

1) *AD Definitions*: It is necessary to formalize the notion of anomaly detection by defining the task of identifying such instances within data.

Definition 1. Anomaly detection, in the context of machine learning, is generally an unsupervised task, though allowing weakly or semi-supervised variants. Its primary objective is to infer binary labels (*normal* versus *anomalous*) or to compute continuous anomaly scores that quantify the degree of deviation from normal behavior.

In general, given the absence of extensive ground-truth annotations, the detection task is treated as unsupervised with respect to the anomalous class, implying that the notion of anomaly must be inferred directly from the data. Consequently, the problem is inherently ill-posed, as the absence of clearly defined criteria and ground-truth labels introduces ambiguity in both the formulation and evaluation of the task.

Anomaly: Numerous studies on anomaly detection provide their own definitions of what constitutes an anomaly. For example, D. H. Hawkins defines an anomaly as “an observation which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism” [25]. Or put more simply, anomalies are “cases that are in some way unusual and do not appear to fit the general patterns of the dataset” [26].

The definition of anomaly adopted in this work, based on Definition 1, is stated as follows.

Definition 2. An anomaly is a sample identified by both its rarity and significant deviation from the normal data points.

To formalize this definition, it is necessary to clarify the notion of rarity, to specify what constitutes a significant deviation, and to define which data points represent normality.

In Definition 2, rarity is identified as one of the defining characteristics of anomalous samples. This property is typically interpreted as the anomaly residing in a low-density region relative to the overall data distribution. However, as noted in the comprehensive review of anomaly types by R. Foorthuis [27], a more nuanced interpretation allows for anomalies to occur in high-density regions, for instance, when the majority of normal observations themselves occupy low-density areas.

In an unsupervised setting, anomalies are identified by their deviation from the majority of the data. In this review, two main scenarios observed in the literature are presented, each describing a different relationship between normal and anomalous data.

- 1) *Outlier*: An outlier is an anomaly present within the data set used for modeling. This implies the available data is contaminated with anomalies, and the detection algorithm must be able to distinguish them from the majority of the data, which is considered normal [28].
- 2) *Novelty*: Given an initial dataset assumed to be free of anomalies, a novelty refers to a previously unseen observation that significantly deviates from the established normal distribution and is therefore identified as anomalous by the detection algorithm [28]. Since the modeling of the data is performed exclusively on normal instances, *novelty detection* can be regarded as a semi-supervised learning task [29], [30], where the model learns the characteristics of normal behavior and subsequently distinguishes deviations from this learned representation as novel or anomalous patterns.

The concepts of outlier and novelty are also discussed in the survey by V. Hodge *et al.* [29], which further introduces a third scenario in which both anomalous and normal data are labeled. This last case, however, falls outside the scope of the present definition of anomaly detection and is instead referred to as fault detection when label information is available.

Given the two scenarios—outlier and novelty—which describe how data is presented to the algorithm, the literature further introduces several types and categories of anomalies. These classifications are discussed in the following section.

Types of Anomaly: The survey by Chandola *et al.* [28] classifies anomalies into three principal categories: point, contextual, and collective anomalies. A point anomaly corresponds to an individual observation that significantly deviates from the expected data distribution. Contextual anomalies typically occur in time series or sequential data, where an observation may appear normal in isolation but is anomalous when considered within its contextual attributes. Collective anomalies, on the other hand, refer to groups of observations that may individually appear normal but collectively exhibit abnormal behavior, often due to relational dependencies or patterns among them. In contrast to contextual anomalies—where the context is derived from intrinsic data attributes—collective anomalies are defined by the relationships between multiple data instances.

The work by Foorthuis [27], [26] provides a comprehensive and systematic characterization of anomaly types, emphasizing their diverse nature and the multiple dimensions along which they can be distinguished. Anomalies may exhibit different characteristics, which can be categorized along several axes, as described below.

Two anomaly types can be identified from a structural point of view:

- *Atomic* (also called *point anomalies*): individual data instances that deviate from the normal distribution.
- *Aggregate* (also called *collective anomalies*): anomalies that arise from groups of related samples, where the

collection as a whole deviates from normal behavior, even if individual instances appear normal.

Two anomaly types can be defined on their dimensionality:

- *Univariate*: anomalies identifiable from a single feature or variable.
- *Multivariate*: anomalies that emerge from specific combinations of multiple features, where each individual feature may appear normal, but their joint occurrence is atypical.

Anomaly groups can be also dependent on the data type:

- *Numeric*: anomalies defined over continuous or discrete numerical attributes.
- *Categorical*: anomalies occurring within categorical or symbolic data.
- *Mixed*: anomalies described by both numerical and categorical features.

In essence, anomalies can be described by any combination of these characteristics, for example, a single faulty sensor reading in an industrial process (atomic, univariate, numeric) or a coordinated network intrusion involving multiple correlated events across hosts (aggregate, multivariate, mixed).

2) *AD Methods*: Anomaly detection methods based on ML can be categorized according to their learning paradigm or model architecture. Following the taxonomy proposed by L. Ruff *et al.* [31], a primary distinction is drawn between *shallow* and *deep* learning approaches. Furthermore, the same authors classify existing methods into four fundamental categories based on their underlying principles: *classification-based*, *probability-based*, *reconstruction-based*, and *distance-based* approaches, which are better described in III.

Shallow machine learning is often referred to as traditional machine learning and typically relies on explicit feature engineering, where domain knowledge is used to design relevant input representations. In contrast, *deep learning* approaches employ end-to-end models that jointly learn feature representations and perform the detection task within a unified architecture.

Given that anomalies are inherently rare and unpredictable, extensive manual feature engineering often implies the use of labeled or domain-specific knowledge. Such cases align more closely with supervised *fault detection* and are therefore discussed separately in Section III-A.

Since extensive feature engineering is not generally applicable within the anomaly detection framework, shallow methods are instead characterized by the use of models that do not rely on deep neural architectures. These methods operate directly on the available feature space and typically include approaches such as support vector machines (SVM), k-nearest neighbors (k-NN), and ensemble-based techniques like Isolation Forests. Such algorithms focus on identifying deviations in data distributions or local density structures without the hierarchical representation learning typical of deep models.

Taxonomy of AD methods: Anomaly detection methods are often specialized depending on the data modality. This survey focuses on **tabular** and **time series** data, which present distinct structural and modeling challenges, summarized in Table II.

Other data modalities such as visual, textual, graph, or 3D point cloud data introduce different challenges and modeling paradigms, but are beyond the scope of this survey.

Table III shows the main categories of anomaly detection methods and their core principles.

In subsequent sections, each category will be further divided into **shallow** and **deep** methods, reflecting differences in model capacity, feature learning, and scalability.

3) *Density-Based AD*: Density-based anomaly detection methods operate on the principle that normal instances occur in high-density regions of the data distribution, while anomalies reside in sparse or isolated areas. These approaches estimate the probability density function of the data, either explicitly or implicitly, and identify samples located in low-density regions as anomalous. The core assumption underlying these methods is that the data-generating process produces normal observations with higher probability than anomalous ones.

Shallow Density-Based Approaches: Shallow density-based methods can be broadly categorized into three families: local density estimation approaches, histogram-based methods, and probabilistic mixture models.

Shallow density-based methods span a spectrum from local density approaches that capture fine-grained neighborhood structure to histogram-based methods that sacrifice multivariate dependencies for scalability. This trade-off is reflected in computational complexity: local methods such as LOF and LDF typically require $O(n^2)$ operations for pairwise distance computations, though approximations and indexing structures can reduce this in practice. Histogram-based approaches achieve linear or near-linear complexity, making them suitable for large-scale applications. LOCI additionally introduces an approximate algorithm with linear complexity through box-counting methods.

Local methods excel at detecting contextual anomalies in datasets with varying density clusters but require careful parameter tuning. Histogram-based approaches provide substantial speedups suitable for large-scale or real-time detection but are generally limited to global outliers unless combined with spatial reasoning. Copula and mixture model approaches offer principled statistical foundations with varying degrees of distributional assumptions. The choice among these methods depends on computational constraints, the nature of expected anomalies (global versus local), and the validity of underlying distributional assumptions.

Local Density Estimation. A foundational approach in density-based anomaly detection is the Local Outlier Factor (LOF) [32], which introduced the concept of quantifying the degree to which an observation deviates from its local neighborhood rather than treating outlier detection as a binary classification problem. LOF computes a local reachability density for each point and compares this density to those of neighboring points, addressing datasets with clusters of varying densities where global methods fail.

Building upon LOF, the Local Correlation Integral (LOCI) method [33] introduced the Multi-Granularity Deviation Factor (MDEF), providing automatic, statistically-principled outlier determination without user-specified cutoffs through standard deviation-based flagging across multiple scales. The Local

TABLE II
DATA MODALITIES CONSIDERED IN THIS SURVEY AND THEIR ASSOCIATED CHALLENGES.

Modality	Description	Challenges
Tabular	Fixed-length vector of features, without inherent order or temporal dependency.	Mixed-type features, high dimensionality, missing data.
Time series	Ordered sequence of observations over time, with temporal dependencies.	Temporal correlations, seasonality, irregular sampling, concept drift.

TABLE III
CATEGORIES OF ANOMALY DETECTION METHODS AND THEIR CORE PRINCIPLES

Category	Core Principle
Density	Estimate the probability density function of the data and identify samples located in low-density regions as anomalies. These methods assume that normal data points occur in dense regions, while anomalies lie in sparse or isolated areas. Approaches range from classical statistical estimations to neural or flow density models.
Distance	Compute dissimilarities between samples using a distance metric and label points that deviate significantly from their neighbors as anomalies. Such methods rely on the assumption that anomalies are distant from most other data points in input space, feature space, or along temporal trajectories.
Ensemble	Combine the outputs or decision functions of multiple detectors to improve robustness and reduce model bias. Ensembles exploit model diversity through random subsampling, feature bagging, or heterogeneous learners to capture different anomaly characteristics and achieve better generalization.
Reconstruction	Learn a model capable of reconstructing normal data patterns and detect anomalies as samples with large reconstruction errors. The underlying hypothesis is that models trained on normal data cannot accurately reproduce unseen or irregular patterns. Both linear (e.g., PCA) and nonlinear (e.g., autoencoders, GANs) reconstruction frameworks are common.
Clustering	Group data into clusters and identify anomalies as points that do not belong to any cluster or are far from cluster centroids. The intuition is that normal data form compact clusters, whereas anomalies are isolated or weakly associated with any group. Methods vary from classical k-Means and density clustering to deep clustering variants.
One-Class	Learn a boundary or manifold that encloses normal data and treats points outside this region as anomalies. These methods directly model the support of the normal distribution, often via kernel machines or deep representation learning, and are particularly suited when only normal samples are available for training.
Forecasting	Model temporal dependencies to predict future observations and identify anomalies as deviations between predicted and actual values. These methods assume that normal behavior follows predictable temporal patterns, while anomalies correspond to unpredictable deviations. Although forecasting detection is conceptually similar to reconstruction as both involve reproducing parts of the input it differs fundamentally in that time series data are inherently ordered, and prediction exploits the temporal dependency between successive observations.

Density Factor (LDF) [34] further refined this direction by providing a statistically grounded framework through variable-width kernel density estimation, handling multivariate distributions where normal points in sparse clusters may exhibit lower absolute density than outliers in dense regions.

Further developments include methods employing specialized kernels and weighted neighborhood formulations for improved robustness to parameter selection [35], frameworks that explicitly decouple density estimation from outlier scoring phases [36], and approaches that unify kernel density estimation with k -nearest neighbor methods through observation-specific kernels adapted to local covariance structure [37].

Histogram-Based Methods. Histogram-based approaches trade modeling sophistication for computational efficiency. The Histogram-Based Outlier Score (HBOS) [38] constructs independent univariate histograms for each feature and combines their density estimates to compute anomaly scores. By assuming feature independence, HBOS achieves substantial speedups over local density methods, though this assumption precludes detection of local outliers dependent on feature correlations.

To retain local outlier detection capability while maintaining efficiency, multivariate histogram approaches have been developed [39]. These methods identify and discard dense, uniform

regions where outliers are unlikely to exist, reducing the computational burden of subsequent local density calculations. The Lightweight On-line Detector of Anomalies (Loda) [40] demonstrates that an ensemble of weak detectors—one-dimensional histograms on random sparse projections—can achieve competitive performance with complex methods, while additionally supporting streaming data and providing anomaly explanations.

Copula and Mixture Models. COPOD [41] leverages copula theory to detect outliers by computing tail probabilities in multivariate distributions. The method constructs an empirical copula to separate marginal distributions from dependency structure, identifying outliers as observations with extremely small tail probabilities. COPOD is parameter-free, deterministic, and provides interpretability through dimensional outlier graphs quantifying each feature's contribution to abnormality.

Gaussian Mixture Models (GMM) offer a parametric alternative [42], fitting a mixture of Gaussian components and identifying outliers as points falling outside high-probability regions. While GMMs capture multi-modal distributions and provide probabilistic interpretability, they assume the data follows a Gaussian mixture distribution, which may not hold for complex real-world datasets.

Deep Density-Based Approaches: Deep density-based methods leverage neural networks to model complex, high-dimensional probability distributions that are intractable for shallow approaches. These methods can be organized into three main families: normalizing flow-based approaches, hybrid autoencoder-density models, and diffusion-based methods. Normalizing flows provide exact likelihood computation through invertible transformations, with variations including conditional flows for structured data, likelihood ratio construction, and pseudo-anomaly generation for classifier augmentation. Hybrid approaches such as DAGMM demonstrate that combining autoencoders with explicit density models through joint optimization outperforms decoupled alternatives. Diffusion models offer stable training dynamics and effective handling of contaminated training data through rejection schemes. Computational considerations vary across families: normalizing flows require invertible architectures with tractable Jacobian computation; DAGMM adds mixture model estimation overhead but avoids autoencoder pre-training; diffusion models incur costs from iterative denoising during both training and inference.

Normalizing Flow-Based Methods. Normalizing flows provide a framework for explicit density estimation through invertible transformations that map complex data distributions to simple base distributions (typically Gaussian). The change of variables formula enables exact likelihood computation, making flows particularly suitable for density-based anomaly detection where anomalies are identified as samples with low probability under the learned distribution.

Conditional normalizing flows extend this framework by factorizing joint probabilities along relevant dimensions [43]. For multivariate time series, factorization along temporal and attribute dimensions transforms complex joint density evaluation into tractable conditional density estimation, with attention mechanisms parameterizing the conditional information

to capture long-range dependencies and dynamic inter-variable correlations.

Normalizing flows also enable construction of fully data-driven likelihood ratios by estimating conditional densities in different data regions [44]. By computing the ratio of densities learned from potentially anomalous regions versus background regions, an optimal test statistic for anomaly presence can be derived, providing both classification capability and direct background estimation.

Beyond direct density evaluation, normalizing flows have been applied to generate pseudo-anomalous samples for training augmentation [45]. By sampling from low-density regions of a flow trained on normal data, synthetic anomalies can be generated to transform one-class classification into binary classification. This model-agnostic approach bridges one-class and two-class methods, enabling effective utilization of any available labeled anomalies while maintaining robustness to unknown anomaly types.

Emerging research explores quantum implementations of normalizing flows [46], leveraging the natural invertibility of unitary quantum gates. Quantum architecture search optimizes gate sequences to map data distributions to target distributions, with anomaly scores computed via divergence measures between quantum measurement distributions and the learned normal distribution.

Hybrid Autoencoder-Density Models. A limitation of using autoencoders alone for anomaly detection is that reconstruction error provides only an implicit density proxy. Hybrid approaches address this by combining dimensionality reduction with explicit density estimation in a unified framework.

The Deep Autoencoding Gaussian Mixture Model (DAGMM) [47] jointly optimizes a deep autoencoder with a Gaussian mixture model through end-to-end training. The key innovation is constructing a low-dimensional representation that concatenates encoded features with reconstruction error features, capturing both how feature correlations differ for anomalies and the reconstruction difficulty. An estimation network predicts mixture membership probabilities, replacing the traditional Expectation-Maximization algorithm and enabling differentiable training. This joint optimization ensures that dimensionality reduction directly serves the density estimation objective, avoiding the information loss inherent in decoupled two-stage approaches.

Diffusion-Based Methods. Diffusion models learn data distributions by training a neural network to reverse a gradual noising process. For anomaly detection, the denoising capability provides a natural density proxy: normal samples are easier to denoise than anomalous ones.

For tabular data, diffusion-based approaches address the challenge of training on contaminated (unlabeled) data through iterative anomaly rejection schemes [48]. By progressively identifying and removing likely anomalies based on denoising difficulty, the learned density increasingly focuses on the normal data distribution. Anomaly scores are computed by aggregating reconstruction losses across multiple diffusion timesteps, providing multi-scale detection that captures anomalies at different noise levels. This approach demonstrates

particular effectiveness on high-dimensional data where classical methods suffer from the curse of dimensionality.

4) *Distance-Based AD*: Distance-based anomaly detection methods operate on the principle that anomalies are significantly distant from most other data points in the feature space. These approaches compute dissimilarities between samples using a distance metric and identify points that deviate substantially from their neighbors as anomalous. The core assumption is that normal instances form cohesive structures where points are relatively close to one another, while anomalies reside in isolated regions far from the majority of observations.

Shallow Distance-Based Approaches: Shallow distance-based methods can be organized into two main families: nearest neighbor approaches and alternative dissimilarity measures. Nearest neighbor methods rely on distances to neighboring points as anomaly indicators, with various strategies for computing and aggregating these distances. Alternative dissimilarity approaches address fundamental limitations of geometric distance in the original feature space, either by introducing data-dependent measures that adapt to local density variations, by exploiting directional information rather than distance magnitudes, or by computing distances in robust subspaces resistant to contamination. The computational complexity of these methods ranges from $O(n^2)$ for classical k -NN to $O(n)$ for ensemble-based density estimators, with the choice depending on dataset scale and dimensionality characteristics.

Nearest Neighbor Methods.

The k -nearest neighbor (k -NN) framework forms the foundation of distance-based anomaly detection, where anomaly scores are derived from distances to neighboring points. Classical approaches compute the distance to the k -th nearest neighbor or aggregate distances across multiple neighbors, with larger values indicating potential anomalies.

A key challenge in nearest neighbor methods is parameter sensitivity, particularly the choice of distance threshold or neighborhood size. Hybrid approaches address this through adaptive, data-driven neighborhood definitions [49]. By computing the average pairwise distance across the dataset and using it as an automatic neighborhood radius, such methods eliminate the need for prior knowledge of data distribution. Recursive partitioning based on neighbor counts progressively concentrates candidate outliers, followed by k -NN distance refinement to extract true anomalies.

For multivariate time series, the choice of local distance metric within Dynamic Time Warping (DTW) significantly impacts detection performance. Mahalanobis distance-based DTW learns a distance matrix that captures feature correlations and assigns appropriate weights to different variables [50]. This approach addresses the limitation that standard DTW with Euclidean distance treats all variables equally, ignoring heterogeneous feature relevance and inter-variable correlations present in multivariate data.

Computational efficiency represents a significant concern for nearest neighbor methods, which traditionally require $O(n^2)$ distance computations. The LiNearN algorithm [51] achieves $O(n)$ time complexity by fundamentally reconsidering the density estimation problem: rather than computing nearest neighbors for every instance, it estimates density by

averaging over multiple local regions derived from small subsamples. This ensemble approach yields adaptive local regions that automatically adjust to varying density distributions without requiring global parameters.

Alternative Dissimilarity Measures.

A fundamental limitation of geometric distance is its inability to capture the data-dependent characteristic where two instances in a dense region should be judged more dissimilar than two instances at the same geometric distance in a sparse region. Mass-based dissimilarity addresses this by replacing geometric distance with probability mass [52]. The dissimilarity between two points is defined as the probability mass of the smallest region covering both instances under a hierarchical partitioning model, typically implemented using isolation trees. A critical departure from traditional distance measures is that mass-based dissimilarity exhibits non-constant self-dissimilarity: points in high-density regions have larger self-dissimilarity than points in sparse regions. This property enables detection of local anomalies in clusters of varying densities, where fringe points of dense and sparse clusters receive similar anomaly scores despite having vastly different absolute densities.

In high-dimensional spaces, distance-based methods suffer from the curse of dimensionality: the relative contrast between nearest and farthest neighbors converges to zero as dimensionality increases, regardless of data distribution. Angle-Based Outlier Detection (ABOD) addresses this by using the variance of angles between difference vectors rather than distance magnitudes [53]. The key insight is that angular relationships remain meaningful discriminators in high dimensions where distance comparisons deteriorate. Points deep within clusters have difference vectors pointing in all directions, yielding high angle variance, while outliers have most points clustered in limited directions, producing low angle variance. ABOD computes the variance of angles between all pairs of difference vectors from a query point, weighted inversely by the squared distances to reduce the influence of distant points. A significant advantage is the parameter-free nature of the method, directly producing outlier rankings without requiring neighborhood size or distance threshold specification.

An alternative strategy computes distances in transformed spaces designed to be robust to outlier contamination. Standard subspace projection techniques such as PCA are vulnerable because anomalous points distort the covariance matrix, corrupting the principal components. CoMadOut [54] addresses this by employing comedian PCA, a robust variant that replaces covariance with the comedian matrix based on median absolute deviation. Since median-based statistics have high breakdown points, the resulting principal components maintain correct orientation even under substantial contamination. Orthogonal distances to subspace axes then serve as anomaly indicators, with distribution-aware scoring through kurtosis weighting adapting decision boundaries to handle varying tail behaviors across different principal components.

Deep Distance-Based Approaches: Deep distance-based methods leverage neural networks to learn representations where distance computations become more meaningful for anomaly detection. These approaches address limitations of

computing distances in raw feature spaces by projecting data into learned latent spaces where normal samples cluster near a reference distribution or manifold, and anomalies manifest as distant points. The computational overhead of deep architectures is offset by the ability to handle high-dimensional data and capture complex patterns that shallow methods cannot model effectively.

Bidirectional GAN architectures enable learning low-dimensional latent representations where distances to a reference distribution serve as anomaly scores [55]. By training an encoder jointly with a generator through adversarial objectives and reconstruction losses, normal data is mapped to approximately follow a standard normal distribution in latent space. The Mahalanobis distance to this reference distribution simplifies to the Euclidean norm of encoded features, providing a computationally efficient anomaly score that requires only a single encoder forward pass at inference time.

For data-center-scale applications involving millions of machines with multivariate time series, learned representations enable tractable clustering through distribution-based distances [56]. Rather than computing distances between raw high-dimensional time series, RNN-VAE encoders compress data to low-dimensional latent representations, and machines are clustered based on Wasserstein distances between their latent representation distributions. This approach reduces the problem from training millions of individual models to training hundreds of cluster-specific models through transfer learning, achieving near-optimal accuracy with orders-of-magnitude speedup.

Self-supervised contrastive learning provides an alternative framework where distances in embedding space reflect transformation consistency [57]. Rather than relying on hand-crafted transformations appropriate only for images, learnable neural network transformations are jointly optimized with an encoder to produce diverse yet semantically consistent views of normal data. The training loss, which measures similarity between transformed and original samples in the learned embedding space, directly serves as the anomaly score, with anomalies exhibiting high scores due to inconsistency with learned transformation patterns.

5) Ensemble-Based AD: Ensemble-based anomaly detection methods combine multiple detectors to improve robustness and reduce model bias. These approaches exploit diversity through random subsampling, feature bagging, or heterogeneous learners to capture different anomaly characteristics. The core principle is that aggregating multiple weak detectors yields more reliable scores than any single model.

Shallow Ensemble-Based Approaches: Shallow ensemble methods comprise two main families: isolation-based approaches that build ensembles of partitioning structures to explicitly isolate anomalies, and selective combination methods that intelligently aggregate outputs from multiple base detectors based on estimated competence. Isolation methods achieve linear complexity through subsampling, while selective methods add competence evaluation overhead but avoid degradation from poorly performing members.

Isolation-Based Methods.

The isolation paradigm [58] exploits the observation that anomalies, being few and different, require fewer partitions to separate from normal data. Random recursive partitioning isolates anomalies closer to the root, with scores derived from average path length across an ensemble. The partitioning can use axis-parallel splits as in the original formulation or random hyperplanes [59] to handle non-axis-aligned structures. However, infinite hyperplane extension causes score inconsistencies as splits in dense regions affect distant sparse areas; finite hyperspheres [60] constrain each split to its local neighborhood, restoring proper locality.

Tree-based partitioning produces global scores that may miss local anomalies. Hypersphere-based isolation [61] centers regions on instances with radii from nearest-neighbor distances, automatically adapting to local density. Multi-scale approaches [62] combine global and local isolation by partitioning space into subregions with separate forests, capturing anomalies at different spatial scales.

For streaming data, static structures cannot adapt to distribution drift. Dynamic histogram-based approaches [63] replace fixed trees with structures that split and merge bins as data arrives and expires, enabling continuous adaptation. The randomness in isolation can also be guided by data statistics: kurtosis-weighted feature selection [64] directs splits toward heavy-tailed dimensions, while single-neighbor similarity in small subsamples [65] demonstrates that extreme simplicity suffices when aggregated across an ensemble.

Selective Combination Methods.

Generic ensemble combination assumes equal detector contribution, but competence varies across feature space and poor detectors can neutralize good ones. Selective methods address this by evaluating which detectors to trust for which instances.

Dynamic per-instance selection [66] evaluates competence within local neighborhoods, selecting detectors whose scores correlate with pseudo ground truth from ensemble statistics. When seeking a global subset, graph-based approaches [67] map rankings to similarity graphs where good detectors form dense communities of mutual agreement, enabling selection through community detection without pseudo ground truth. Competence estimation can be further refined near decision boundaries where detector disagreement matters most, using adaptive neighborhoods stretched along boundary directions [68] to improve validation in regions where class probabilities change rapidly.

Deep Ensemble-Based Approaches: Deep ensemble methods extend the isolation and combination principles to learned representation spaces, addressing limitations of shallow approaches that operate directly on raw features. The core insight is that anomalies difficult to isolate or detect in the original feature space may become separable in appropriately transformed spaces, with neural networks providing the non-linear mappings required for complex data structures.

Deep Isolation Forest [69] addresses two fundamental limitations of shallow isolation: the inability to isolate non-linearly separable anomalies and systematic score biases from axis-parallel partitioning artifacts. Rather than improving the splitting geometry as in extended or generalized variants, the method transforms the data space itself using randomly initial-

ized neural networks, making simple axis-parallel cuts in the transformed space equivalent to non-linear cuts in the original space. Critically, these networks require no training—casual initialization provides sufficient diversity across ensemble members, maintaining the computational efficiency that makes isolation methods practical. The approach generalizes across data types by simply changing the random architecture: fully connected networks for tabular data, recurrent layers for time series, and graph neural network layers for graph-structured data.

When labeled anomalies are available, even in small quantities, supervised deep ensembles can refine decision boundaries beyond what unsupervised methods achieve. Stacked autoencoders combined with ensembles of probabilistic neural networks [70] use deep feature extraction to project data into more separable representations, then employ multiple classifiers with varying smoothing parameters to balance the trade-off between false alarms and missed detections. The ensemble across hyperparameters provides diversity complementary to data-based diversity, and the framework naturally handles multiple outlier types through multi-class probabilistic outputs.

For time series data, hybrid approaches combining statistical preprocessing with deep architectures [71] demonstrate that feature selection maintaining interpretability can enhance deep models including LSTM-autoencoders, GANs, and transformers. Ensembling multiple deep architectures captures different aspects of temporal anomalies, with reconstruction-based and forecasting-based paradigms converging on similar effectiveness when properly combined with domain-informed preprocessing.

6) Reconstruction-Based AD: Reconstruction-based anomaly detection methods learn models capable of reconstructing normal data patterns and identify anomalies as samples with large reconstruction errors. The underlying hypothesis is that models trained on normal data cannot accurately reproduce unseen or irregular patterns, causing anomalies to exhibit systematically higher reconstruction loss. This principle applies across representation frameworks from linear subspace methods to sparse coding and deep autoencoders, with deep methods enabling non-linear mappings through neural architectures and introducing adversarial training or probabilistic latent spaces to improve reconstruction quality and anomaly discrimination.

Shallow Reconstruction-Based Approaches: Shallow reconstruction methods primarily build on Principal Component Analysis (PCA) and sparse representation frameworks, with reconstruction error measuring deviation from learned normal structure.

PCA-based approaches project data onto principal subspaces capturing dominant variance directions. Classical linear PCA computes reconstruction error in the original feature space but fails to capture complex non-linear manifolds. Kernel PCA [72] addresses this by performing PCA in an implicit infinite-dimensional feature space via the kernel trick, where reconstruction error becomes the squared distance to the principal subspace in feature space. For RBF kernels, decision boundaries follow principal subspace geometry with heterogeneous variance modeling, providing tighter enclosure than

uniform boundaries from one-class SVM or SVDD. However, both linear and kernel PCA face scalability challenges on large or streaming datasets where eigenvalue decomposition and covariance matrix storage become prohibitive. Online variants [73] address this through incremental updating without explicit eigen-decomposition, using an oversampling strategy that duplicates target instances to amplify outlier effects on principal directions—necessary because single anomalies produce negligible eigenvector variation in large datasets.

Sparse representation methods offer an alternative framework where normal data admit sparse decomposition over dictionaries. Rather than measuring reconstruction magnitude, joint sparse coding and anomaly detection [74] explicitly models outliers as structured deviations through a unified optimization that separates sparse normal components from a column-sparse outlier matrix. Non-zero columns directly indicate anomalous samples, treating anomalies as signals unexplainable by the sparse dictionary model.

Deep Reconstruction-Based Approaches: Deep reconstruction methods leverage neural network architectures to learn non-linear mappings between input and latent spaces, enabling modeling of complex data distributions beyond the capacity of linear methods. These approaches can be organized into two families: autoencoder-based methods that focus on deterministic or probabilistic encoding-decoding pipelines, and adversarial methods that incorporate discriminator networks to improve reconstruction quality or anomaly discrimination.

Autoencoder and Variational Methods.

Standard autoencoders learn compressed representations through encoder-decoder architectures, with reconstruction error serving as the anomaly score. However, the choice of reconstruction metric significantly impacts detection performance. While most methods use Mean Squared Error treating dimensions independently, Mahalanobis distance [75] incorporates covariance structure between variables, providing scale-invariant measurement particularly suitable for correlated multivariate data where sensor measurements exhibit strong interdependencies.

Variational Autoencoders (VAE) extend deterministic autoencoders with probabilistic latent spaces, learning distributions rather than point encodings. For time series data, temporal structure in both observation and latent spaces becomes critical. OmniAnomaly [76] addresses this by combining VAEs with GRUs and introducing stochastic variable connections that establish temporal dependence among latent variables across time steps, enabling the model to capture patterns missed by methods treating latent representations independently. Planar normalizing flows further enable learning non-Gaussian posterior distributions, while Peaks-Over-Threshold from Extreme Value Theory provides principled automatic threshold selection.

Architectural innovations for multivariate time series include explicit modeling of inter-variable dependencies alongside temporal patterns. GTAD [77] combines Temporal Convolutional Networks for capturing long-range temporal dependencies with Graph Attention Networks that learn input features correlation structures without requiring domain expertise. The hybrid prediction-reconstruction objective and

per-dimension loss optimization enable fine-grained anomaly localization while the parallel architecture provides computational efficiency over recurrent alternatives.

Adversarial Methods.

Adversarial training addresses limitations of standard autoencoders, which minimize reconstruction error universally and may fail to distinguish subtle anomalies from normal data. USAD [78] employs two autoencoders sharing an encoder, where the second autoencoder learns to distinguish real data from reconstructions of the first through adversarial objectives. This dual-path architecture amplifies reconstruction errors for anomalous patterns while maintaining training stability absent in pure GANs, with configurable sensitivity through weighted combination of both reconstruction paths at inference.

Dual-discriminator architectures provide complementary regularization at multiple points. DAEMON [79] uses separate discriminators for latent space regularization (guiding posterior distributions toward the prior) and reconstruction space regularization (preventing overfitting by ensuring reconstructed distributions match real data statistics). This addresses both generalization failures from irregular latent encodings and overfitting to training-specific patterns, while per-dimension reconstruction errors enable root cause analysis.

GAN-based methods offer alternatives where generators learn normal data distributions and anomalies are identified through various comparison mechanisms. AnoGAN principles adapted for tabular data [80] use per-sample latent vector optimization to find representations best reconstructing each test instance, with reconstruction error indicating distributional deviation. Rather than relying solely on reconstruction, GASN [81] exploits the implicit bias of GANs toward majority distributions when trained on contaminated data. Since normal samples dominate statistically, generated synthetic data serves as a normal reference benchmark, with anomalies identified through neighborhood analysis comparing original samples to synthetic neighbors via Euclidean distance.

7) *Clustering-Based AD*: Clustering-based anomaly detection operates on the principle that normal data points naturally form cohesive groups, while anomalies appear as isolated instances or points weakly associated with any cluster. Anomalies are identified either as noise points that cannot be assigned to clusters under density or distance constraints, or as points exhibiting significant deviation from cluster centroids. This approach provides unsupervised structure discovery without requiring labeled anomalies and naturally handles multimodal distributions through multiple clusters, though performance depends critically on the appropriateness of the clustering algorithm for the data distribution and careful parameter selection.

Shallow Clustering-Based Approaches: Shallow clustering methods apply classical algorithms directly to the feature space, identifying anomalies through distance to centroids, membership in sparse clusters, or classification as noise points. A fundamental challenge across these methods is parameter sensitivity, determining cluster counts, density thresholds, and neighborhood sizes typically requires domain expertise or expensive tuning.

Density-based approaches address this challenge through various parameter automation strategies. Both [82] and [83] enhance classical density-based clustering by incorporating domain-specific structure rather than relying on manual parameter tuning. The former introduces automated parameter selection for OPTICS through geometric covering space models that identify optimal neighborhood sizes based on pairwise distance distributions, while the latter adapts DBSCAN to seasonal time series by transforming temporal data into circular coordinate spaces that capture periodicity. This geometric feature engineering enables detection of local anomalies within seasonal contexts, deviations that appear normal globally but are anomalous relative to their temporal position. Taking a different approach, [84] replaces density thresholds entirely with statistical modeling, using multivariate normal distributions and Mahalanobis distance for membership determination in streaming data. The method performs single-pass clustering and outlier detection simultaneously, constructing virtual statistical distributions around outliers to enable consistent distance calculations and tracking population evolution through hierarchical sub-clustering.

Hybrid methods combine clustering with supervised classification to leverage complementary strengths. [85] and [86] both employ cascaded architectures where clustering pre-processes data for subsequent supervised learning. The earlier work uses k-Medoids with SVM in a two-stage framework, exploiting k-Medoids' robustness to outliers, medoid-based representatives are less biased by anomalies than mean-based centroids, critical when anomalies are the detection target. The recent work extends this to a three-stage cascade: K-Means for efficient initial partitioning, DBSCAN for density refinement, and SVM for final classification. This progression reflects recognition that no single algorithm excels across all data distributions, with each stage compensating for limitations of previous components. However, multi-stage pipelines increase complexity and propagate errors between stages.

[87] addresses multivariate time series through weighted Fuzzy C-Means, learning variable importance weights via Particle Swarm Optimization to balance contributions from time series components with different scales and relevance. The method unifies amplitude and shape anomaly detection by applying clustering to both original subsequences and their autocorrelation representations, the latter removing temporal shifts to enable pattern-based comparison.

These approaches demonstrate enhancement strategies for classical algorithms: geometric feature engineering (circular coordinates, covering space models), statistical modeling (Mahalanobis distance, virtual distributions), architectural composition (cascaded clustering-classification), and adaptive weighting (PSO-optimized distance metrics). While computationally efficient and interpretable, shallow methods operate on fixed feature representations and struggle with high-dimensional data or complex patterns without appropriate preprocessing.

Deep Clustering-Based Approaches: Deep clustering methods jointly learn feature representations and cluster structure through neural architectures, operating in learned latent spaces rather than hand-crafted features. This paradigm en-

ables automatic discovery of representations optimized for clustering separability while addressing high-dimensional and complex temporal data where shallow methods struggle.

A central question in deep clustering design is how to combine representation learning with clustering objectives. [88] demonstrates that simple architectural constraints can outperform complex joint optimization, showing that L2 normalization of autoencoder representations, constraining encoded vectors to the unit hypersphere, yields superior clustering with standard k-means compared to sophisticated methods like DEC and IDEC. The geometric insight is that normalization makes Euclidean distances equivalent to angular distances, inherently more suitable for clustering algorithms. For anomaly detection, this approach dramatically outperforms reconstruction error-based methods by measuring deviation from cluster structure rather than reconstruction quality, revealing a critical flaw: reconstruction-based detection fails when anomalies are structurally simple despite being semantically anomalous.

Recognizing that single representations cannot optimally serve multiple objectives, [89] splits the autoencoder latent space into specialized subsets: clustering-friendly variables optimized for discrimination and reconstruction-friendly variables for input fidelity. A clustering loss applies only to the top-K most discriminative features, selected dynamically during training based on inter-cluster distances. This meta-learning principle, the network learns which representational components specialize in which tasks, distinguishes the approach from static feature allocation. Validation on industrial time series demonstrates substantial improvements over standard autoencoders and shallow methods.

Beyond pure accuracy, [90] addresses algorithmic fairness by integrating adversarial training with deep clustering. The framework jointly optimizes reconstruction, clustering, and fairness through a dynamic weighting scheme that de-emphasizes outliers during training while using adversarial losses to hide sensitive attribute information in learned representations. This demonstrates that deep clustering's representational flexibility enables multi-objective optimization, accuracy, fairness, discrimination, simultaneously, with fairness constraints potentially improving detection by forcing more robust features.

[91] eliminates the requirement to specify cluster count by replacing k-means with BIRCH clustering, which dynamically determines cluster assignments through hierarchical tree structures. The method combines hyperparameter-optimized autoencoders with BIRCH's single-scan processing capability, suitable for streaming network traffic where cluster counts evolve over time. Experimental results show that this component-based approach, carefully optimized autoencoder plus adaptive clustering algorithm, can outperform integrated joint optimization methods.

The evolution of deep clustering reflects three design philosophies: architectural simplicity through normalization constraints versus specialized objectives through latent space splitting, integrated joint optimization versus optimized component composition, and pure accuracy versus multi-objective optimization including fairness and automatic adaptation. The field has progressed from establishing feasibility to addressing

practical deployment requirements, streaming data, unknown cluster counts, fairness constraints, with methods representing different trade-offs between performance, interpretability, and computational cost.

8) *One-Class-Based AD*: One-class methods explicitly learn decision boundaries or manifolds that enclose normal data. Observations outside this region are treated as anomalies. While many anomaly detection approaches train exclusively on normal samples, this category is distinguished by its direct optimization of boundary compactness or support estimation. This contrasts with reconstruction-based methods that measure deviation through reconstruction error, or distance-based methods that rely on proximity metrics. The explicit boundary formulation enables principled threshold determination and theoretical interpretability. However, it introduces challenges in preventing degenerate solutions where boundaries collapse to trivial representations.

The one-class paradigm was established through Support Vector Data Description and OCSVM in the early 2000s. These classical methods remain relevant for contextualizing modern deep learning extensions. Contemporary research addresses three main challenges: representation collapse in neural architectures, contamination robustness for real-world data, and modality-specific adaptations for temporal and tabular domains. The progression from foundational collapse-prevention (2020) through contamination robustness (2024) to decomposition and disentanglement paradigms (2025) reflects maturing research addressing practical deployment challenges.

Shallow One-Class-Based Approaches: Classical kernel-based methods established theoretical foundations through OCSVM extensions that address two key limitations: adapting to sequential data and handling contaminated training sets.

For temporal data, phase space embeddings [92] bridge one-class SVMs to time series by converting sequential data into vector representations via time-delay windows. The core innovation addresses low-frequency signal bias through projected phase space. This orthogonal projection implements adaptive high-pass filtering without manual tuning. Multi-scale detection across varying embedding dimensions ensures robustness by requiring consistent anomalousness. This exemplifies shallow learning characteristic approach: explicit feature engineering precedes established algorithms.

For contaminated training, adaptive sample weighting [93] combines local density with global positioning to identify suspicious samples. Standard OCSVM equally penalizes all training samples, allowing outliers to distort boundaries. The adaptive approach downweights suspicious samples through combined local-global criteria, preventing contamination from corrupting the decision boundary. Experiments demonstrate significant improvements, validating that intelligent sample weighting enhances boundary robustness without requiring clean training data.

Deep One-Class-Based Approaches: Deep one-class learning confronts representation collapse as its fundamental challenge. This occurs when networks learn trivial mappings that satisfy objectives without capturing meaningful structure. Solutions diverge into three areas: preventing collapse through

adversarial training or architectural design, handling contaminated training data, and adapting to specific modalities.

Preventing Representation Collapse. Early deep one-class methods, particularly DeepSVDD, suffered from collapse where all points map to identical representations. This trivially satisfies hypersphere objectives but captures no meaningful patterns. Two complementary strategies emerged to address this challenge: adversarial training and hierarchical architectural design.

The adversarial approach [94] leverages the manifold hypothesis: normal data resides on low-dimensional manifolds, enabling synthetic anomaly generation through off-manifold perturbations. Training alternates between creating hard negatives via gradient ascent around normal points and optimizing classification between originals and perturbations. Collapse becomes impossible because identical mappings eliminate discriminability. This manifold-based strategy generalizes across modalities without requiring domain-specific augmentations, unlike transformation-based self-supervision. The DROCC-LF variant learns Mahalanobis distances when limited negatives are available.

In contrast, the hierarchical approach [95] prevents collapse through architectural constraints rather than adversarial pressure. THOC introduces multi-hypersphere structures for time series, replacing DeepSVDD single global hypersphere with hierarchical layers that capture multi-scale temporal dynamics. Dilated RNNs with exponentially increasing skip connections enable lower layers to model short-term patterns while upper layers capture long-term dependencies. Each layer learns multiple hypersphere centers organized via differentiable soft clustering. Orthogonal regularization prevents center redundancy while temporal self-supervision enhances representation learning. Ablations validate that both hierarchical structure and multiple hyperspheres per layer are essential, removing either substantially degrades performance. These complementary approaches establish that collapse prevention requires either discriminative pressure or explicit structural constraints.

Contamination-Robust Time-Series Methods. Real-world training data frequently contains unlabeled anomalies, violating clean-data assumptions. Recent work addresses this through three distinct philosophies: soft uncertainty-based masking, hard input decomposition, and learned feature masking with pseudo-labeling.

Uncertainty modeling [96] treats contamination as a confidence problem. COUTA produces dual distance predictions whose disagreement signals potential contamination. High uncertainty triggers exponential masking coefficient reduction while penalizing uncertainty itself to ensure confident predictions on legitimate normal samples. This soft masking preserves informative boundary samples that hard filtering would discard. Domain-specific synthetic anomalies, point, contextual, and collective perturbations, provide anomaly-aware guidance through an auxiliary classification head. The dual calibration (uncertainty modeling and synthetic anomalies) contributes synergistically.

Hard decomposition [97] takes a fundamentally different approach inspired by Robust Principal Component Analysis. Rather than weighting samples, ROC explicitly separates

inputs into low-rank clean structure and sparse anomalous components. Only clean data feeds the autoencoder while sparsity constraints apply to anomalous parts. Neural networks enable nonlinear temporal decomposition through alternating optimization, extending linear Robust PCA to complex temporal patterns. This hard separation proves particularly effective when contamination exhibits structured sparsity. The method also addresses single-hypothesis limitations by combining reconstruction and one-class objectives.

Learned masking with pseudo-labeling [98] addresses both contamination and the identical shortcut problem simultaneously. AMFormer network-driven masking breaks identity shortcuts by forcing reconstruction from partial features, while active MSE loss assigns exponentially decreasing weights based on reconstruction errors. Unlike COUTA uncertainty-based soft masking, this exponential pseudo-label weighting provides more aggressive downweighting. Unlike ROC hard decomposition, the soft re-weighting preserves boundary sample information. The unified single-stage optimization jointly learns which features to mask and how to weight samples, avoiding the misaligned objectives of two-stage methods.

These approaches represent a spectrum from soft to hard contamination handling, each suited to different contamination characteristics.

Tabular-Specific Adaptations. Tabular data presents unique challenges: unknown structure, mixed types, heterogeneous features, low correlation, and sensitivity to single-feature changes. Image-focused methods fail to address these characteristics. Two recent approaches demonstrate that specialized architectures are essential, though they pursue fundamentally different design philosophies.

The GAN-based approach [99] adapts generative adversarial training for tabular heterogeneity through dual encoders and 1D-convolutional layers. Where image GANs use 2D convolution, 1D convolution captures tabular feature correlations and single-feature sensitivity. The dual encoder architecture addresses the non-intersecting manifold problem during training: the real encoder learns the generator inverse for detection while the fake encoder co-trains with the generator for learning guidance. This separation differs from single encoder approaches or integrated dual encoders by providing separated training objectives that prevent mode collapse. Ablations validate that 1D convolution contributes feature correlation modeling while dual encoders provide training stability.

The disentanglement approach [100] addresses a fundamental limitation of prior masking methods where learnable masks produce uniform soft values that essentially mask nothing, leading to identity mappings. Disent-AD employs two-head self-attention to extract non-overlapping attribute subsets with orthogonality constraints ensuring genuine separation. Successful reconstruction from either subset independently validates learned correlation rather than memorization. Ablations confirm that removing disentangling loss causes dramatic drops, the model fails to extract non-overlapping subsets without orthogonality constraints. Analysis reveals two subsets are optimal; more subsets become harder to train and may exceed actual correlation structure.

Both approaches validate that tabular characteristics demand

specialized architectures rather than adapted image methods. The GAN-based strategy focuses on robust training through dual encoders, while the disentanglement strategy focuses on explicit correlation modeling through subset decomposition.

9) *Forecasting-Based AD*: Forecasting-based anomaly detection operates on the principle that normal behavior exhibits predictable temporal patterns. These methods model expected future values and identify anomalies as significant deviations between predictions and observations. The paradigm assumes that models trained on normal data cannot accurately forecast anomalous events, manifesting as elevated prediction errors or violations of learned uncertainty bounds.

Shallow Forecasting-Based Approaches: Classical statistical time series methods form the foundation of forecasting-based anomaly detection. [101] demonstrates this approach through ARIMA and a modified Holt-Winters method applied to network traffic monitoring. Both techniques construct confidence bands around forecasts, flagging observations that simultaneously violate thresholds on multiple metrics as anomalies. The ARIMA model captures linear dependencies through autoregressive and moving average components, while the modified Holt-Winters method explicitly models trend and seasonality through exponential smoothing. A critical design choice emerges in confidence band construction: symmetric intervals based on variance-to-mean ratios versus asymmetric intervals adapted to error distribution characteristics, with the latter proving more effective at reducing false positives.

The 2001 work by [102], while focused on outlier detection within ARIMA model fitting rather than operational anomaly detection, provides important historical context. This earlier research addressed a fundamental challenge: classical maximum likelihood estimation fails when training data contains outliers, creating a bootstrapping problem where contaminated data yields biased models. The solution of robust parameter estimation combined with iterative detection and cleaning influenced subsequent developments in making forecasting methods resilient to noisy real-world conditions.

Modern shallow methods have evolved toward online operation and novel anomaly indicators. [103] introduces online gradient descent for continuous ARIMA parameter adaptation, enabling true streaming detection without batch retraining. Beyond computational efficiency, this approach proposes monitoring discrete differences in model weight coefficients rather than prediction residuals. The rationale is that sudden shifts in optimal parameters may signal system state changes more robustly than residual spikes, though this introduces heuristic complexity through multiple aggregation metrics and threshold selection strategies.

Deep Forecasting-Based Approaches: The evolution from statistical to neural forecasting methods began with recurrent architectures capable of learning long-term dependencies. [104] established stacked LSTM networks as the foundation for this paradigm. By modeling prediction errors as multivariate Gaussian distributions estimated from normal training data, the method provides anomaly likelihoods rather than binary classifications. A key insight is multi-step prediction: forecasting several timesteps ahead ensures the network cap-

tures meaningful temporal structure, generating richer error signatures than single-step prediction.

Alternative neural architectures soon emerged. DeepAnT [105] demonstrated that convolutional networks can effectively replace recurrent models for time series forecasting, offering dramatic computational advantages through parallel processing while maintaining competitive accuracy. Meanwhile, HTM [106] explored neuromorphic principles, using sparse distributed representations and continuous online learning to handle concept drift. A critical innovation in HTM is the two-layer detection framework: the sequence memory predicts future representations, then an anomaly likelihood layer models the distribution of prediction errors to reduce false positives in noisy streams. This separation of prediction from anomaly scoring, rather than direct residual thresholding, proves essential for robust performance.

The shift to multivariate time series required explicitly modeling inter-sensor relationships. Graph Deviation Network [107] addresses this through learned graph structures where attention mechanisms weight the importance of different sensor dependencies for forecasting. Each sensor receives an embedding that captures its behavioral characteristics, and the graph structure itself emerges from similarity in embedding space. Anomalies manifest as deviations from graph-constrained predictions, with attention weights providing interpretability about which relationships were violated. A related idea is applied to in-vehicle CAN traffic in [108], where correlated signals are forecast jointly and an anomaly is raised when their combined prediction error exceeds a threshold.

Transformer architectures brought significant advances through their ability to process entire sequences in parallel and model long-range dependencies. TranAD [109] combines transformers with adversarial training and self-conditioning: reconstruction errors from an initial inference phase become focus scores that guide attention in a second phase, amplifying subtle anomalies. Two decoders engage in an evolutionary adversarial game where one minimizes reconstruction error while the other maximizes it, with training weights shifting from reconstruction-focused to adversarial-focused over epochs. The GAT-Informer hybrid [110] extends this by combining graph attention for sensor relationships with Informer architecture for efficient long-sequence processing, using both short-term and long-term forecasting to validate predictions against future trends.

A distinct line of research focuses on uncertainty quantification rather than point predictions. [111] constructs prediction intervals using Monte Carlo dropout and quantile bootstrap methods across multiple architectures including cascaded networks, echo state networks, and LSTMs. Anomalies are identified when observations fall outside learned uncertainty bounds, with dynamic thresholding based on Chebyshev's inequality adapting to data characteristics. This approach makes explicit what remains implicit in residual-based methods: the confidence in predictions varies across the time series, and anomaly detection should account for this uncertainty.

The comprehensive benchmark study by [71] systematically compares seven deep learning paradigms on the same problem, revealing that hybrid architectures combining LSTM

temporal modeling with autoencoder reconstruction principles significantly outperform either approach alone. The study demonstrates that pure reconstruction methods without temporal modeling produce excessive false positives, while classification-based approaches require labeled data. Ensemble methods and data augmentation via GANs prove particularly effective for handling the severe class imbalance inherent to anomaly detection.

Several design principles emerge across these methods. The shift from univariate to multivariate modeling requires explicit mechanisms for capturing inter-dimensional dependencies, whether through graph structures, attention mechanisms, or joint reconstruction. Finally, interpretability increasingly comes not just from detecting that an anomaly occurred, but from explaining which features or relationships caused the deviation, a capability that graph-based and attention-based methods provide through their learned structures.

10) Summary of AD Methods: Table IV provides a comprehensive summary of the AD methods discussed in this section, organized by category and approach, along with their key references.

11) XAI in AD: In this section, we explore the application of XAI approaches in AD. In many real-world industrial scenarios, explicit fault or anomaly labels are scarce, unreliable, or entirely unavailable, making unsupervised learning a natural and often unavoidable choice. While this paradigm enables the identification of deviations from normal operating conditions without prior knowledge of failure modes, it further increases the interpretability challenges of data-driven models. In particular, it is often unclear why a specific observation is flagged as anomalous, which process variables, signal features, or temporal dynamics contribute most to the detected deviation, or whether the anomaly reflects a meaningful physical phenomenon rather than noise or data artifacts. This lack of transparency limits trust, hampers root-cause analysis, and complicates the deployment of AD systems in safety-critical industrial environments. In this context, XAI techniques become essential to uncover the internal reasoning of AD models, providing both global insights into learned normality patterns and local explanations of individual anomalies, in support of reliable, human-centered monitoring solutions aligned with the Industry 5.0 vision. As will be discussed in the following section, the majority of application-oriented studies on AD in the industrial domain rely on a limited set of deep learning architectures that have demonstrated high detection performance. However, despite their effectiveness, deep anomaly detection models are often criticized for their limited interpretability. For this reason, the majority of XAI methods adopted in this context are post-hoc, with several works proposing their use to improve the transparency of the underlying Neural Networks (NNs) [112]. Importantly, the availability and suitability of XAI methods strongly depend on the specific model architecture employed. In particular, Recurrent Neural Network (RNN) such as Long Short Term Memory Network (LSTM) and Gated Recurrent Unit (GRU), together with reconstruction- and generation-based approaches including Auto Encoders (AE), Variational Auto Encoder (VAE), and Generative Adversarial Network (GAN), are among the

most commonly adopted solutions. Consequently, the following discussion focuses on XAI methods that are applicable to these model architectures, with the aim of demonstrating how interpretability can be enabled in real-world, industrial AD scenarios. AEs are among the most widely adopted architectures for AD, based on the assumption that anomalous samples are more difficult to reconstruct than normal ones. However, their encoder-decoder structure inherently limits interpretability, as anomalies are typically identified through aggregate reconstruction errors. To address this issue, most explainability approaches for AE-based models rely on post-hoc XAI techniques. Common strategies include feature attribution methods, such as Shapley Additive Values (SHAP) [113] and gradient-based explanations, which quantify the contribution of individual input variables to the reconstruction error, as well as model-specific analyses inspecting feature-wise or temporal reconstruction residuals. In addition, visualization techniques, particularly in image and video applications, are often employed to localize anomalous regions by highlighting discrepancies between input and reconstructed data. Despite these efforts, explanations for AE-based AD models are generally local and instance-specific, providing limited insight into the global behavior of the learned normality model [112], as observed in industrial applications such as the AE-based approach proposed by Jang et al. [114]. RNNs, in particular LSTM and GRU architectures, are widely used for AD in sequential and time-series data due to their ability to model temporal dependencies and predict normal system dynamics. However, the temporal recurrence and internal gating mechanisms of RNNs make their decision process difficult to interpret. As a result, explainability for RNN-based AD models is predominantly addressed through post-hoc XAI techniques. Common approaches include feature attribution methods such as SHAP [113] and gradient-based explanations to identify the most influential variables or time steps contributing to an anomalous prediction, as well as surrogate models like Local Interpretable Model-Agnostic Explanations (LIME) [115] to approximate local model behavior. Additionally, attention mechanisms and relevance propagation techniques are sometimes exploited to provide insights into temporal importance patterns. Despite these efforts, explanations for RNN-based AD models are typically local and instance-specific, often computationally demanding, and strongly dependent on the specific RNN architecture and input sequence length, offering limited insight into the global temporal behavior learned by the model, as observed in industrial applications such as the LSTM-based approach proposed by Nor et al. [116].

B. AD Industrial Applications

This is the section where several industrial case studies are considered to showcase how the methodologies presented in II-A2 are applied in real-world scenarios.

Unlike II-A2, where the focus was on the methodological aspects, in this section we mainly treat case studies highlighting the specific choices that are made to ensure a successful integration of such AI-based technologies inside complex industrial scenarios.

TABLE IV
SUMMARY OF AD METHODS REVIEWED IN SECTION II-A, ORGANIZED BY CATEGORY, LEARNING PARADIGM (SHALLOW/DEEP), AND APPROACH.

Category	Type	Approach	Key Methods	References
Density-Based	Shallow	Local density estimation Histogram-based Copula and mixture models	LOF, LOCI, LDF HBOS, Loda COPOD, GMM	[32], [33], [34], [35], [36], [37] [38], [39], [40] [41], [42]
	Deep	Normalizing flows Hybrid AE-density Diffusion-based	Conditional flows, likelihood ratio DAGMM TabADM	[43], [44], [45], [46] [47] [48]
Distance-Based	Shallow	Nearest neighbor Mass-based dissimilarity Angle-based Robust subspace	k -NN, adaptive neighborhood, DTW variants Probability mass distance ABOD CoMadOut	[49], [50], [51] [52] [53] [54]
	Deep	Learned representations	BiGAN-Mahalanobis, CTF, contrastive learning	[55], [56], [57]
Ensemble-Based	Shallow	Isolation-based Selective combination	Isolation Forest, Extended IF, Hypersphere IF LSCP, graph-based selection	[58], [59], [60], [61], [62], [63], [64], [65] [66], [67], [68]
	Deep	Deep isolation Hybrid ensembles	Deep Isolation Forest Stacked AE + PNN, LSTM-AE ensembles	[69] [70], [71]
Reconstruction-Based	Shallow	PCA-based Sparse representation	Linear PCA, Kernel PCA, Online PCA Joint sparse coding	[72], [73] [74]
	Deep	Autoencoder variants Adversarial methods GAN-based	AE, VAE, OmniAnomaly, GTAD USAD, DAEMON AnoGAN variants, GASN	[75], [76], [77] [78], [79] [80], [81]
Clustering-Based	Shallow	Density-based clustering Hybrid clustering-classification Fuzzy clustering	OPTICS, DBSCAN variants k-Medoids + SVM, K-Means + DBSCAN + SVM Weighted FCM with PSO	[82], [83], [84] [85], [86] [87]
	Deep	Deep clustering	Normalized AE + k-means, latent space splitting, fair clustering	[88], [89], [90], [91]
One-Class-Based	Shallow	Kernel-based	OCSVM, phase space embedding, adaptive weighting	[92], [93]
	Deep	Collapse prevention Contamination-robust Tabular-specific	DROCC, THOC COUTA, ROC, AMFormer Dual encoder GAN, Disent-AD	[94], [95] [96], [97], [98] [99], [100]
Forecasting-Based	Shallow	Statistical	ARIMA, Holt-Winters, Online ARIMA	[101], [102], [103]
	Deep	RNN-based CNN-based Graph-based Transformer-based Uncertainty quantification	Stacked LSTM, HTM DeepAnT Graph Deviation Network, GAT-Informer TranAD MC dropout, quantile bootstrap	[104], [106] [105] [107], [110] [109] [111], [71]

Before diving into specific approaches and applications, it is useful to provide an overview of the main industrial contexts in which AD, FD, and PdM techniques are deployed. Industrial systems typically involve multiple interacting layers — mechanical, electrical, cyber-physical, and human — each of which can be a potential source of faults or abnormal behaviors. Identifying and classifying such faults is therefore essential to understand how AI-based diagnostic and prognostic frameworks are tailored to different operational environments.

Tables V and VIII present a taxonomy of representative industrial fault categories and subcategories that form the conceptual foundation for this overview. The classification highlights the domains where data-driven and hybrid diagnostic systems are most relevant, including:

- Equipment / Mechanical faults, such as wear, imbalance, or component degradation;
- Process / Quality deviations, including process drifts, raw material inconsistencies, or quality defects;
- Sensor / Data anomalies, arising from calibration errors, data loss, or malicious signal manipulation;
- Control System / Cyber faults, including PLC malfunctions, control loop instabilities, and cybersecurity threats.

These fault categories capture the complex and interconnected nature of industrial operations, illustrating how anomalies and failures may emerge and propagate across physical and digital layers. Each group also corresponds to a set of challenges and research efforts in developing AI-driven models capable of ensuring robust, interpretable, and human-centric decision-making.

Building on this taxonomy, the following subsections provide a structured examination of the three key paradigms in

industrial diagnostics and prognostics — AD, FD, and PdM — highlighting their methodological principles, application domains, and relevance within the evolving landscape of Industry 4.0 and 5.0.

As summarized in Table V, AD techniques are utilized across multiple fault categories — from mechanical degradation and process deviations to sensor and cyber-physical anomalies — enabling early identification of abnormal patterns and improving overall system resilience.

1) Mechanical/equipment faults: Mechanical and equipment-related faults represent one of the most extensively studied application domains for AD, encompassing a wide range of phenomena such as rotating machinery degradation, tool wear, bearing defects, and electromechanical component failures. Within this broad category, rotating machinery has attracted significant attention due to the availability of rich vibration data and well-established condition monitoring practices, motivating the development of numerous AD approaches. For instance, [117] employs a hybrid bidirectional LSTM–GRU autoencoder trained on healthy vibration data; reconstruction error is used to identify abnormal machine behavior and predict maintenance needs. [118] also adopts an AE framework but directly processes raw vibration signals without manual feature extraction, reaching an F1-score of 99.6% on real-world datasets. [119] proposes a modular methodology combining AD with XAI tools (SHAP and Local-DIFFI), enabling both automated fault diagnosis and interpretability of model decisions. Recent studies on tool wear detection emphasize AD approaches to overcome the limitations of traditional FD, which often requires extensive labeling. [120] proposes two low-label or label-free strategies for predicting the end of life of drilling tools: a classification-based AD method using XGBoost, and an unsupervised AE model. Both approaches demonstrate high accuracy across different cutting parameters, with the XGBoost model providing an interpretable wear indicator and the autoencoder's reconstruction error effectively correlating with tool degradation. [121] presents an unsupervised deep learning–K-means hybrid method for tool health monitoring, where vibration signals undergo unsupervised pre-training to automatically extract relevant fault features, exploiting an AE model. This integration allows adaptive feature learning and accurate clustering of tool conditions in large-scale, multi-tool datasets, eliminating the dependence on expert knowledge or manual feature design. AD methods have shown strong potential for identifying faults in induction motors and electrical drives, where abnormal data are often scarce. [122] introduces a GAN-based model (GANomaly) that uses audio signals transformed into mel-spectrograms from multiple microphones to detect bearing faults in induction motors. The method achieves high performance, with an AUROC of 0.9822 and 92.52% accuracy, effectively distinguishing between normal and faulty conditions without prior exposure to fault data. [123] proposes a VAE-based approach for anomaly detection in electric drives, relying on reconstruction error as the fault indicator. Trained and tested on data from a 200 W servo-motor testbed, the model demonstrates reliable detection performance and is successfully implemented on a

DSP platform for real-time evaluation, confirming its practical applicability to industrial systems.

2) Process/Quality faults: Within the industrial domain, AD methods have also been applied to monitor process stability and product quality, addressing both process drift and manufacturing defects.

For process drift and anomalies, [124] presents an AD framework for complex industrial machinery such as gas turbines. The proposed method combines AE-based architectures with sliding windows to detect early signs of abnormal behavior directly from operational data, without requiring previously recorded fault instances. Validated on real industrial environments, the approach provides real-time monitoring capabilities and generates valuable information for predictive maintenance. Its adaptable design allows application to other variable-duration industrial processes where labeled data are limited or unavailable.

Regarding product quality defects, [125] investigates vacuum testing in refrigerator production, introducing a Bayesian Neural Network (BNN) for DL-based AD on time-series pressure profiles. The model produces an anomaly score for each unit, supporting operators in inspection and quality control decisions. By explicitly accounting for model uncertainty, the BNN enhances interpretability and reliability, reducing dependence on manual evaluation. Integrated within a decision support system, this approach mitigates human bias, optimizes inspection efforts, and minimizes production costs associated with false positives and false negatives.

3) Sensor/Data faults: In modern industrial environments, the integration of cyber-physical systems (CPS) and Industrial Internet of Things (IIoT) infrastructures has increased exposure to security threats such as false data injection (FDI) and other signal manipulation attacks. AD methods have been proposed to address these challenges by learning normal system behavior directly from data, without requiring labeled attack instances.

[126] introduces an unsupervised RNN-based approach for detecting cyber-attacks in CPS, using a time-series prediction model combined with the Cumulative Sum (CUSUM) technique to identify anomalies. Applied to the Secure Water Treatment Testbed (SWaT) dataset, the method effectively detects a wide range of attack scenarios while maintaining a low false positive rate, and can also pinpoint the specific sensor under attack, enhancing interpretability and response accuracy.

[127] focuses on false data injection detection within IIoT systems, leveraging AEs to exploit both temporal and spatial correlations among sensor data. The approach not only detects falsified measurements but also employs Denoising Autoencoders (DAEs) to recover clean data from compromised signals. Experimental results demonstrate that this method outperforms Support Vector Machine (SVM)-based techniques, offering a more robust and self-correcting mechanism for safeguarding industrial sensor networks.

4) Control/Cyber systems faults: AD techniques are increasingly employed in control and CPS to ensure operational reliability, real-time responsiveness, and resilience against both process-related faults and cyber intrusions.

TABLE V
TAXONOMY OF INDUSTRIAL FAULT TYPES, REPRESENTATIVE SUBCATEGORIES, AND CORRESPONDING USED METHODS.

Fault Type	Category	References
Equipment / Mechanical	Rotating machinery faults (bearings, shafts, gears, imbalance)	[117], [118], [119]
	Tool wear and breakage (cutting tools, milling, drilling)	[120], [121]
	Induction motor / electrical drive faults (stator winding, rotor bars)	[122], [123]
Process / Quality	Process drift and anomalies (chemical, thermal, flow processes)	[124]
	Product quality defects (surface cracks, mis-shapes, etc.)	[125]
Sensor / Data	Cyber-physical signal injection (spoofed/manipulated IoT data)	[126], [127]
Control / Cyber systems	ICS faults (PLC malfunction, loop instability)	[128], [129], [130]
	Cybersecurity faults (intrusion, denial-of-service, adversarial manipulation)	[131]

For Industrial Control System (ICS) faults, [128] demonstrates the use of a composite AE model trained on HIL-augmented ICS data to identify anomalous behavior in ICS without relying on labeled datasets. The approach effectively captures both temporal and value-based deviations, enabling robust real-time monitoring and enhanced process reliability. [129] investigates AD in ICS communication networks under simulated lateral cyberattacks, comparing several machine learning models such as One Dimensional Convolutional Neural Networks (1DCNNs), LSTMs, SVMs, and Isolation Forests (IFs). The 1DCNN achieved the best trade-off between accuracy (0.92) and processing time, making it well-suited for real-time intrusion detection in critical infrastructure. [130] introduces a decentralized and real-time AD framework for cyber-physical production systems, employing multiple 1D Convolutional Autoencoders (1DCAEs) in a sliding-window configuration. By decentralizing model execution across individual subsystems, the approach reduces communication overhead and eliminates the need for expert configuration, achieving reliable detection of local process anomalies in real industrial settings.

Regarding Cybersecurity faults, [131] explores unsupervised clustering-based AD applied to heterogeneous cybersecurity data streams. The study evaluates several clustering algorithms using metrics such as WSSSE and Silhouette scores, identifying K-Means as the most effective method for detecting anomalies in real-time streaming data. The proposed framework demonstrates high detection accuracy while maintaining computational efficiency, supporting timely decision-making in dynamic cybersecurity environments.

III. FAULT DETECTION AND DIAGNOSIS

A. FDD Methods and Theoretical Foundations

In this section the theoretical foundations and the state of the art methodological approaches for FDD will be presented.

1) **FDD Definitions:** FDD is a key function in industrial automation, aimed at identifying malfunctions in monitored systems. As part of condition monitoring, it enables timely recognition of faults, helping to maintain reliability, safety and efficiency.

FDD is usually divided into two main tasks:

- 1) **Fault Detection:** deciding whether the system is operating normally or if a fault is present, producing a binary outcome.
- 2) **Fault Diagnosis:** determining the specific type, cause and/or location of the fault, namely fault classification.

Methods for FDD are commonly grouped into three categories: *knowledge-based*, *model-based*, and *data-driven*. Knowledge-based methods rely on expert knowledge, heuristics, or rule-based reasoning to identify faults.

Model-based approaches, on the other hand, employ mathematical or physical representations of a system nominal behavior. Faults are detected by comparing outputs predicted with such representations against measured data. Abnormal conditions are identified when significant deviations are obtained in the latter comparison.

Both categories perform well when the process is thoroughly understood and accurate models or expert rules can be established. In many complex industrial systems, however, developing such detailed knowledge or precise models is impractical or prohibitively costly, which limits their applicability in such cases. Data-driven approaches, by contrast, overcome these limitations by extracting patterns that characterize normal and faulty system behaviors by large volumes of data [132].

Within data-driven techniques, statistical-based approaches, such as Statistical Process Control (SPC) and Principal Component Analysis (PCA), are widely used for FDD [133]. These methods detect deviations from normal operations by analyzing the underlying data structure, for example through hypothesis testing or dimensionality reduction. Since they do not learn predictive mappings, they are not considered ML methods and are therefore not reviewed here.

In the context of machine learning, FDD is typically formulated as a *supervised learning* problem. Labeled examples of both normal operation and known faults are used to train models, which can then detect and/or classify incoming data according to these predefined categories. This limits FDD to faults seen during training, so novel faults cannot be reliably classified. In contrast, AD is typically unsupervised, detecting any deviation from a learned normal behavior. While this distinction is not universally shared in the literature, we adopt it in this work, forming the basis for the following discussion. Finally, FDD differs from PdM in terms of the task objective.

FDD is diagnostic, assessing the current state of the system, whereas PdM is prognostic, aiming to predict future system degradation and failures (see Section IV-B).

In ML-based FDD, different types of data are collected and used. For machinery fault diagnosis, vibrational data are the most utilized [24], as they are informative for a wide range of machinery faults. Acoustic emissions are also employed, particularly for early fault detection in rotating components and induction motors [134], [135], as they are cheaper and less invasive than vibration-based analysis. Motor current signals are commonly used in Motor Current Signature Analysis (MCSA) for the diagnosis of electrical faults, such as broken rotor bars or shorted stator windings [136]. Other data, such as wear debris analysis, oil analysis or thermal imaging are also employed [24]. For discrete manufacturing machines, common types of logs are cycle time, throughput, utilization, or acceleration/deceleration rates [132]. In addition, for industrial processes, control variables are typically used, such as temperature, pressure, flow rates. For IIoT and CPSs faults may include not only mechanical or process ones, but also cyber-attacks (e.g., in SCADA systems), communication failures or firmware faults. In such context, data for FDD can include network traffic logs or protocol in addition to sensor and control data [137], [138].

2) FDD Methods:

Traditional ML: FDD using traditional machine learning relies heavily on feature engineering, a critical pre-process for the successful deployment of the models. Raw sensor measurements or system logs must be transformed into informative descriptors that capture the characteristic of normal and faulty operation.

The specific types of designed features depend on the domain and available measurements. For *rotating machinery and equipment* standard transformations such as Fast Fourier Transform (FFT), Wavelet Transform (WT), Wavelet Packet Transform (WPT), Empirical Mode Decomposition (EMD), Hilbert-Huang Transform (HHT), Variational Mode Decomposition (VMD), or Short Time Fourier Transform (STFT), are commonly used to expose frequency- and time-frequency patterns that indicate mechanical or electrical faults [24]. Features are typically grouped into:

- **Time-domain features:** statistical descriptors such as Root Mean Square (RMS), kurtosis, skewness, peak-to-peak value, and crest factors. It is also a common practice to compute cumulative versions of these features in order to obtain monotonically increasing variables to better represent the advancement of equipment degradation.
- **Frequency-domain features:** spectral peaks, fault-characteristic frequencies (e.g. Ball Pass Frequency Outer Race), band energy ratios or sideband patterns in induction motors (via MCSA, indicative of rotor or eccentricity faults) [139]. In particular, fault-characteristic frequencies are particularly useful in rotating machinery equipment (i.e. bearings, gears, ...) [140] where the frequency spectrum is characterized by the presence of harmonics associated with the rotational speed of the equipment.
- **Time-frequency features:** These kind of features are used in presence of non-stationary or highly oscillatory

signals, where time and frequency domain features are not able to provide HI capturing the true health state of the equipment. Typically used time-frequency domain features are: STFT, Wigner-Ville Distribution (WVD) and WT.

For *Discrete Manufacturing Machines* (i.e., machines that operate in distinct cycles to produce individual items [132]), additional features such as cycle time, throughput, utilization, and other discrete operational KPIs are often used [132], [141]. Discrete events such as actuator commands and sensor triggers can also serve as informative indicators of faulty behavior [142]. Despite their relevance FDD approaches for discrete manufacturing machine are relatively underexplored in the literature. For *IIoT-specific faults*, including attacks, intrusions or communication failures, features typically describe network communication patterns (e.g., traffic flows, protocol usage), enabling the detection of cyber-physical faults.

Once features are extracted, standard supervised models are trained. Widely used classifiers include Random Forest (RF) [143], [144], [145], [146], [147], gradient-boosted trees (e.g., XGBoost, LightGBM) [148], [145], [147], SVM [149], [150], [151], K-nearest neighbors (k-NN) [152], shallow NN [153], [150] and ensemble models [154]. These models have also been shown to perform effectively in multi-fault scenarios, where a single model must identify and classify multiple simultaneous faults, a more realistic and robust setting [155].

In addition to feature engineering, feature selection is important for improving model accuracy and generalization and mitigate overfitting, for both machinery faults and cyber-attacks. Common approaches include *recursive feature elimination* [148], filtering based on their importance (e.g., derived from RF or XGBoost) [156], correlation-based selection [157], [145], or more advanced optimization-based methods [158].

Hybrid architectures combining deep learning for automatic feature extraction with shallow classifiers for decision making have also been proposed. These approaches often outperform both purely traditional and purely deep models, especially when labeled datasets are limited or fault signatures are complex. For example, [159] transformed vibration signals into 2D Continuous Wavelet Transform (CWT) images, extracted multi-level features with a pretrained LeNet-5 Convolutional Neural Network (CNN), and classified them using an ensemble of RF. Similarly, [160] demonstrated that CNN features combined with RF or SVM improved diagnostic performance while maintaining computational efficiency. On the contrary, [156] first applied RF-based feature selection and then trained a deep network, surpassing both shallow and deep baselines for arc-series fault detection. Hybrid strategies are also applied in multi-fault and explainable settings [155].

Although shallow ML is no longer the state-of-the-art for industrial FDD, it remains effective when distinctive fault signatures can be explicitly extracted, labeled data are limited [143], physical interpretability is important, or computational resources are constrained. In contrast, DL dominates in scenarios with high-dimensional, large-scale, and heterogeneous data or when engineered features are not sufficient [161], [162].

The hybrid approaches described above provide a practical compromise.

Deep Learning: Industrial IIoT systems generate large-scale, high-dimensional and heterogeneous data from diverse sources, including physical sensors and/or cyber sources. These data streams often exhibit nonlinear, noisy and nonstationary behavior, making manual feature engineering for traditional ML methods challenging and potentially insufficient for capturing complex fault patterns. DL methods address these challenges by automatically learning hierarchical discriminative features directly from raw data, enabling end-to-end modeling. Architectures such as CNNs, RNNs, Transformers (TFs) or Graph Neural Networks (GNNs) effectively extract spatial, temporal and contextual features across diverse data modalities, achieving state-of-the-art performance.

Despite the end-to-end feature learning ability, studies have demonstrated that data preprocessing remains a critical step also in DL. For physical sensors, for example, techniques such as filtering, normalization, time-frequency transformations or empirical mode decomposition can enhance significantly signal quality and highlight fault-relevant structures, resulting in improved performance and robustness [163], [164].

A trend is multiscale modeling, with the idea of capturing fault signatures that manifest at different temporal, spatial and spectral resolutions. Mechanical degradation process often produce both short-lived transient impulses and long-term effects, which cannot be fully represented at a single analysis scale. Multiscale architectures address this by constructing feature extractors (convolutional or wavelet-based) with multiple receptive fields sizes or frequency bandwidths, followed by adaptive fusion mechanisms such as attention weights or hierarchical pooling. In practice, such architectures tend to have better generalization under varying speeds, loads and noise conditions [165], [166].

In the following an overview of the most used architecture is described. The distinction lies in feature extraction, while the final decision-making is performed by a classification head that maps these learned features to fault labels.

Convolutional Neural Networks: CNNs are the most adopted models in FDD literature [167] due to their ability to learn local structured features from raw sensory inputs, multivariate signal streams or transformed time–frequency representations.

Two principal structural forms are prevalent:

- **1D-CNNs:** operate on ordered 1D sequences and perform convolution along that axis. They are parameter-efficient, preserve the natural ordering of the data and are appropriate when the discriminative information is one-dimensional (for example temporal) with strong local correlations.
- **2D-CNNs:** operate on two-dimensional representations, typically time–frequency images (e.g., spectrograms, scalograms) or spatially arranged multi-sensor maps, and capture joint spatial, time–frequency or cross-sensor patterns via 2D receptive fields.

Beyond these structural forms, numerous model architectures have been adopted, ranging from ResNet, DenseNet, CapsNet and CNNs with dilated convolutions [168], [24].

The 1D-CNNs are the preferred choice for univariate time series, either in the time or frequency domain (e.g., after applying FFT). For example, [169] presents a gearbox fault diagnosis framework under variable operating conditions that uses vibration data and 1D-CNN. To mitigate the effect of variable operating conditions, which increase variability in the learned features of the same fault types and reduce separability between different fault types, the network applies intra-class and inter-class constraints during training. This ensures that features of the same fault type cluster tightly, while features of different fault types remain well separated, enhancing discriminability and improving fault diagnosis robustness. The study in [170] addresses bearing fault diagnosis using motor-current signals. It applies time-shifting, to remove dominant current components, wavelet-based denoising and a lightweight 1D causal-convolutional network that preserves temporal causality while enabling efficient parallel training. The results show higher accuracy and faster training relative to 2D-CNNs and RNNs. In [171], features extracted from multi-modal sensor signals (vibration and acoustic) are fused and fed into a 1D-CNN: empirical evaluation on a dedicated test platform shows that sensor fusion improves diagnostic performance over single-sensor approaches.

To apply 2D-CNNs, one typically need to convert the time series into a 2D input. Some common approaches are:

- **image transformation:** a univariate time series is divided into equal-length segments and stack them as rows of a 2D matrix, with the pixel/cell values being normalized amplitudes of the signal, resulting in a grayscale image [172]. However, the spatial correlations captured by the network may be artificial, since the 2D structure does not necessarily reflect inherent temporal or frequency relationships in the original signal.
- **Multivariate arrangement:** for multivariate time series, each row may correspond to a different sensor channel, and grayscale values represent amplitude, or multiple channels in the image may encode different sensors [173], [174].
- **Time-frequency transforms:** such as wavelet transform [175], [176] or Short Time Fourier Transform [177], [178], [179] to convert mechanical/process time series into 2D time-frequency representations. These advanced techniques often results in better performance [164]. Some recent research embed these transforms directly in the first convolutional layer so that the model can learn the transformation parameters jointly with the classification task (see Sec. III-A2).

Recently, attention mechanisms have demonstrated to be a powerful enhancement to CNNs in fault diagnosis, allowing network to selectively emphasize the most informative parts of the input while suppressing irrelevant information. Depending on the dimension along which adaptive weighting is applied, attention modules in CNNs can be broadly categorized into channel attention, spatial (local) attention and nonlocal or self-attention. Channel attention learn the relative importance of each feature map, enabling the network to emphasize discriminative sensor channels or frequency bands. For example

in [180], a two-branch CNN-based architecture is used for jointly learn bearing fault diagnosis and signal denoising. Channel-wise attention modules in both branches adaptively reweight feature maps according to task relevance, improving robustness under noisy conditions. In [181], channel attention is combined with nonlocal attention in a residual CNN for multi-task diagnosis from multi-sensor vibration data. In [168] channel attention is used in conjunction with multi-feature parallel fusion.

Spatial attention focuses on intra-feature spatial or temporal local dependencies, highlighting the most informative regions or time intervals within feature map. Nonlocal and self-attention extend this principle to a global scale, capturing long-range temporal or contextual correlations beyond the convolutional receptive field [182]. For instance, in [164], raw bearing vibration signals are decomposed using empirical wavelet transform into low-, medium-, and high-frequency components, which are then stacked as channels and processed by a 1D self-attention-enhanced CNN. The attention mechanism adaptively reweights the frequency channels, emphasizing fault-relevant temporal and spectral features. Other studies effectively combine different types of attention, capturing richer feature interactions and improving generalization in many domains, including rotating machinery, industrial processes and industrial robotics [182], [183], [184], [185].

Recurrent Neural Networks.: RNNs and enhanced variants, LSTM, GRU and bi-directional architectures (BiRNNs) remain conceptually appealing for time-series-based fault diagnosis because they implement a principled mechanism for modeling temporal dependencies. In supervised industrial FDD, however, they increasingly appear as components of larger hybrid architectures rather than as standalone predictors. The reason is structural: while recurrent units capture temporal dynamics, they do not represent spatial correlations or multiscale patterns. Hybrid designs fill these gaps and empirical studies shows that they surpass pure recurrent models. An example is the Bi-LSTM-Quadratic NN (QNN) framework in [186]. The Bi-LSTM provides long-range temporal modeling, while the QNN excels at nonlinear feature extraction and offers improved interpretability. Evaluated on both the CWRU dataset and a custom experimental platform for rolling bearing fault diagnosis, the approach outperforms transformer-based and residual architectures, even if computational complexity remain a challenge for deployment in resource-constrained industrial environments.

Hybrid CNN-RNN architectures are widely spread in the literature. Convolutions extract localized and multiscale structure and recurrent units integrate these features across time, surpassing both architectures when used in isolation. In [162], a modified LSTM performs multi-step autoregression, and a 1D-CNN with channel decay handles fault classification on industrial process data with injected faults. A similar approach is presented in the Bayesian-optimized CNN-LSTM model for hydro-turbine FDD in [187]. In there, the CNN constructs a compressed denoised feature basis before the LSTM processes temporal evolution, while Bayesian optimization tunes the hyperparameters. Similar approaches are specifically developed for industrial robots [188] or rotating machinery [189]. More

elaborated designs extend this principle to multi-sensor and multiscale settings. The integrated parallel CNN-deep residual network-LSTM architecture in [190] fuses heterogeneous sensor streams through parallel convolutional branches, followed by a deep residual network and an LSTM module, improving robustness on both cutting-tool monitoring and bearing-fault diagnosis. Extensions combining CNNs, recurrent units and attention mechanisms further enhance the accuracy, as demonstrated in [165], [191].

Overall, these works converge on the same insight: recurrent networks remain valuable, but their strength is highlighted within hybrid models.

Transformers.: Transformers have recently gained traction in time-series modeling and FDD due to their ability to capture long-range dependencies without relying on recurrence. Unlike RNNs, which propagate information sequentially through hidden states, TFs use self-attention to model global relationships across the entire input. This allows scalable parallelization, making TFs particularly effective for large or high-resolution sequences. However, they typically necessitate more training data than recurrent architectures, which can be limited in the data-scarce and imbalanced scenarios often encountered in industrial FDD.

Transformer-based FDD architectures typically use only the encoder, consisting of stacked self-attention and feedforward layers, followed by a classification head. The self-attention mechanism computes contextual relationships across all input positions, whether in the time, frequency, or time-frequency domain, enabling global interaction throughout the representation space.

In mechanical fault diagnosis, early studies applied TFs directly to raw time-domain signals, such as in [192] and [193]. The latter introduces a hybrid CNN-TF architecture to extract multiscale temporal features, where convolutional layers downsample redundant local information to generate refined key-value representations for self-attention. This approach enhances noise robustness and outperforms baseline models, including CNN, CNN-LSTM, RNN, LSTM and GRU on both CWRU dataset and a rotating machinery experimental platform. In the same direction, [194] extends the hybrid CNN-TF framework into a multitask learning setting, enabling simultaneous optimization of fault type and severity while preserving robustness under variable operating conditions and noise.

TFs can also be successfully adapted to the frequency domain. For example, [195] first transforms raw signals into spectral representations via FFT, which are then processed through parallel transformer encoder branches to capture local, medium-scale and global spectral features. Similarly, [196] introduces a global frequency encoding layer within the TF, performing learnable amplitude and phase filtering to facilitate multiscale feature fusion, achieving robust fault detection under high noise levels.

Time-frequency representations are a dominant input modality for TF-based FDD in rotating machinery. In [197], vibration signals are processed using synchrosqueezed wavelet transforms before attention-based feature extraction. In [198], the vibration signal is decomposed into frequency-band sub-

signals via discrete wavelet transform, converted into time-frequency maps using continuous wavelet transform and fed into a multiple Vision Transformers (ViTs), with final classification achieved through soft voting. Similarly [199] combines CNN and TF modules on vibration data, surpassing purely CNN-based methods on a experimental testbed. While most of these studies focus on vibrational data, TFs have also been applied to other signal modalities, such as acoustics [200].

Efforts to reduce TF-based FDD complexity while retaining global dependency modeling include [201], which reformulate self-attention as depthwise convolution. This lightweight design reduces parameters and computational cost by over 60% while maintaining superior performance on bearing and gearbox datasets.

Although TFs generally outperform CNN-based baselines when sufficient data are available, their generalization suffers under limited or imbalanced datasets, a common challenge in FDD [202]. Beyond conventional data augmentation strategies such as SMOTE or GANs [203], [204], tailored solutions have been explored. Notably, [202] treat attention weights as random variables learned via Bayesian variational inference, enhancing robustness to data scarcity and noise. Experiments demonstrate that this Bayesian transformer surpasses ResNet, Bidirectional Long Short Term Memory (BiLSTM), and vision transformer baselines on small-scale datasets, maintains performance under noisy conditions and achieves good cross-domain generalization.

Graph Neural Networks. In many industrial systems, sensors and process components exhibit structured interconnections that can be naturally represented as graphs. Such data are inherently non-Euclidean, making conventional architectures such as CNNs suboptimal. In contrast, GNNs leverage relational structure to learn fault-sensitive representations that capture both local and global dependencies among process variables.

Formally, a system at a given time (or over a time window) can be represented as a graph

$$G = (\mathbf{X}, \mathbf{A}),$$

where $\mathbf{X} \in \mathbb{R}^{N \times F}$ is the node feature matrix (each row encodes F features for one of the N nodes) and $\mathbf{A} \in \mathbb{R}^{N \times N}$ is the adjacency matrix defining pairwise relationships among nodes. Under normal operating conditions, the system graph is denoted as $G = (\mathbf{X}, \mathbf{A})$, while under faulty conditions it becomes $G_f = (\mathbf{X}_f, \mathbf{A}_f)$, since both node features and topological relations may vary due to faults. Hence, a GNN can, in principle, detect and localize faults whose signatures manifest either in node attributes or in the graph topology [205].

To apply GNNs, two main steps are typically required:

- 1) **Graph construction:** define the set of nodes, their feature vectors, and the adjacency matrix encoding pairwise relationships.
- 2) **Model selection, training, and inference:** choose a GNN architecture and train it to propagate information through message passing, producing node or graph embeddings used for fault classification.

Regarding graph construction, the definition of nodes and edges depend on the data source and, for time series, in the

univariate or multivariate nature. For univariate time series, two main modeling approaches are commonly adopted [206]:

- **Series-as-graph:** each time series is segmented into windows, and each window is treated as a node. Features extracted from each segment form the node attributes, while the adjacency matrix encodes relationships among segments. The task is graph classification: extracted node embeddings are aggregated into a graph-level representation, on which the final classifier operates to predict the fault class for the entire time series [207], [208], [209].
- **Series-as-node:** each complete time series in a dataset is assigned to a single node, with statistical or learned features forming the node attributes. The task is node classification: the classifier outputs a label (or probability) for each node. This configuration enables modeling of inter-sample relationships through the adjacency matrix, for instance by encoding similarities between healthy and faulty conditions [210].

For multivariate (multi-sensor) time series, inter-variable dependencies can be modeled explicitly. Each sensor or process variable corresponds to a node, and its measurements or extracted features (statistical, temporal, or spectral) form the node attributes. The adjacency matrix encodes relationships among sensors, based on physical connectivity (e.g., from process topology [211]), statistical dependencies or learned during training. Single graphs are usually defined per time window to represent the system state. Some strategies do not explicitly model temporal continuity between windows/graphs [212], [213], [214], treating each snapshot independently. Other studies, instead, jointly capture both spatial and temporal dependencies by constructing spatio-temporal graphs. In this case, a sequence of graph instances:

$$\{G_t = (\mathbf{X}_t, \mathbf{A}_t)\}_{t=1}^T$$

is defined over consecutive time intervals, where both $\mathbf{X}_t$ and $\mathbf{A}_t$ may evolve with time. The GNN-based model used to deal with such representations, called Spatio-Temporal Graph Neural Network (STGNN), comprises two interacting modules: a spatial module (GNN, Graph Convolutional Network (GCN), ...) operating on each G_t to capture inter-sensor dependencies at each time step, and a temporal module (e.g., RNN-, Temporal Convolutional Networks (TCN)- or attention-based) to model temporal dynamics, either in time or frequency domain [215], [216], [217]. Alternative formulations also exist. For instance, [218] divide multivariate time series into non-overlapping windows, each containing data from all sensors. Each window is treated as a node in the graph, reformulating the problem as node classification.

Regarding edges/adjacency matrix construction, the literature distinguishes between heuristic-based and learning-based approaches. Heuristic methods rely on pairwise similarity, where edges are created between nodes with similar features (e.g., using cosine similarity [207], [219] or Pearson correlation [210], [216]). To maintain sparsity, only the top-k most similar nodes are typically retained, forming a k-NN graph [220], [218]. Other heuristics capture functional dependencies, where edges represent causal relationships (e.g.,

inferred from Granger causality tests [221]) or reflect known physical process topologies [212], [222]. An increasingly adopted alternative is learning-based adjacency construction, where the matrix $\mathbf{A}$ is optimized jointly with the downstream task and defined as a function of trainable parameters and input features [223]. A common formulation expresses $\mathbf{A}$ via attention mechanisms, as in Graph Attention Network (GAT) [224], which assign data-dependent edge weights based on node relevance. Attention can be static, when learned once and fixed during inference, or dynamic, when recomputed adaptively [215], [213], [217].

In the input layer of the GNN, each node can be associated with raw or transformed signals, or with extracted feature vectors. In the latter case, the features can be obtained manually, based on expert-designed statistical, temporal, or spectral descriptors [221], or automatically, for example using CNN [223], in an end-to-end manner. In particular, in [223], the authors transform the CNN features into node embedding through a dynamic graph embedding module. Then, the constructed graph is fully connected, with an edge-dropout mechanism, enabling the diagnosis of rolling bearing faults under variable working condition.

Once the graph is defined, different GNN-based architectures can be used:

- GCN [225]: performs graph convolutions to aggregate feature information from neighboring nodes according to the adjacency matrix $\mathbf{A}$. Two main formulations exist: *spectral* and *spatial*. Spectral GCNs define convolution operations in the graph Fourier domain, using the graph Laplacian eigenbasis, and are effective when $\mathbf{A}$ reflects global statistical relationships or spectral similarity among features (e.g., in vibration or frequency-domain graphs). Spatial GCNs, instead, aggregate messages directly over local neighborhoods and are more efficient and interpretable for process or sensor networks with explicit topological connections. Both are particularly suitable for systems with relatively static process structures, where physical or empirical couplings among sensors remain approximately constant over time.
- GAT [224]: generalizes GCN by introducing learnable, data-dependent attention coefficients. Instead of uniformly weighting neighbors, each edge e_{ij} receives a learned attention score reflecting the relative importance of node j to node i . This mechanism enables the model to adaptively focus on fault-relevant relationships, for example when correlation strengths between sensors change under fault conditions. Multi-head attention further stabilizes training and enhances feature discrimination.
- GraphSage [226]: addresses inductive learning—generalization to unseen nodes or graphs—by learning an aggregation function rather than relying on the full adjacency matrix. Instead of computing convolutions over the entire graph, GraphSage samples a fixed-size neighborhood and aggregates its features via mean, max pooling, or LSTM-based operators. This sampling-based strategy allows generalization to unseen nodes or new operating conditions, advantageous in dynamic IIoT environments.

- STGNN: designed for dynamic processes in which fault evolution exhibits both spatial and temporal dependencies. The spatial module (often a GCN or GAT) captures inter-sensor correlations at each time step, while the temporal module (e.g., RNN, TCN, or attention mechanism) models temporal evolution. This joint modeling enables learning spatio-temporal fault signatures.

Beyond conventional GNN-based architectures, recent research has explored more flexible frameworks. Some works explore more discriminative architectures, for example Graph Isomorphism Networks [227]. A major direction is hierarchical or multi-scale modeling, where nodes or subgraphs are progressively coarsened into higher-level abstractions to capture fault patterns across different structural or temporal resolutions [166]. For instance, [228] introduces a hierarchical supergraph framework for rotating machinery fault diagnosis. Segments from a vibration signal are first represented as spatio-temporal graphs, whose nodes correspond to frequency bins from the STFT and edges encode spectral proximity. Each graph is embedded into a low-dimensional representation and it is then treated as a node in a higher-level supergraph. This design reformulates the original graph-classification task as meta-level node classification. Similarly, [216] combines local subsystem graphs and a global graph to jointly model intra- and inter-unit interactions within industrial systems.

Large Language Models.: In the last couple of years, the proliferation of Large Language Models (LLMs) beyond conventional text generation tasks has also prompted interest in their use for FDD. Emerging research on integrating LLMs into industrial FDD can be broadly grouped based on the role the LLM plays in the diagnostic pipeline: (i) as the primary fault detector, or (ii) as supporting component.

In the first paradigm, the LLM serves as the primary fault detector. Some approaches leverage in-context reasoning using knowledge graphs (KGs). For example, [229] incorporates a causal knowledge graph via prompt-tuning to guide quality defect analysis in aerospace manufacturing, while [230] exploits the LLM to construct a Dynamic Master Logic (DML) knowledge graph from unstructured documentation for nuclear power plant subsystems. In this latter framework, the LLM effectively extracts DML logic, links functional models to natural-language reasoning, interprets user queries, selects reasoning tools to propagate faults through the KG, and retrieves relevant KG segments to generate human-readable explanations. Another major class fine-tuned the LLM on sensor data. The LLM is fine-tuned on paired datasets, where the input is a processes representation of the sensor signal, eventually augmented with context-information, and the label is the corresponding fault information, often with explanatory text. For example, [231], [232] convert vibration features into semantic text enriched with bearing geometry, operating conditions, and characteristic fault frequencies; these texts are paired with fault type and severity labels for fine-tuning. Similarly, [233] constructs a multimodal dataset of envelope spectral images paired with textual descriptions of the fault class and underlying mechanisms; during inference, new spectral images are converted to the same representation, and the fine-tuned LLM predicts the fault class while generating explanations. More

complex pipelines, such as [234], first embed raw time-series signals via an LSTM and align them with textual embeddings using contrastive learning; fuzzy semantic embeddings handle ambiguity across similar fault patterns, prompt learning injects domain knowledge, and then a fine-tuning adapts the model for inference. The authors in [235] conduct an empirical study on fine-tuning LLM for fault diagnosis in complex systems, using both a simulated chemical process dataset (TEP) and a real high-speed train braking system dataset. The study demonstrates high diagnostic accuracy and strong few-shot learning performance, with the models performing well even when data are limited or imperfect. The effectiveness of LLMs is further enhanced by incorporating contextual information, while smaller models offer greater efficiency with minimal accuracy loss. However, despite their predictive strength, the LLMs exhibit limited explainability, often producing reasoning that is unreliable or inconsistent.

In the second paradigm, the LLM functions as a supporting component rather than directly predicting faults. LLMs help to construct or refine knowledge graphs [236], manage human-in-the-loop workflows by overseeing data preprocessing, model selection, and result explanation [237]. For example, [236] leverages an LLM to build detailed ontologies for robotic fault diagnosis and to act as a “teacher” that refines synthetically generated training data, improving model reliability in low-data environments. Similarly, [237] implements a human-in-the-loop pipeline where the LLM manages preprocessing and visualization, corrects causal relations obtained from standard algorithms using domain knowledge, controls model selection and updates, and explains diagnostic results in plain language, while humans provide feedback to ensure accuracy and consistency. An exhaustive review is provided in [238].

Physics-Informed Machine Learning: Previous approaches to FDD often face challenges like scarcity of labeled fault and the inherent “black-box” nature of many models. Physics-Informed Machine Learning (PIML) mitigates these issues by embedding physical domain knowledge into the diagnostic framework, shifting FDD from a purely statistical pattern-recognition task to a more physics-consistent inference task. This enhances model interpretability, diagnostic accuracy and robustness under varying operational conditions.

While physical models can be employed either to generate synthetic faults for data augmentation [239], [240] or to introduce new features based on system equations, enhancing discriminativity [241], PIML goes further. It incorporates physical laws directly into the model structure or the learning process. Following [242], two categories can be identified:

- 1) Physics-informed regularization, and
- 2) Physics-guided architectures.

In physics-informed regularization, physical laws appear explicitly in the training loss function:

$$\mathcal{L} = \mathcal{L}_{\text{accuracy}} + \lambda \mathcal{L}_{\text{physics}}. \quad (1)$$

where $\mathcal{L}_{\text{accuracy}}$ measures prediction error with respect to labeled data, $\mathcal{L}_{\text{physics}}$ penalizes violations of known physical constraints. A widely adopted example of this idea are Physics-Informed Neural Networks (PINNs), which incorporate differential equations describing system dynamics into the loss

function. In FDD, these losses have been used, for example, to estimate physical parameters which are interpretable fault indicators. For instance, [243] trained a PINN to model pressure dynamics in axial piston pumps, an indicator of wear severity, embedding conservation law of mass and fluid compressibility, while [244] applied multistage PINN to estimate valve flow characteristic under normal conditions, ensuring physical consistency via valve sizing equations. The residuals between predictions and observed values are analyzed to detect anomalies. Authors of [245] use a PINN to estimate five parameters that characterize the health state of rotating shafts (i.e., the radial and angular position of the disk-induced static unbalance, the stiffness along the principal elastic axes, and the damping coefficient in the non-rotating reference frame). Alternative approaches enforce energy conservation in rotating machinery under normal and faulty operation regimes and using the resulting estimates as diagnostic features for downstream classifiers [246], constrain decisions to agree with coarse physics-based threshold models to stabilize critical classes [247], or align the model to approximate physics-based labels [248]. Recent extensions apply partial derivative equations residuals together with boundary conditions to hydraulic pipelines and pump flow-ripple prediction [249].

On the other hand, physics-guided architectures embed physics directly into the network structure. A representative family of methods replaces or initializes early convolutional kernels with physically motivated filters. [250] designed a modular CNN, for bearing fault diagnosis, with one specialized feature extractor module for each type of fault. The first convolutional layer of each module is replaced with kernels corresponding to analytically derived fault characteristic frequencies. The features are then combined to make the final classification. Similarly, [251], [252], [253], [254] initialize convolutional filters with wavelet functions (e.g., Morlet or Laplace) providing a physically interpretable basis for vibration analysis while still allowing the filters to be fine-tuned during training. In [255], a continuous wavelet layer replaced the first layer of a CNN, where only the scale and translation parameters are learned. Deterministic modal-property layers explicitly extract modal features that are invariant to operating speed and excitation, improving cross-condition robustness [256]. In [257], a parametric spectral-weighting layer learn a weighting function over the envelope/order spectrum that emphasizes analytically derived characteristic frequencies. The weighting parameters are trained jointly with the classifier to maximize discriminativity while remaining physically interpretable. Finally, physics-informed priors is also used to guide hyperparameter selection [258].

Deep Reinforcement Learning: The architectures discussed so far have been trained under the supervised learning paradigm, relying on labeled pairs of examples $(x_i, y_i)_{i=1}^N$. In contrast, Reinforcement Learning (RL) follows a fundamentally different trial-and-error approach, in which an agent learns to select *actions* based on a *reward signal* that indicates the effectiveness of those actions, particularly suitable for sequential decision making. Deep RL has also been applied to diagnostic tasks.

In a large portion of studies, FDD is formulated as a

single-step decision making problem or "guessing game" problem [24]. The RL agent receives a portion of a sensor signal (vibrational, acoustic, etc.) as its *state*/input and selects an action corresponding to a diagnostic decision. In a K -class classification problem, the action space contains K elements (one for each fault type plus the absence of the fault). The reward signal replaces traditional labels, providing positive feedback for correct diagnoses and negative feedback otherwise. These formulations have demonstrated effectiveness in a range of FDD applications, including rotating machinery, hydraulic pumps, gearboxes and nuclear power plant equipment [259], [260], [261], [262]. Differences consist in various learning algorithms (e.g., Q-learning, REINFORCE, etc.), underlying architectures that constitute the feature extractor parts of the agent (e.g., CNNs, attention-based models), preprocessing strategies and datasets used for evaluation. Since RL does not solve the problem of class imbalance, [263] introduces a data-augmentation method based on a generative model that combines a Variational Autoencoder with a Wasserstein Generative Adversarial Network and Gradient Penalty, before training an RL agent. This approach enhances training data diversity and has demonstrated effectiveness in both laboratory experiments and electric locomotive applications.

While these guessing-game formulations are advantageous in scenarios with nonstationary or evolving data, where online RL allows the agent to adapt to changing patterns, the benefit for static datasets, where all labels are already available, is less clear. In such cases conventional supervised learning may be generally straightforward and efficient, as RL may require extensive data interactions.

Beyond direct fault classification, RL has been employed to optimize diagnostic model components. Neural architecture search is one such application, where RL agents explore, for example, candidate CNN architectures and receive rewards based on resulting accuracy [264]. Similarly, RL has been applied to hyperparameter optimization, as adaptive learning-rate scheduling [265].

Other approaches use RL to mitigate class imbalance typical of FDD. In [266], an agent learns adaptive sample weights to scale a cross-entropy loss of a downstream classifier. The rewards are derived from the classifier performance, especially on minority classes. An alternative approach treats imbalance as a sample-selection problem, modeled as a multi-armed bandit [267]. Here, the agent selects a subsample of points for training of a separate classifier, with rewards based on its validation performance. The method outperforms conventional oversampling and undersampling techniques. A similar methodology has been successfully applied to HVAC systems [268]. While effective, these approaches require iterative retraining, which can introduce substantial computational overhead.

Challenges:

Scarce and imbalanced datasets: Since faults cannot be freely generated in industrial environments, real-world datasets for FDD are typically small and highly imbalanced. In standard supervised learning, this imbalance leads to poor classification performance on minority fault classes, degrading overall diagnostic accuracy. Data augmentation techniques aim

to mitigate this issue by generating synthetic samples from existing data. Oversampling approaches are the most common, with SMOTE [269] being a widely adopted example [24]. Transformations in the time or frequency domains are also possible. Generative models such as VAE or GAN have been explored for fault data synthesis [270], [271], [272]. In a real manufacturing plant, [273] embeds GAN-based synthetic data generation into an automated MLOps pipeline [274] to compensate for the scarcity of defective-component data in fault detection. Although these methods can model complex data distributions, they are less straightforward to apply and require careful training generating realistic and diverse samples.

A particularly promising approach is Transfer Learning (TL), which enables the transfer of knowledge from a source domain, typically consisting of abundant and well-labeled data, to a target domain characterized by limited, imbalanced or unlabeled samples. This capability enables improved model generalization under challenging data conditions. For example, TL allows models training on a large public dataset (like CWRU) or simulated and laboratory-controlled data (source domain) to be fine-tuned for deployment in specific industrial environments with distinct operating conditions (target domain). This setting is often referred to as cross-domain fault diagnosis and conventional DL models may fail because shifts in data distributions. Parameter-based and feature-based TL using deep architectures have proven effective [275], as they can extract invariant feature representations across domains [276], [24].

3) *XAI in FDD:* Most FDD approaches, whether based on traditional ML or DL models, function as "black boxes," providing little insight into how fault-related decisions are made. For example, it is often unclear which process variables, signal features, or temporal patterns contribute to a given fault detection or classification outcome. While Physics-Informed Machine Learning can naturally enforce physically consistent reasoning and improve interpretability, it is not always feasible due to incomplete knowledge of system dynamics, unmodeled interactions, or complex process behaviors. In such cases, Explainable Artificial Intelligence techniques become essential, not only for validating model reasoning against domain knowledge (with *global* explanations), but also for supporting effective root-cause analysis, enhancing the actionability of diagnostic results, and enabling operators to make informed, trustworthy decisions (with *local* explanations), in line with the Industry 5.0 paradigm.

The majority of the FDD literature applies post-hoc XAI methods to explain pretrained models, rather than relying on inherently interpretable architectures. This preference arises from the typically superior predictive performance of complex, opaque models. Among post-hoc techniques, feature relevance scores (also referred to as feature importance) are the most common modality [277]. In FDD tasks, which are generally formulated as classification problems, these approaches explain the predicted class probability rather than the final discrete decision. The model-agnostic SHAP [278] is the most widely adopted method due to its solid theoretical foundation, ease of use, and ability to provide both global and local explanations [279], [280], [113], [281], [282], [283].

TABLE VI
SUMMARY OF FDD METHODS IN MANUFACTURING AND INDUSTRIAL IIOT.

Method	Description	Notes
Shallow ML	Feature engineering (time-, frequency- or time-frequency-domain) → traditional ML classifier (e.g., SVM, RF).	Suitable when data size is moderate and features are well understood. Works well for interpretable, low-latency and edge deployable FDD. Good for proof-of-concept with low compute cost. Less effective for high-dimensional or time-dependent signals. Relies heavily on domain expertise for feature extraction.
CNNs	Automatically learn local spatial or temporal features from raw or transformed sensor data (e.g., 1D vibration or 2D spectrograms).	Appropriate when local fault signatures dominate (e.g., impact transient) and sufficient labeled data are available. Less effective for long-range temporal patterns or strongly non-stationary processes.
RNNs/LSTMs/GRUs	Capture temporal dependencies and degradation dynamics.	Appropriate when fault behavior evolves over time (wear, drift, progressive failures) and sequential relationships are key. Handles long time windows better than CNNs but higher computational cost, but impractical for very long sequences.
CNNs+RNNs (Hybrid)	Combine convolutional feature extraction with recurrent temporal modeling.	Best suited when short-term and long-term signal patterns are relevant. Strong diagnostic accuracy. Demands larger, well-labeled dataset and more computational power.
Transformers	Model global temporal and cross-sensor dependencies using attention mechanism.	Preferred for high-dimensional, multi-sensor or multi-modal setup. Enable transfer learning. Requires large datasets, high compute and careful regularization.
GNNs	Represent sensors, process units as graph nodes with physical or statistical interconnections.	Capture structural and non-euclidean correlations that CNNs and RNNs ignore. Need proper graph definition (either learned or manually defined). Powerful for multi-machine or networked IIoT monitoring.
PIML	Integrate process physics, conservation laws, analytical models into learning to guide or constrain predictions.	Valuable when labeled fault data are scarce but physical understanding is strong. Physically consistent and explainable outcomes. Flexibility is limited, performance may drop for unmodeled failure modes.
RL	Learn optimal diagnostic policies via interaction with the environment or digital twin.	Require a reliable simulator/safe experimentation, training is resource-intensive.

LIME [115], which offers local explanations, is also used, though less frequently [155]. In [284], when detecting faults on a rotating electric motor testbed using RF, the authors state that LIME is especially beneficial for understanding the reasons behind single predictions, particularly with LIME rules provided for RF. Other model-agnostic, feature-importance-based techniques, such as Permutation Feature Importance and Partial Dependence Plots, are less frequently applied in recent studies [285], [277].

When dealing with particular types of classifiers, model-specific XAI methods can be applied to exploit intrinsic model properties and produce more faithful or computationally efficient explanations. For instance, ensembles of decision trees, such as RF, provide built-in feature importance measures, which have been used to provide explanations for rotating machines using multi-sensor fusion [155].

Gradient-weighted Class Activation Mapping [286] is a widely used visualization technique for CNNs that highlights the regions of the input most influential in a specific model

prediction. It is particularly popular in FDD, where it is employed to generate heatmaps over 2D images [287], as well as over 1D time-series data, both in the time domain [209], [288] and in the frequency domain [209], [289], [290], [291], [292], where it identifies the frequency bands most responsible for a particular decision. Building upon this, [293] proposed applying Gradient-weighted Class Activation Mapping (GradCAM) to CNNs trained on fuzzy-based energy pattern images rather than raw sensor signals, enhancing interpretability. The approach was validated on a robotic post-welding application. Many variations of GradCAM have been proposed. The study in [294] introduced a multi-layer GradCAM variant that considers gradient flow across multiple CNN layers to improve explainability in vibration-based fault diagnosis for industrial equipment. Another adaptation, specifically designed for HVAC fault diagnosis, was presented in [295]. In parallel, a few studies have explored Layer-Wise Relevance Propagations (LRPs) [296], a technique that

decomposes the model output backward through the network layers to assign a relevance score to each input feature. This allows identification of the parts of the input that contribute most to the decision. LRP has been applied, for example, in gearbox fault diagnosis with CNNs [297], in chemical processes [298] or HVAC [299]. Moreover, other studies have explored integrated gradients [300], [301]. Beyond standard XAI methods, novel explanation strategies for FDD have been proposed. For instance, [291] introduced a frequency-domain perturbation-based method to visualize and assess the decision boundaries of vibration-based classifiers. After selecting the most relevant frequencies using GradCAM, the authors perturb the power spectrum of a given sample around these dominant frequencies, varying bandwidth and intensity, to evaluate and visualize the robustness of the classifier. XAI, and particularly feature-importance analysis, is not only used for model interpretation but also for feature selection and model refinement. Studies such as [285], [298], [302], [301] demonstrate how interpretability can guide the identification of the most discriminative features, improving the performance of fault classifiers.

Even if other XAI modalities exist, such as counterfactual, example-based and concept-based explanations, their application remains very limited in FDD [277].

While post-hoc XAI methods dominates the FDD literature, an increasingly important direction focuses on models that are inherently interpretable by design. One branch of this research aims to improve interpretability by encouraging sparsity in the model parameters [254], [303], [304]. Beyond its interpretability benefit, sparsity is also advantageous in fault diagnosis for rotating machinery. As highlighted in [305], vibration signals from faulty components often exhibit sparse spectral signatures. Sparse representation therefore provide a natural way to isolate these fault-related patterns and can facilitate downstream tasks, including data generation [306].

Other work uses logical inference to obtain symbolic expressions that describe systems properties directly from data, an approach consistent with neurosymbolic AI. In this setting, the diagnostic task is formulated as the generation of expression in a formal language, for example temporal-logic formulas, and these expressions act as explicit classifiers. The interpretability follows immediately from the fact that the learned rules have a fixed syntax and well-defined semantics. Recent studies to interpretable FDD effectively use temporal-logic-based languages, learning the parameters of temporal operators [307], [308], [309], [310], [311]. To enhance deep learning models inherent interpretability for FDD, *algorithm unrolling* is an emerging alternative [305]. It transforms iterative optimization or signal-processing algorithms into neural network architectures. Each iteration of the original algorithm corresponds to a layer in the network and algorithm parameters are made trainable, allowing end-to-end optimization while retaining the interpretability of the original procedure. This strategy has been applied to sparse coding, where signals are represented as linear combinations of few dictionary elements. For example, convolutional sparse coding solved via iterative shrinkage-threshold algorithm (ISTA) was unrolled into CISTA-Net [312]. Extensions further incorporated multi-

layer sparse coding formulations that preserving theoretical transparency [313], [314].

Attention mechanisms are also considered to enhance interpretability. Some studies go beyond pure attention, for example in [315], a Variational Attention-based Transformer Network is proposed. Unlike standard transformers, whose attention weights may lack physical meaning, their model introduces probabilistic and sparsity constraints through variational inference with Dirichlet priors. This forces the attention mechanism to focus on a few fault-relevant impulsive segments, aligning model behavior with the physical characteristics of vibration signals, offering both high diagnostic accuracy and causally interpretable attention maps.

Finally, PIML, which is thoroughly discussed in Section III-A2, is another form of inherent model interpretability [305].

4) *Summary of FDD Methods:* Table VII provides a comprehensive summary of the FDD methods discussed in this section, organized by category and approach, along with their key references.

B. FDD Industrial Applications

In this section, the most common applications of FDD in the industrial domain are analyzed. We follow the same organization as in Section II-B, so that the reader can directly compare unsupervised anomaly detection approaches with supervised fault detection methods applied to the same use cases.

A non-exhaustive overview of specific papers, divided by application domain, is reported in Table VIII, while more general considerations are made in the subsections.

1) Mechanical/equipment faults:

Rotating Machinery: Rotating machinery is the most extensively studied application domain in the FDD literature for manufacturing systems. Commonly investigated fault types include bearing defects (e.g., inner-race, outer-race and rolling-element faults), shaft-related faults (such as rotor imbalance and misalignment) and gear/gearboxes faults (e.g., missing, broken or chipped teeth, surface wear, spalling and root cracks). Scenarios involving multiple simultaneous faults are also considered [318].

Vibration signals are by far the most commonly used data source, typically acquired via accelerometers and sampled at rates sufficient to capture characteristic fault frequencies [223] and suitable for early FD. Additional measurements, such as temperature and torque are sometimes used [24].

From a methodological perspective, the literature shows a clear evolution from feature-engineered pipelines, based on frequency- or time-frequency features [143], [148], toward deep learning models operating directly on time-frequency inputs. In this domain, CNNs are the dominant architecture [24], [271], [178], [175], even if hybrid LSTM-CNN models are also frequently investigated. More recent work explores newer architectures, such as TF [193], [202], often in combination with synthetic data generation strategies [272], attention mechanisms [164], [185], [168], [184], or graph-based models [135]. Due to its scale, this application is the

TABLE VII
SUMMARY OF FDD METHODS REVIEWED IN SECTION III-A, ORGANIZED BY CATEGORY AND APPROACH.

Category	Approach	Key Methods/Techniques	References
Traditional ML	Feature engineering	FFT, WT, EMD, STFT, time/frequency/time-frequency features	[24], [139], [140], [132], [141], [142]
	Classifiers	RF, XGBoost, LightGBM, SVM, k-NN, shallow NN, ensembles	[143], [144], [145], [146], [147], [148], [149], [150], [151], [152], [153], [154], [161]
	Feature selection	Recursive elimination, importance-based, correlation, optimization	[148], [156], [157], [145], [158]
	Hybrid DL-ML	CNN features + RF/SVM, DL extraction + shallow classifier	[159], [160], [156], [155], [162]
CNNs	1D-CNN	Time/frequency domain convolutions	[169], [170], [171], [167]
	2D-CNN	Image transformation, multivariate arrangement, time-frequency transforms	[172], [173], [174], [175], [176], [177], [178], [179], [164], [163]
	Attention-enhanced	Channel attention, spatial attention, self-attention	[180], [181], [168], [182], [164], [183], [184], [185]
RNNs	Enhanced variants	Bi-LSTM, QNN hybrids	[186]
	CNN-RNN hybrids	CNN + LSTM, parallel multi-sensor fusion	[162], [187], [188], [189], [190], [165], [191]
Transformers	Time domain	Raw signal transformers, CNN-TF hybrids	[192], [193], [194]
	Frequency domain	FFT-based, spectral encoding	[195], [196]
	Time-frequency	Wavelet transforms + ViT, synchrosqueezed WT	[197], [198], [199], [200]
GNNs	Lightweight/robust	Depthwise convolution, Bayesian attention	[201], [202], [203], [204], [316]
	Univariate graphs	Series-as-graph, series-as-node	[207], [208], [209], [210], [206]
	Multivariate/STGNN	Spatio-temporal graphs, inter-sensor modeling	[211], [212], [213], [214], [215], [216], [217], [218], [205]
	Adjacency learning GNN architectures Advanced architectures	Similarity-based, attention-based, causal GCN, GAT, GraphSAGE Hierarchical, supergraph, multiscale, uncertainty-aware	[207], [219], [210], [221], [222], [223], [220] [225], [224], [226] [227], [166], [228], [317]
LLMs	Primary detector	Knowledge graphs, fine-tuned on sensor data	[229], [230], [231], [232], [233], [234], [235]
	Supporting component	KG construction, human-in-the-loop	[236], [237], [238]
PIML	Data augmentation/features	Synthetic fault generation, physics-based features	[239], [240], [241], [242]
	Physics-informed regularization	PINNs, conservation laws, physics losses	[243], [244], [245], [246], [247], [248], [249]
	Physics-guided architectures	Wavelet kernels, modal layers, spectral weighting	[250], [251], [252], [253], [254], [255], [256], [257], [258], [166]
Deep RL	Direct classification	Q-learning, REINFORCE for FDD	[259], [260], [261], [262], [263]
	Model optimization	Neural architecture search, hyperparameter tuning	[264], [265]
	Imbalance handling	Adaptive sample weighting, multi-armed bandit	[266], [267], [268]
Data Challenges	Data augmentation	SMOTE, VAE, GAN synthesis	[269], [270], [271], [272]
	Transfer learning	Cross-domain adaptation, feature-based TL	[275], [276]

only one in which niche approaches like RL and LLMs have been studied [262], [231], [232].

Practical considerations. TL is often employed to mitigate the gap between models trained on simulated/lab-controlled datasets and deployed in real industrial plants [275], the gap across different machines or operating conditions. When computational resources or labeled data are limited, feature-engineered representations combined with traditional ML may be preferable. Conversely, when richer datasets are available, CNN-based models operating on spectrograms are typically adopted.

As highlighted in [24], FDD for rotating machinery presents several open challenges, including complex motion dynamics and frequent occurrence of multiple interacting faults, which complicate the identification of characteristic fault frequencies, external noise, non-stationary fault behaviors and strong

dependence of diagnostic performance on sensor placement.

Tool wear and breakage: Tool wear monitoring is a key task in manufacturing, as undetected wear can degrade machining quality. In milling, turning and drilling, tool wear monitoring is typically formulated as an estimation problem, with the objective of quantifying the tool's current state. The most widely used metric is *flank wear*, which directly reflects tool degradation.

ML-based monitoring is generally performed online using indirect sensor measurements, enabling data collection and estimation without stopping the machine. Cutting forces are the most used signals [319], followed by vibration and acoustic emissions [320].

The SVM and NN models dominate traditional approaches, using time, frequency and time-frequency features. SVM often shows superior performance even with respect to DL on small

sample sizes [149], [319]. However, especially in milling applications, CNN-based models, sometimes combined with LSTM, represent the current state of the art [321], [322], [323], [324], [325], [326], [327]. Emerging methods have been proposed to reduce the required number of training samples, such as GNNs-based one [323]

Hydraulic and pneumatic systems: Hydraulic and pneumatic systems are widely employed for actuation, clamping and material handling in manufacturing and process industries.

The most commonly investigated fault types include leakage, valve-related faults (e.g., sticking, clogging, spool wear, solenoid failures), pump and compressor faults (wear, cavitation, seal degradation), actuator faults (e.g., cylinder seal wear) and sensor malfunctions. In hydraulic subsystems, faults are frequently reported in pumps, valves, cylinders, and hydraulic motors, with axial and radial piston pumps being among the most extensively studied components.

A wide range of sensing modalities is employed. Pressure signals directly reflect energy losses, flow restrictions and system dynamics. Additional measurements include vibration, acoustic emission, flow rate, temperature and current, often collected in multi-sensor configurations to improve fault observability. However, unlike rotating machinery, vibration transmission in fluid power systems is strongly affected by fluid resistance and pressure losses, leading to more attenuated and indirect fault signatures.

From the methodological perspective, early studies rely on feature-engineered representations combined with traditional ML classifiers, suitable for small-sample or low-computational settings. An example is given in [328], where an SVM pipeline using variational mode decomposition and evolutionary optimization strategy is used for hydraulic pump diagnosis. Classical pipelines are also employed in more structured fault-detection workflows, such as the multi-step electromagnetic hydraulic valve group diagnosis method in [329], which combines time-frequency analysis, similarity-based health indicators, statistical change detection, and SVM-based fault classification.

However, also in this application, recent work adopt DL architectures. CNNs, RNNs and hybrid CNN-RNN models dominate the literature, often enhanced with attention mechanisms to improve feature discrimination under varying operating conditions. For example, [187] proposes a Bayesian-optimized CNN-LSTM architecture for hydro-turbine fault diagnosis using acoustic vibration signals, demonstrating improved robustness over standard deep models. Multi-channel and multi-modal fusion strategies are also widely investigated, aiming to exploit heterogeneous sensor data and different sampling frequencies. Representative examples include attention-based multi-sensor fusion networks [330], adaptive multichannel time-series fusion frameworks coupled with attention-integrated residual networks [331], and transformer-based models designed to capture multiscale interactions in hydraulic pump signals [332]. A distinctive trend in this domain is the increasing integration of data-driven learning with physics-based knowledge. Digital twin technologies are used to generate labeled fault data under controlled conditions, as in [333], where domain-adaptation Wasserstein GANs are employed to

bridge the distribution gap between simulated and real data before training a CNN-GRU-attention model for directional valve diagnosis. Physics-informed neural networks further extend this paradigm by embedding hydraulic principles directly into the learning process. Examples include multistage PINN frameworks for regulating valve fault detection in nuclear power plant hydraulic systems [244], and inverse PINN-based approaches that reconstruct pump flow ripple from pressure measurements to diagnose axial piston pump faults without direct flow sensing [249]. Transfer learning and hybrid residual-based strategies are also explored to address data scarcity and domain shift. For instance, [334] combines residual multi-head deep models with transfer learning for hydraulic pump diagnosis using vibration signals, improving generalization across operating conditions. Siamese neural networks have been proposed for piston pump wear detection under small-sample regimes, addressing the limited availability of labeled industrial fault data.

Electrical drives, induction motors: Electrical drives are another major application domain in ML-based FDD literature. Among them, induction motors and permanent magnet synchronous motors (PMSMs) are of particular interest.

Reported fault classes span electrical, mechanical, magnetic and drive-related categories. Electrical faults include stator winding short circuits, broken rotor bars and cracked end rings. Mechanical faults encompass bearing defects, air-gap eccentricity, rotor unbalance and shaft misalignment. Drive-related faults affecting auxiliary subsystems, such as cooling system failures, are also investigated [136]. In PMSMs, additional magnetic faults are considered, e.g., partial or irreversible demagnetization [335].

Current signals are among the most common data source for diagnosis electrical faults [336], [337], [338], [170], owing to their non-intrusive acquisition. Voltage measurements are sometimes combined with current signals to enhance model performance [161]. For faults with a dominant mechanical or thermal nature, vibration, temperature and acoustic emission signals are frequently employed, either individually [339], [340] or within multi-sensor configurations [341], [342]. To train a model to detect both electrical and mechanical faults, vibration signals can be used [343].

Methodologically, a considerable amount of the literature adopts feature-engineered pipelines based on spectral and time-frequency representations, including Fast Fourier Transform, Short-Time Fourier Transform, continuous and discrete wavelet transforms, empirical mode decomposition, Hilbert and Park transforms [344]. These representations are typically combined with classical machine learning classifiers such as support vector machines (SVM), k-nearest neighbors (kNN), random forests (RF), multilayer perceptrons (MLP), and tree-based ensemble methods [335], [148], [339]. In parallel, deep learning approaches have gained increasing attention. CNNs are the dominant architecture, often operating on transformed signal representations to enable automated feature extraction [179], [170]. For example, [337] employs bispectrum-based preprocessing of current signals as input to a pretrained CNN for PMSM stator winding fault diagnosis. Vibration-based CNN models have also been explored, including multi-

head 1D-CNN architectures for detecting multiple fault types using multi-directional accelerometer data [341]. Multi-input CNNs fusing vibration and acoustic features derived from time-frequency spectra have been proposed for induction motor diagnosis [342]. Attention-enhanced CNNs operating on continuous wavelet transform representations of vibration signals further improve discriminative capability under varying operating conditions [340]. Beyond standard CNNs, hybrid and advanced architectures have been investigated. Hierarchical CNN-SVM models have been used to jointly detect incipient faults and estimate fault severity from current signals [338]. Hybrid RNN-CNN with attention mechanisms have been proved effective to diagnose induction motor faults directly from raw voltage and current signals [345]. Graph-based deep learning has also emerged, which combines signal-to-image transformation with superpixel-based graph construction and graph attention networks to achieve robust induction motor fault diagnosis under steady and transient operating regimes [343].

2) *Process faults*: Process fault diagnosis concerns the monitoring of complex industrial processes in which failures do not originate from a single component, but emerge from deviations in system-level behavior. Typical application scenarios include chemical production units, semiconductor manufacturing lines, metallurgical plants, and large-scale continuous processes.

In chemical and process industries, faults often manifest as slow drifts in temperature, flow, or level measurements, abnormal reaction dynamics, or inconsistencies between correlated variables. These behaviors are difficult to detect with local thresholds and motivate predictive monitoring strategies. For example, multi-step fault prediction has been investigated in chemical processes subject to false-data injection and slow-developing faults, where temporal models are used to anticipate abnormal evolution before operational limits are violated [162].

In semiconductor manufacturing, an applicative example is the detection of abnormal wafer produced multi-stage fabrication processes such as thin-film chemical vapor deposition. Here, faults appear as unexpected changes in sensor readings, deviations in processing time steps, or abnormal temporal patterns across recipe phases. Data-driven fault diagnosis methods are applied directly to real fab data to flag defective wafers before downstream processing, reducing yield loss and rework costs [154]. Related studies on benchmark semiconductor datasets confirm that ensemble-based classifiers remain effective for this application, even when synthetic data augmentation is required to compensate for the rarity of faulty samples [346]. CNNs are also used, for example in [347], where an approach to detect faulty wafers is implemented via a multiple time-series CNN using internal equipment sensor variables (SVIDs), and in [282], which uses a CNN-autoencoder. Since these last two studies use real data and real faults, they are better described at the end of this subsection.

In heavy industries such as iron-making, process fault diagnosis is applied to large-scale systems like blast furnaces, where faults emerge under strongly nonlinear and non-stationary operating conditions. The application challenge here

lies in separating fault-related behavior from normal process variability caused by changing raw materials, loads, and operating regimes. Hybrid pipelines that remove non-stationary components before classification have been shown to improve fault detection reliability when applied to real blast furnace data [348]. In metal processing industries, FDD is also applied to rolling mills, where faults directly affect product thickness, surface quality and throughput. The paper [141] diagnoses faults in the Finishing Mill Area using a modified recurrent unit, using as input process variables and KPIs.

Accounting for interactions between sensor and process units is fundamental in industrial process FDD. In large interconnected facilities, such as three-phase flow installations, power systems and nuclear plants, faults propagate across sub-systems. Graph-based modeling approaches have been adopted to explicitly represent dependencies among components [213], [214].

Interestingly, almost all the reported studies in this section use real world data collected under controlled laboratory conditions for dedicated test rigs, where faults are intentionally introduced and operating conditions carefully regulated. While this is fundamental to validate and benchmark models, it only partially reflects the complexity of real industrial environments. Nevertheless, there are two recent real-case studies that use actual operational data and naturally occurring faults for process FDD. The first, [144], addresses the problem of alarm floods in industrial plants, situations where a large number of alarm trigger simultaneously, overwhelming operators and delaying critical responses. Using real operational data from an Algerian gas plant (Hassi R'Mel), the authors develop and compare several supervised machine-learning models to classify alarms and predict whether a new alarm is true or false. Among all tested algorithms, Random Forest achieves the best performance, reaching the highest precision, recall, and F1-score, making it the most reliable model for industrial alarm management. Additionally, the study in [282] is based on real multivariate time-series data from an 8-inch semiconductor ashing tool, including genuine failure events associated with actual part replacements. Using this industrial dataset, the authors construct a framework that transforms sensor streams into correlation-based signature matrices and detects anomalies with a convolutional autoencoder. Residuals from the autoencoder are then used to classify which equipment part failed, and SHAP analysis identifies the sensor-pair correlations driving each prediction. The approach demonstrates strong performance in both failure detection and root-cause identification under real manufacturing conditions. Always in the semiconductor domain, [347] use real wafer-level data from a chemical vapor deposition process. The empirical study is based on actual production wafers exhibiting real process faults.

3) *Cyber attacks*: Cyber-attack detection for IIoT is a major application domain that can be framed within FDD. The detection is typically addressed through intrusion detection systems that monitor network traffic and system behavior to identify abnormal patterns. The literature considers a broad spectrum of attack types, including malware, denial-of-service, man-in-the-middle, phishing, supply-chain attacks, botnets and

false data injection [349].

The primary data sources are network- and system-level measurements, such as packet-level features, flow statistics, protocol attributes and SCADA communication variables, collected either from real IIoT deployments or realistic testbeds. These data are often high-dimensional and highly imbalanced.

While approaches based on traditional ML (e.g., on random forest and tree-based ensembles) still appear in the literature [350], [147], [145], state-of-the-art performance is consistently achieved by DL models. Recurrent architectures, including GRU, LSTM, Bi-LSTM, are widely adopted to capture temporal dependencies [351], [352]. Feedforward NNs and CNNs are also employed [353], [354]. Hybrid architectures that jointly model spatial and temporal patterns, such as CNN-GRU combinations, have proven especially effective for detecting complex attacks like botnets [349]. In [162], the authors propose a model that integrates a modified LSTM with multi-lookback inputs to enable multi-step autoregressive prediction, and couples it with a 1D-CNN, enabling the detection of false data injection attacks on networked industrial control systems. Hybrid and ensemble models are frequently explored to improve robustness and coverage of diverse attack patterns. Examples include multi-modal deep ensembles models, combining RNN, Bi-LSTM and Deep Belief Network (DBN) for cyber-physical systems [355], as well as architectures that integrate deep feature extractors with lightweight classifiers. A representative approach is the two-layer denoising autoencoder framework in [356], where successive layers suppress neutral features and extract discriminative representations, which are then classified using an SVM. For SCADA-based IIoT networks, hybrid models combining advanced recurrent units with decision-tree aggregation have been proposed [357].

TL is sometimes adopted to mitigate the scarcity of labeled attack data and to improve generalization, as demonstrated by the ResNet-based transfer-learning intrusion detector in [353]. Data imbalance is typically addressed through augmentation strategies such as SMOTE [351], while there exists also recent works that explore generative diffusion models to synthesize realistic attack traffic [352]. Since modern attacks often affect multiple system layers, while most existing approaches rely on single-modality time-series data, the design of robust multi-modal models and effective fusion strategies remains an open challenge. This issue is further compounded by common limitations such as the scarcity of publicly available datasets, limited explainability of AI-based methods, increasing system complexity and scalability requirements and the need for privacy-preserving solutions [358].

4) *Human-machine interaction*: Human-machine collaboration is a pillar of Industry 5.0. Although human-related errors cannot be strictly classified as faults in the traditional sense, they can still be framed within the FDD as deviations from expected operational behavior. Consequently, recent works investigate data-driven methods to detect human-related anomalies, primarily relying on visual input such as images or video streams. To give some examples, [359] introduces an end-to-end spatio-temporal learning framework for real-time monitoring of assembly operations. The method predicts future human actions from video using a convolu-

tional encoder and ConvLSTM-based sequence modeling and detects human faults such as unfixed parts, incorrect assembly order and missing components, by comparing predicted and observed action sequences. In [360], a two-stage deep learning pipeline is introduced to identify unsafe ergonomic behaviors. Monocular video is first used to reconstruct 3D human body poses via the VIBE architecture; the resulting decimated body meshes are then classified using MeshCNN to recognize ergonomically hazardous actions across diverse industrial tasks. Finally, [361] presents a multimodal framework for safety monitoring in human-robot collaboration. The approach combines scene-graph-based visual understanding with a BERT-Bi-LSTM-conditional random field model to interpret scene descriptions and verify compliance with robot-specific personal protective equipment (PPE) requirements, enabling the detection of missing or improperly used PPE.

5) *HVAC faults*: HVAC systems are important in manufacturing plants for maintaining thermal and humidity conditions critical to processes such as semiconductor fabrication, precision machining, pharmaceuticals and clean-room-assembly. Faults can affect both energy efficiency and process reliability, with typical issues including sensor drift, heat exchanger fouling, refrigerant leaks, valve or damper malfunctions, chiller degradation and air-handling-unit (AHU) control faults. HVAC FDD relies on operational data (temperatures, pressures, flows, power and control signals). A key challenge is separating true fault symptoms from variations caused by production schedules or environmental conditions. Causal discovery and inference methods have been shown to identify real fault drivers and build interpretable models with high accuracy and low computational cost [362]. Chiller faults are particularly complex due to thermal coupling and compound failure modes. Knowledge-embedded models that integrate expert rules into data-driven networks (e.g., deep belief networks) improve generalization and enable detection of multi-fault scenarios, validated on real experimental data and deployed in cloud-based monitoring platforms [363]. Data scarcity is another practical challenge: some studies use physically induced faults under controlled conditions to select informative features for chiller diagnostics [158], while others generate synthetic fault samples with GANs and reinforcement learning to enhance AHU fault detection under extreme data limitations [268]. For a comprehensive overview of methods and applications, see [364].

IV. PREDICTIVE MAINTENANCE

This section is devoted to introducing the field of PdM which differs from the previously presented AD (Section II) and FDD (Section III) because of its prognostic nature. In fact, PdM is a component of the broader field of Prognostics and Health Management (PHM) which encompasses the combination of several technologies including advanced monitoring, diagnostics, prognostics and prognostics management. These are characterized by the common goal of minimizing the occurrences of failures or downtime in industrial manufacturing processes [367].

TABLE VIII
REVIEW OF INDUSTRIAL APPLICATIONS OF ML-BASED FDD

Fault Type	Category	References
Mechanical/ Equipment	Rotating machinery (bearings, shafts, gears, imbalance).	ML: [143], [148], [152], [153], [150], [157], [151], [146], DL: [271], [272], [135], [365], [172], [175], [176], [178], [177], [169], [171], [163], [180], [164], [185], [181], [168], [183], [182], [186], [197], [193], [202], [195], [223], [258], Hybrid: [160], [159], [366], [184], [191].
	Tool wear and breakage (cutting tools, milling, drilling).	ML: [319], [149], [320], DL: [326], [321], [322], [323], [325].
	Hydraulic and pneumatic.	ML: [328], [329], DL: [187], [330], [331], [332], [333], [244], [249], [334].
	Induction motor, electrical drive faults (stator winding, rotor bars).	ML: [344], [335], [148], [339], DL: [179], [170], [337], [341], [342], [340], [338], [343], Hybrid: [345].
Process	Three-phase flow facility.	DL: [213], [317], [214].
	Chemical plants.	DL: [162].
	Nuclear power systems.	DL: [214].
	Metallurgic process.	DL: [348], [141].
	Semiconductor manufacturing.	ML: [154], DL: [347], [282].
	Gas plant.	ML: [144].
Cyber attacks	Product quality.	LLM: [229].
	Intrusion, denial-of-service, man-in-the-middle, supply-chain attacks, botnets and false data injection.	ML: [147], [145], [350], DL: [351], [352], [353], [354], [349], [162], [355], [356], [357], [353], [351], [352].
Human-machine interaction	Operator error/unsafe action detection.	DL: [359], [360], [361].
HVAC	HVAC faults.	ML: [362], [158], DL: [363], RL: [268].

A. PdM Theoretical Foundations

In this section we start by reporting the definitions of some basics concepts in the field of PHM, PdM and RUL estimation which will be used throughout the rest of the survey.

Definition 3. Predictive Maintenance (also referred to as **Condition-based Maintenance**) is an advanced maintenance strategy which employs data analytics, ML and IoT technologies to predict equipment failure before their occurrence with the aim of reducing maintenance costs and minimize unexpected breaks and downtime in industrial applications.

The pipeline of maintenance policies adopted in the industrial field is usually divided into five main steps [368]:

- **Remote Monitoring:** This step involves monitoring the equipment under analysis by embedding sensors or cameras to enable real-time data collection regarding its health state. The acquired data may then be transmitted on site or to external devices through cloud infrastructures. Nowadays, this step is handled by IIoT infrastructures which are becoming more and more advanced and able to handle huge amounts of data produced by sensors, more details of which will be presented in IV-D3. Monitoring data represents the main resource for the predictive models used in later stages of the pipeline. Some pre-processing steps (i.e. data cleaning, data normalization and handling of missing or corrupted data) are traditionally executed before feeding the data to the models for the prediction.
- **Failure Prediction:** Pre-processed data are analysed through ML or DL based models to produce predictions. In this stage we may be interested both in prognostics predictions (i.e. when the asset is most likely to fail) or in diagnostic ones (i.e. identify the most vulnerable

components of the system). This is achieved through RUL or AD estimation, respectively. This highlights the strong interconnection between these two fields.

- **Task Scheduling:** Based on the predictions produced by prognostic systems and the domain knowledge of experts, maintenance operations are scheduled in order to prevent the equipment from failing. In order to solve this complex optimization problem Reinforcement Learning based approaches have recently been proposed, more details of which will be provided in IV-B4.
- **Communication and Integration:** To translate the decisions made in the previous steps into actionable measures, communicating with maintenance personnel is crucial. This step is traditionally carried out through the usage of Enterprise Resource Planning (ERP) and Asset Management software tools.
- **User Interface and Reporting:** The User Interface should guarantee a clear understanding of the model decision to the machine operators. This is normally carried out through interactive dashboards, reports and alerts. In this context the trustworthiness of the predictions produced by the predictive models, achieved through XAI (presented in IV-C), is also a required guarantee.

PdM is employed in the second stage of the maintenance pipeline (i.e. *Failure Prediction*) with the aim of outperforming two other traditional prognostic approaches, typically referred to as **Reactive Maintenance** and **Preventive Maintenance**.

Definition 4. Maintenance interventions based on **Reactive Maintenance** (Run-to-Failure) wait for the machine failure to be performed. While this strategy fully exploits a machine's operational lifespan, it incurs significant repair costs and risks, such as cascading failures (i.e. subsequent failures in dependent components or interconnected systems). As the

name suggests, this strategy represents a reactive approach with no way of controlling or monitoring the system's health state during its lifespan.

Definition 5. Preventive Maintenance occurs at fixed time intervals, decided based on historical failure times of the machine. Although more systematic and proactive than Run-to-Failure approaches, this strategy often results in premature maintenance cycles, underutilization of component lifespan, inefficient resource allocation, and excessive spare part consumption.

To address the downsides of both the reactive and preventive maintenance approaches, in PdM maintenance is dynamically scheduled based on predicted time-to-failure estimates, enabling near-optimal utilization of component lifespan while mitigating failure risks. Differently from Preventive Maintenance, where the time interval between two consecutive maintenance interventions is decided a priori based on the time-to-failure experienced in the past, in PdM the maintenance time is determined based on the current condition of the machine which is monitored in real-time.

In order to ensure a continuous monitoring of the equipment's health state sensor networks, often incorporating signal processing techniques, are employed. Several benefits arise from this approach including the prevention of catastrophic failures, identification of undetected failure modes, minimization of downtime, reduction of maintenance costs, optimized maintenance scheduling and also valuable insights for equipment design improvements.

This survey will mainly focus on the RUL estimation task, an alternative formulation of the PdM task, defined as follows:

Definition 6. Remaining Useful Life is defined as the amount of time remaining in the productive life of a manufacturing equipment. The RUL is typically represented as a time varying signal characterized by a monotonically decreasing trend. It can be expressed in terms of number of cycles, seconds, minutes or hours, depending on the specific application.

PdM gained popularity thanks to the rapid development of IoT-based systems and it's one of the main components of Industry 4.0. In fact, sensors composing IIoT infrastructures represent a fast and reliable approach to collect data (i.e. vibration, temperature, humidity) strictly related to the equipment's health which are used to design and train the predictive models used for the estimation of the RUL signal.

In particular, the huge amount of data collectable through IoT sensors is the main source of information for ML and DL based algorithms. Differently from older approaches where the machine's life degradation is modelled exactly through complex physical or mathematical equations, these approach base their predictions just on the relationship found between the input sensor readings and the RUL signal. This provides a more computationally efficient method to solve this complex task.

This is the main feature of data-driven PdM, introduced in IV-B2, which will be the main focus of the PdM survey.

The application of PdM technologies has brought several benefits in industry. According to [369], the integration of

PdM strategies in manufacturing plants led to a 20% reduction in unplanned downtime and decrease in maintenance cost of about 15%.

By predicting the RUL the objective is to perform maintenance interventions at the right moment: not too late (as it happens in reactive maintenance) or too early as in preventive maintenance. In this way both costly machine repairment and long downtime intervals are avoided.

B. PdM Methods

This section is devoted to the presentation of the current state of the art in the field of PdM. The operational efficacy of PdM systems fundamentally depends on accurate RUL estimation, considered as the most complex yet most beneficial PHM task [370]. This problem serves as the foundation for maintenance decision-making processes and will be the main focus of this section.

PdM has become the predominant strategy in the field of prognostics and health monitoring in industrial applications in recent years [371], [372]. PdM is applied in several industrial fields, such as: turbofan engines [373], wind turbines [374], rotating machinery [375], high-speed trains [376], aviation systems [377], railway infrastructures [378], [379], [380], pharmaceuticals production [381], lithium-ion batteries [382], vehicle lifts [383] and also operating systems [384]. Moreover PdM can also be integrated as one of the services offered by DT technologies [371], [385], predominant in the field of Industry 4.0 and 5.0.

The RUL estimation research field is very broad and comprises several different industrial domains which will be covered in IV-B, IV-D. For the sake of clarity the different branches of RUL-based PdM are graphically represented in Figure 1.

Three main macro areas of PdM estimation can be extrapolated from 1: *model-based*, *data-driven* and *hybrid* RUL estimation whose main characteristics and differences are highlighted in IV-B1, IV-B2 and IV-B3, respectively.

The methods presented in this section and in IV-D are grouped in Tables IX and X divided in hybrid and data-driven methods. The few model-based approaches presented in IV-B1 are not considered in this comparison because they are not the main target of the survey which focuses on the usage of AI-enhanced models for the automation of prognostic decisions.

1) *Model based RUL Estimation*: Model based methodologies for RUL estimation exploit prior physical and mechanical knowledge on the equipment with the aim to design a physical model of the degradation process which represents how the machine transitions from an initial state of full health to the end of its life (i.e. machine breakage). The main advantage of these methods is that if an accurate model of the machine is built it will fully characterize the degradation process and provide accurate estimates of the RUL. However crafting these kind of models is highly challenging because of the complexity of the physical and mechanical processes typically involved in real world equipments. Physical processes are in fact well understood just for critical or very simple components [490], [491], [492], [493].

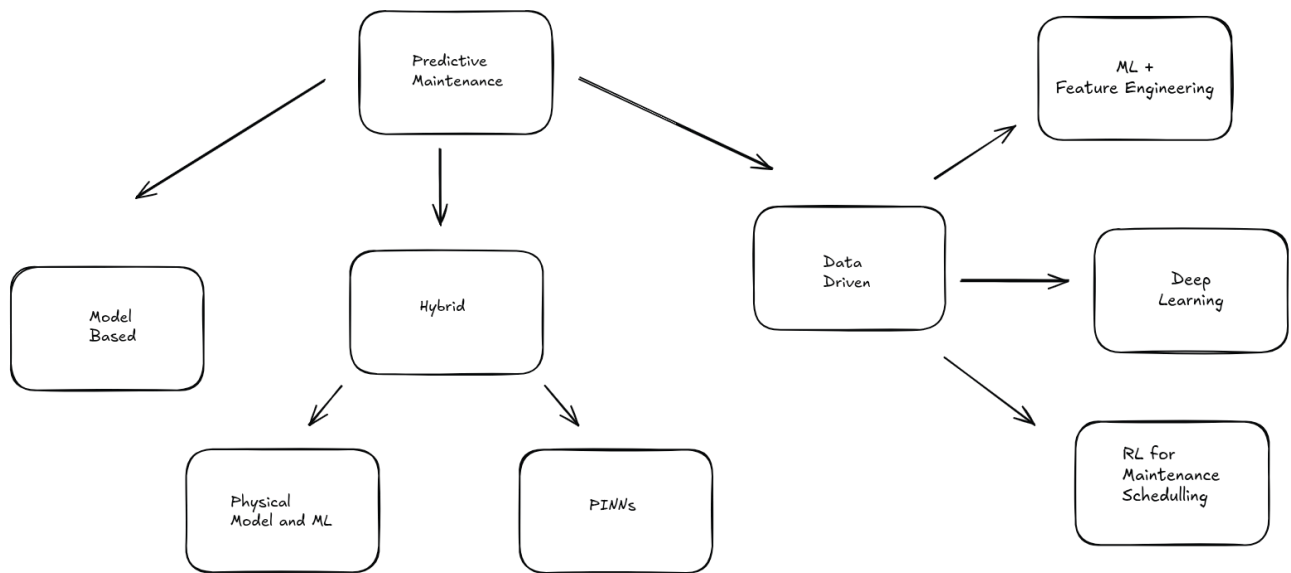


Fig. 1. The scheme above subdivides the topic of PdM into different sub topics which will be covered in the next sections

TABLE IX
HYBRID PDM APPROACHES

Category	Techniques	References
Hybrid PdM Approaches	RUL estimation approaches where both model-based and data-driven approaches are used	[386], [387], [388], [389], [390]
PINN	PINN models embed physical principles as a regularization term in the loss function	[391], [392], [393], [394], [395]

Additionally these models are usually equipment specific and may not be easily transferable to completely different or similar machines working with different operating or manufacturing conditions. For these reasons, the employment of these methodologies is becoming less and less common in the literature in favor of the more flexible and general purpose data-driven methodologies.

2) *Data driven RUL Estimation:* Data driven methods do not require domain knowledge of the system under analysis but will instead rely on the availability of historical data, usually in the form of raw sensor measurements, acquired over the entire life cycle of the machine. These methods raised in popularity thanks to the developments of AI related technologies (i.e. ML and DL) and IoT networks for data collection. These models are capable of estimating the degradation process of the machine directly from the historical data and, based on that, will provide estimates of the RUL.

Once enough historical data are available, accurate RUL predictions can be obtained without the need of designing overly complicated physical models. In particular, models are usually designed in order to easily generalize to assets similar to the ones producing the training data. AI powered models may be adapted to work on different equipments through the employment of transfer learning [494], [12] or domain adaptation [445] techniques. In general, the machine degradation processes estimated by such models are less accurate with respect to the one produced by a model based approach. In

fact, the outcomes generated by these models are not based on a physical model of the system but rather on statistical patterns identified in the historical data. However, even if not perfect, these estimates are in general good enough to trigger maintenance operations at the right time and come at a significantly lower cost in terms of computational and human resources. As a result, nowadays data driven models are overtaking model based approaches in the industrial field [402]. For this reason, data driven models will be the main focus of this review.

With the rapid development of data-driven prognostics and the high amount of data required the issue of data privacy arises. As a consequence in the literature this challenge is addressed through Federated Learning (FL), a paradigm in which the privacy of user's data is ensured by the fact that each user trains the model locally with private data and shares with other users just the model's updates. Examples of FL based RUL estimation are [495], [496], [479].

RUL Estimation using Feature Extraction: The main source of information for data driven models employed in the task of RUL estimation are the raw data produced by sensors monitoring key physical quantities of the equipment under analysis (i.e. temperature, flow, pressure, acceleration, humidity, ...). These signals are typically distorted by noise and may interfere with other signals. This means that the signs of the machine's useful life degradation, which are the key component for a precise estimate of the RUL, are often not directly observable in the raw data. Consequently, some

TABLE X
DATA-DRIVEN PDM APPROACHES

Category	Techniques	References
Data Driven PDM Approaches	Generic RUL estimation approaches where the RUL signal is estimated through data (i.e. sensor readings) rather than with the usage of complex physical model of the equipment	[396], [397], [398], [399], [400], [401], [402], [140], [403], [404], [405], [406], [407], [405], [408], [409], [410], [411], [412], [413], [414], [415], [416], [417]
Feature Extraction + ML	These methods exploit ML algorithms for the RUL estimation, thus an initial feature extraction step is required to handcraft the features needed by the model to accurately estimate the equipment's end of life.	[396], [397], [398], [399], [400], [401], [402], [140], [403], [404], [405], [406], [407], [405], [408], [409], [410], [411], [412], [413], [414], [415], [416], [417]
Lithium-ion batteries	RUL estimation is a very important task in the context of lithium-ion batteries whose degradation process can lead to catastrophic consequences as unpredictable burn of electric vehicles.	[418], [419], [406], [396], [391], [382], [420], [421], [422], [423]
Turbofan engines C-MAPSS	The C-MAPSS dataset is the most commonly used benchmark dataset in the context of RUL estimation. The sequential nature of the data promotes the usage of sequential models such as LSTM, CNN and combinations of those.	[370], [424], [425], [426], [427], [428], [429], [430], [431], [432], [433], [434], [399], [435], [436], [437], [438], [439], [440]
Turbofan engines N-CMAPSS Bearings	N-CMAPSS is a more complex and recent version of C-MAPSS. The approaches adopted on this benchmark are similar to the ones used on its predecessor. Rolling element bearings are one of the most common reasons of machine breakdown in the industry. The periodical nature of the data enables the definition of notable fault-frequencies that facilitate the detection of faults in the context of FDD and inherent interpretability in PdM.	[441], [12], [442], [443], [444], [433], [445]
Transformer	Attention based models are widely used in the field of NLP but have also extensive applications in time series modelling. In particular, in the context of RUL prediction the inclusion of attention modules in the model architecture facilitates comprehension over the root causes of the machine degradation.	[446], [403], [447], [448], [374], [400], [449]
Uncertainty	UQ is a fundamental requirement in PdM considering the elevated economical costs associated with underestimation or overestimation of the target RUL signal. The most common approaches are based on BNN and Quantile Regression.	[450], [451], [452], [433], [453], [444], [454], [455], [445], [456]
XAI	UQ is a fundamental requirement in PdM considering the elevated economical costs associated with underestimation or overestimation of the target RUL signal. The most common approaches are based on BNN and Quantile Regression.	[457], [395], [458], [459], [460], [461], [462], [463], [441], [464], [465], [466], [467]
IoT	In the context of the fifth industrial revolution, where the role of the human being is gaining more and more relevance, being able to provide insights on the model decisions is of paramount importance. Traditionally XAI approaches exploit variants of the SHAP explanation algorithm, which may also be included on the AD module of a complete PdM pipeline that performs both diagnostic and prognostic activities.	[468], [469], [470], [433], [471], [446], [472], [473], [474], [475]
Industrial Case Studies	IoT infrastructures are one of the key elements of the latest industrial revolution and their interconnection with health monitoring and AI is considered in order to enhance performance of PdM based applications. In these settings usually cloud computing infrastructures are employed to better store and manage the data collected from IoT sensors which are later used by ML or DL algorithms to perform end of life predictions.	[476], [477], [478], [479], [376], [480], [481]
	In this category several case studies of PdM applications in the context of Industry 4.0 and 5.0 are presented. One of the adaptations employed to embed RUL prediction models in high-stakes industrial contexts is the usage of business metrics that shape the model's predictions to account for the business costs associated with imprecise estimations, preferring a pessimistic approach rather than an optimistic one.	[6], [482], [483], [484], [485], [486], [487], [488], [489]

pre processing steps are required before feeding the sensor measurement to the ML models. These involve the creation of handcrafted features, extracted through transformations directly applied on the raw data. The resulting variables are often referred to as Health Indexes (HI) or State of Health (SOH) [396] since they provide a complete representation of the health state of the monitored equipment throughout its entire lifespan. This pre processing step is often denominated as **Feature Engineering**.

Industrial equipments are characterized by a monotonic deterioration of their properties (i.e. without sudden regenerations). In order to effectively capture their health condition over time, features showing a monotonic trend should be extracted [397]. Coble et al. defined in [398] a set of measures to score the importance of a given feature as a prognostic parameter: *Monotonicity*, *Prognosability* and *Trendability*.

These are employed in [399], [400] where they are merged together into a weighted sum to measure the relevance of sensor measurements for the RUL estimation task, and in [401] where *Monotonicity* is used as a feature selection criteria. An interesting study is the one reported in [497] where SHAP values are computed on models employed on the C-MAPSS dataset showing an high correlation with these three prognostics measures.

Typical features extracted in this stage are presented in III-A2.

In addition to the construction of handcrafted features other typically used methods are: dimensionality reduction (i.e. PCA) ([443], [403], [404]), EMD [396], clustering to discover additional features, noise filtering (i.e. Gaussian, Kalman or Exponential filters) or even physics-inspired feature transformations [406].

The extracted features or HI are successively used as inputs to the ML models: Regression models trained to directly predict the RUL values. An alternative approach consists in defining a threshold on the values of the HI to trigger maintenance operations [403], [498] once that value is reached.

There are also unsupervised methods of HI estimation used where no predefined degradation process information on the equipment are given but are automatically extracted through signal preprocessing or signal entropy [407]. Also DL models can be exploited to extract hidden features from sensor measurements, in particular AE and VAE based architectures [403], [405], [408], [409].

Several works in the literature exploit HI for their RUL prediction using different strategies to extract the equipment State of Health (SOH): graph modelling [410], weighted correlation of known fault frequencies over cycles in bearings [411], a quasi-orthogonal sparse projection algorithm [412] to trace different degradation trends for different fault types, information entropy [413], frequency energy shift exploiting the signal produced by a microphone sensor [414], CWT [415] and Gaussian Process (GP) Regression for SOH estimation [416].

Another example of the usage of feature extraction for RUL estimation is represented by sensor fusion based approaches [404], [417].

In recent times Feature Engineering based methods became less popular because of the development of DL algorithms. Deep models are known for stacking multiple hidden layers sequentially building a hierarchical representation of the features embedded in the data. This makes it possible to directly extract key degradation features from raw data without the need of an handcrafting feature extraction process as the ones explained above.

Moreover relying on handcrafted features makes it challenging to adapt to different prognostic applications [435]. However, the combination of feature engineering with ML based prediction model can still be considered as a valid alternative in case end-to-end DL solutions become too computationally complex for the implementation on a real-world IoT based industrial system.

A peculiar case is represented by [482], [453], [443] where handcrafted features are combined to raw data using feature fusion to enrich the training data used to fit a deep learning model combining feature engineering with the most recent DL approaches.

RUL Estimation for Turbofan Engines: As mentioned in paragraph I-A the C-MAPSS dataset is very commonly used to evaluate the performances of RUL estimation methods, as highlighted in [370], [424]. The main challenge posed by this dataset are the different operating conditions and degradation patterns of the turbofan engines due to manufacturing variations. In order to solve these issues the majority of the works in the literature employs DL algorithms. In particular, sequence learning DL architectures are employed [425], [426], [427], [428], [429]. A common example is the combination of the LSTM [430], [431], [432], [433] and CNN [499] architectures. In fact, CNNs are good in merging different sensor dimensions such as vibration and acoustic data. On the other hand, LSTM networks are specialized in capturing sequential dependencies

in time-series data. As a consequence, the combination of these two architectures is usually the go-to solution for a task like RUL estimation where input data are multi-sensor time-series in the majority of the cases. In some cases the addition of an attention mechanisms, to weight the importance of different sensors and time stamps, is also considered a valid solution.

In [434] and [399] the machine degradation process is not assumed to start from the beginning of the production phase but after the first incipient fault. Before this moment the machine is considered to be in a full health state, thus the RUL signal is constant at the initial value and the degradation towards failure begins only after the incipient fault. This model for RUL degradation is typically named **piecewise linear degradation** model while the simpler approach of considering the RUL as a monotonically decreasing signal is called **linear degradation** model. The piecewise linear degradation model represents a more realistic scenario but adds an additional step in the estimation pipeline: the detection of the point in which the machine transitions from a full health to a degradation state. This is achieved in [434] by using a pair of CNNs architectures: the first one to estimate the incipient fault and the second one devoted to the RUL estimation. Moreover, a reliability index is also used in order to distinguish the RUL of two different components. Another application of the piecewise linear degradation model is presented in [399] where the so called *elbow point* (i.e. the point when degradation is first observable) is detected through a Z-test and then only sensor data from the *elbow point* onwards are employed for prognostics, which is realized through an Echo State Network (ESN) optimized through a Differential Evolution (DE) optimization algorithm.

In [435] the capacity of DBN to extract hierarchical features is employed as an alternative to the commonly used feature engineering combined with ML models approaches. To increase model efficiency an ensemble of DBN is considered and it is trained through a multi-objective evolutionary algorithm needed to tune the optimal hyperparameters of the base learners (i.e. number of hidden neurons in each hidden layer). Another example of ensemble learning is employed in [436] where Stacking Ensemble Learning is used to exploit the predictions of different base learners (i.e. RF, Decision Tree, Gradient Boosting) and combine them through a meta-learner (i.e. Linear Regression).

To take into account for the different working conditions characterizing the C-MAPSS dataset a Mixture of Encoders based model is presented in [437] in order to specialize and Autoencoder model on the specific features of each working condition. Finally, features extracted in this way are fed in input to a Transformer to obtain the final RUL prediction.

Together with CNNs and LSTM based architectures another effective approach to learn the spatio-temporal relationships underlying the RUL prediction is to employ TCN. TCN is an optimized version of 1DCNN to model both long and short temporal dependencies through the usage of dilated convolutions, causal convolutions (i.e. the output of layer t depends solely on the convolutional outputs at layers $t-1, t-2, \dots$) and residual connections. In [438] TCN layers are used together with LSTM layers and a data augmentation strategy

is employed to enrich the training and testing trajectories contained in C-MAPSS. Other works employing TCN for the feature extraction part are [439], [440].

Another commonly used dataset to evaluate PdM models in the field of turbofan engines is the N-CMAPSS dataset and several works have used it as the benchmark dataset: [441], [12], [442], [443], [444], [433], [445].

RUL Estimation employing Transformer based architectures: The Transformer architecture, introduced in [500], has had a lot of success in NLP but several applications and adaptations of attention-based models on time series data have also been proposed, as an alternative to the classic CNN and LSTM architectures, presented in IV-B2. These models employ the attention mechanism to discover correlations between different time stamps in order to identify the most important ones. Some of these Transformer's variants have also been used in prognostics: [450] uses a Temporal Fusion Transformer (TFT) [451] and [452] a Temporal Flow Transformer which employs ProbSparse self-attention, a modification of the traditional self-attention mechanism tailored for time series processing, introduced in [501].

The employment of the attention mechanism, which assigns weights to the time stamps composing the input sequence based on their importance, provides also some explainability since it focuses on the most important features composing the input sequence [433], [453].

It also helps in the feature extraction process by identifying the most relevant spatio-temporal features for the task. As an example in [373] a Dual-Attention mechanism is used to enhance RUL prediction: the combination of CNN with a Self Attention layer is used to extract the most relevant spatial features while the temporal ones are retrieved through the combination between a LSTM and a Self Attention layer. In [444] a dual channel is used to separately treat sensor data and time-varying operating conditions which are firstly modelled through a 1DCNN to extract spatial features that successively are fed into two parallel LSTM layers to generate temporal features which are passed through a soft attention layer to filter out invalid information. Finally, the RUL prediction is produced through a Regression Head represented by two Fully Connected (FC) layers.

Attention based feature extraction is also employed in [454] together with a gated convolutional unit specialized in discovering local features. The combination of CNNs and Transformer architectures is also used in [455] where a Convolutional Transformer model is proposed.

In [445] the focus is not solely on the RUL estimation task but also on the Domain Adaptation one which has a lot of beneficial aspects in prognostics: for example, it can achieve transferability of RUL prediction capabilities across different machines. The main contribution brought by this work is the focus on the adaptation not only at the feature level but also at the semantic one. In particular, an adversarial approach is employed for the Domain Adaptation task while the Transformer Encoder is used as a feature extractor. In order to consider both feature-level and semantic-level adaptation two discriminator networks are used: feature-level source identification is obtained from the extracted features

while semantic-level identification is done based on the output RUL curves. Essentially the semantic discriminator acts as a regularization term during training, punishing inaccurate and unreasonable outputs that can be produced in the target domain if Domain Adaptation is only applied at the feature level.

The Transformer architecture was originally introduced to solve the machine translation task in the field of NLP and was later applied to other common NLP tasks such as sentence and emotion classification with the Deep Bidirectional Transformers for Language Understanding (BERT) [502] model. In [456] this model is adapted to bearing RUL estimation. Time features are extracted from the raw vibration signals and passed as features blocks to the BERT input layers. The training strategy involves a pre-training phase on a large dataset in order to equip the model with a general knowledge on the RUL estimation task and then fine-tuning is used to specialize the model on a specific scenario.

Uncertainty Quantification in RUL estimation: One of the main advantages of PdM is the reduction of downtime and machine breakages costs, guaranteed by a precise estimation of an equipment's health state and RUL. In this context it is of paramount importance to take into account the uncertainty in the model predictions which can cause an *unexpected break* if too optimistic (i.e. RUL overestimation) or *unexploited lifetime* conditions if too cautious (i.e. RUL underestimation). This estimate is successively used to guide maintenance operators in interventions making uncertainty quantification equally important as prediction accuracy [395]. From an economical perspective designing a more pessimistic model is the preferable choice, although in specific cases when an intense use of the machine is required (e.g. in agriculture where the time to collect crops is limited and highly subjected to weather conditions) an aggressive use of the machine may instead be considered. For this reason, another crucial aspect to take into account is not only the point estimation of the RUL value but also the uncertainty associated with that. UQ is a very important task to produce trustworthy models and it is widely adopted in DL [458] and, consequently, in PdM and RUL [457] estimation.

A natural solution to introduce uncertainty estimation in RUL prediction, or in any other regression task, is the usage of BNN, a class of NN with weights modelled by a probability distribution that naturally provide confidence intervals in their estimates. This approach is applied to different data-driven architectures: DBN [435], traditional NN [459], RNN [460], GRU [390], PINN [395] and CNN [461]. Traditionally training BNN is achieved through Bayesian Variational Inference [459] or Markov Chain Monte Carlo (MCMC) derived methods such as Hamiltonian Monte Carlo (HMC), employed in [463]. [462] applies BNN to RUL prediction and uses the Stein Variational Gradient method for the efficient training of such models. Moreover in [462], [463] the the RUL prediction is enhanced by its uncertainty which works as a corrective factor in order to reduce it in case a too high value is produced. Moreover BNN models are inherently regularized and this provides a natural improvement of the predictive accuracy with the respect to their non-bayesian counterparts. [441], instead, introduces a Bayesian Gated-Transformer including a

customized encoder block with gated connections to improve feature extraction capabilities and both epistemic and aleatoric uncertainty provided by the probabilistic nature of the model weights and the noise channel, respectively.

In [464] a GNN model is used to perform RUL prediction on a complex mechanical equipment. In particular, GCN and GAT modules are used for spatiotemporal feature extraction which is succeeded by a Bidirectional Long Short Term Memory (Bi-LSTM) module to also produce temporal features. Finally Maximum Likelihood Estimation (MLE) is incorporated in the loss function to model the output space as a Gaussian distribution and enrich the model with an estimation of the uncertainty on the RUL prediction, which is a fundamental ingredient in prognostics to improve system reliability and operational efficiency.

Another approach for uncertainty estimation is Quantile Regression where different quantiles of the posterior predictive distribution are estimated. Applications of Quantile Regression in the field of RUL estimation can be found in [465], [466]. In [465] a State Space Model (SSM) architecture is proposed for RUL prediction and it is enhanced by the Simultaneous Quantile Regression (SQR) [503] framework for both training and evaluation. On the other hand, [466] uses eXtreme Gradient Boosting (XGBoost), a non DL architecture, for feature selection and extraction and Multi-Quantile Recursive Neural Networks (MQ-RNN) [467] for a Quantile Regression based RUL estimation. The advantage of both SQR and MQ-RNN is that, being used for multi-quantile regression there are no prior requirements on the output distribution.

3) *Hybrid Approaches for RUL Estimation:* Data-driven models are nowadays the standard for RUL estimation but they still have some limitations. First of all, in the field of PdM it's difficult to collect high-quality data because of the rarity of faults and of the unwillingness to reproduce faults on purpose just for the sake of training such models, as explained in section I-A. Additionally, environmental conditions may have a huge impact on the physics of the system and it's difficult to collect data representing all the possible environmental conditions encounterable. A possible solution is represented by data augmentation techniques [504], [438] but still researchers and practitioners have to deal with model complexity which is strictly correlated to the predictive accuracy: the more accurate the model the higher its parameter count needs to be hardening the deployment stage. Finally, a common characteristic of data-driven models is their lack of transparency and explainability which is instead a key requirement in Industry 5.0, as explained in section I.

For the reasons mentioned above, model based approaches can still be useful in the modern era of PdM in the development of hybrid systems where domain knowledge is merged with data-driven architectures leveraging the advantages of both worlds. The inaccuracy of physics-based models can be enhanced using ML or DL algorithms to model more complex phenomena through data and reduce the amount of training data for data-driven solutions acting as data simulators and producing physics-enhanced input features for the models.

In [386] model-based approaches are used to produce data but also to enrich the feature space of the training set by

discovering some additional parameters which can improve the predictive power of DL models. Specifically, the training set of the data-driven model (i.e. a CNN architecture), which is normally composed of condition monitoring data produced by sensor readings, is enriched with scenario-descriptor operating conditions, virtual sensors and unobservable model parameters obtained as the result of the calibration of a physics-based model solved using an Unscented Kalman Filter (UKF) [505]. The obtained hybrid model shows improved performances with the respect to a solely data-driven model and it also requires less training samples thanks to its physics-enhanced input features.

The typical methodology employed in hybrid RUL estimation involves using a physics-based model to describe easy phenomenon in a very precise way, while more complex behaviors are explained through data using DL models. For example in [387] a PINN based RNN model is composed of physics-informed layers to model well known components of airplane fleets (i.e. fatigue crack growth) while data-driven layers are used to model more complex aspects like the internal loads. Similarly in [389], where the task is to predict corrosion-fatigue of an AI 2024-T3 alloy employed in fuselage panels of aircraft wings, physics-based layers are used to model crack growth through the Paris Law while challenging effects such as damage bias due to corrosion are modelled by data-driven layers. Finally, in RUL estimation of main bearings in wind turbines [388] relatively well known physical effects (i.e. L10 fatigue life) are modelled through physics-informed layers while degradation due to poor greasing conditions are left for data-driven layers.

Another very common example of hybrid PdM is represented by PINN [506], NN enhanced or integrated with physics rules and concepts. PINN combine the strengths of physics-based models, which are highly accurate in estimating machines' HI and degradation patterns, with those of purely data-driven models, which offer computational efficiency.

There are two kinds of PINN: *sequential integration models* in which the two components (i.e. the physics-based model and the data-driven model) act independently and *hybridized physical models* where the equations governing the physics-based models are integrated in the data-driven model. An example of this approach is contained in [391] where a CNN model for RUL estimation of lithium-ion batteries is integrated with Open Circuit Voltage (OCV) curves reconstruction methods to fully characterize the batteries degradation pattern and exploit those information in the training process.

A traditional application of PINN is presented in [392] where physical rules are included as regularization terms in the definition of the loss function used to train an RNN model which is used for the RUL estimation of discrete power electronic devices. In particular, the loss function is enriched with a term to control the monotonicity of the predicted RUL signal and to make sure that the prediction falls inside a specific bound. Similarly in [393] domain knowledge is incorporated in the loss function of an LSTM network for RUL prediction on supercapacitors. Moreover, Bayesian Optimization (BO) Optimization is used to tune the balance between the two components of the loss function (i.e. the training loss and the

physics-inspired regularization term).

In [390] a hybrid method for RUL estimation of rolling element bearings is proposed. In this case the degree of bearing damage is estimated by a combination of the Adaptive Spectrum Mode Extraction (ASME) theory with Fault Feature Gain (FFG) to adaptively monitor bearings faults frequencies. After the detection of incipient faults a DL models combining GRU and BNN is used to obtain the final RUL prediction.

In [394] a central role is played by the self-attention mechanism, typical of the transformer model [500], which is firstly used to map the high-dimensional sensor measurement data into a low-dimensional space by selecting only the key features. An additional attention-based network is used to learn the physics of the system as the partial differential equations between the hidden states and the predicted RUL.

4) *Reinforcement Learning Approaches for RUL Estimation*: Maintenance Scheduling is the third and final step of the maintenance pipeline, presented in IV-A. It is usually a component of Decision-Making Modules triggered after the ML or DL models failure predictions are produced [507]. It is a challenging task because it may involve several machines with different failure types and it requires to coordinate the maintenance operations among the team of experts that will act on the equipments necessity reparations. The optimization of this process is crucial in the industrial production chain since it may lead to a significant reduction of the machines downtime under maintenance and to a more efficient use of human resources. This task requires the ability to perform sequential decisions in a stochastic environment, which is the main reason why RL approaches are becoming more and more popular in Maintenance Scheduling. As RL is an automatic learning paradigm that learns through interaction with the environment it can solve some of the typical challenges of PdM like the scarcity of labeled data, documented failures and the presence of multiple operating and working conditions of equipments. RL agents are in fact able to learn the optimal policy to follow in order to minimize the downtime of the machines and to maximize the efficiency of the maintenance operations.

It is worth noting how the application of RL in PdM is quite challenging. In fact, differently from traditional RL formalizations in which the environment is fully observable, stationary and states fully measurable, in industrial applications there are several factors that can quickly modify the state transition probabilities such as production load variations, performed or missed maintenance and external weather [367]. Moreover, in Multi Agent RL applications in PdM an action taken by an agent can change the environment for the other agent/s. As a consequence Hidden-Mode MDP (HM-MDP)s are used to formalize the problem. This is the most complex type of Markov Decision Process (MDP) since no model of the environment is available and states are not fully measurable because determined by noisy sensor recordings. Moreover there is also the necessity of considering continuous and variable length time intervals for action duration, in fact using discrete time formulations could lead to the miss of a maintenance event. This additional requirement defines Semi HM-MDP.

The typical formulation of an RL agent in the field of PdM can thus be summarized as follows:

- **Environment**: The environment is represented by the equipment to monitor or even by the entire plant. Producing a detailed model of the environment can turn into an extremely complex task, thus model-free approaches are preferred.
- **States**: States represent different equipment health stages and are determined by condition monitoring performed through sensors measuring various quantities such as vibration, temperature, toll wear and similar.
- **Actions**: Typically three actions can be performed by the PdM scheduler agent: in case the machine's health is acceptable no action is performed, a single component can be replaced or a complete maintenance operation can be scheduled.

Several works can be found in the literature, according to [367] the most commonly used RL approaches are the basic Q-learning and its variants [508], Actor Critic based methods [509], policy-gradient methods (i.e. Proximal Policy Optimization (PPO) [510] and Deep Deterministic Policy Gradient (DDPG) [511]).

Some works in the literature are worth to be described in more details: for instance, [368] proposed a Multi Agent RL algorithm. The agents work in parallel on different copies of the same machine with the aim of maximizing the time in which the machine is in the *working* state. Maintenance is scheduled when needed assigning the task to a technician.

[512] focuses on a serial production system and uses a RL algorithm based on value function estimation to avoid Machine Stoppage Bottlenecks. In [513] a PPO-LSTM model is proposed in the IIoT framework. As already discussed in previous sections the LSTM architecture is chosen thanks to its excellent capabilities of extracting temporal features from sensor measurements. To speed up training of Deep Reinforcement Learning (DRL) based architectures [514] proposed a knowledge transfer method called Transfer Learning from Demonstrations (TLD) where expert-demonstration data are stored in an Experience Replay (ER) buffer to improve learning efficiency when the system is adapted to a new environment. [515] combined supervised learning for RUL prediction with DRL for maintenance scheduling. In particular a CNN network equipped with Monte Carlo Dropout (MCD) is used to estimate the probability distribution of the RUL signal, which is later used to generate the states of the DRL algorithm (i.e. Soft Actor Critic) which is used to schedule maintenance action based on the trend of the estimated RUL. A final work worth citing is [516] which focuses on bringing interpretability on the table through a Hierarchical DRL framework where, to facilitate the RL task, a probabilistic model (i.e. Input-Output Hidden Markov Model (IOHMM)) is used to filter the large amount of condition monitoring data to include just recordings of equipment's critical states. When these states are reached the DRL model comes in the picture. Moreover to limit the RL agent training time Behavioral Cloning (BC) is used to initialize a baseline policy.

C. XAI in PdM

One of the main drawbacks associated to the usage of AI related technologies in PdM is the black-box nature of the used models and the lack of transparency and explainability on their outputs, which can cause untrustworthiness in the users of these systems, who are generally not familiar with these technologies. Consequently, one of the key requirements when applying PdM methodologies in industrial settings is the ability to provide explanations on the model predictions to increase their trustworthiness towards the domain experts.

Traditionally the field of XAI modelling is characterized by two stages: model prediction and prediction explanation [468]. This is particularly true in case a model agnostic explanation algorithm is used. These algorithms, of which SHAP [278] and LIME [115] are the two most common examples, are agnostic in the sense that are independent on the model architecture but base their interpretation just on the relationship between model's inputs and outputs. A more efficient solution is represented by model specific algorithms which exploit the internal structure of a specific model to provide insights on its reasoning. As a consequence these XAI approaches come with the limitation of being usable just on the specific models for which they are designed. LRP is an example of model specific XAI since it is tailored for NN architectures.

Model agnostic XAI approaches find applications also in PHM: [468] benchmarks different regression models (i.e. from Random Forests to NN) on end of life prediction and successively interprets their outcomes using SHAP and LIME.

A peculiar ad-hoc interpretation method is introduced in [469] which consists in inserting a so called *feature selection layer* immediately after the input layer of a Deep Neural Network (DNN) based model which will contain the weight (i.e. importance) of each input feature which is learned during model training. The model is trained using an L1 loss function in order to produce sparse solutions and thus eliminate irrelevant features, indirectly performing a feature selection operation. Moreover a Ranking Quality Score (RQS) is introduced in order to provide a quantitative evaluation of the goodness of different feature subsets and also to compare different models on the RUL estimation task.

As described in previous sections CNNs architectures are very common in the field of RUL estimation. While being very accurate and efficient in the extraction of local spatial features, crucial for the production of acceptable RUL estimates, the architecture of these models is highly complex making interpretability an hard task. A possible approach to increase its trustworthiness is contained in [470] where the LRP XAI algorithm is employed on time-frequency spectral images of vibration signals produced by an induction motor. With LRP it's possible to identify regions of the input highly contributing to the final prediction by analyzing the pixel-level representation returned by the algorithm. Similarly [406] also applies LRP as an explanation algorithm on a Multi Scale Deep CNN used for the RUL prediction. Additionally, in the pre processing steps raw data are clusterized into different operational modes and physics-informed features are extracted to better capture the nature of degradation. Another DL based

model widely used in PdM is LSTM which is explained through an hybrid attention mechanism in [433].

Another utilization of XAI tools for PdM is their insertion in a complete PdM pipeline comprehending AD, RUL estimation and maintenance planning. In this context, explanations are performed in the AD stage with the aim of detecting the most relevant feature describing the detected anomaly or failure mode, acting as a feature selection mechanism to equip RUL estimation model with the most essential descriptors needed for end of life estimation [471], [446], [119].

For a more detailed description of how XAI is used in PdM the reader is referred to the following surveys on the topic: [472], [473], [474], [475].

D. PdM Industrial Applications

In this section an overview of the most common applications of PdM in the industrial domain is provided, with a focus on how these methodologies are applied and deployed and how they can enhance the performance of industrial systems.

When considering the application of PdM in the industrial setting several challenges have to be taken into account. This task in fact requires failure data of the equipment, a rarity in the industrial domain, where the practice of sharing fault records is avoided for economical and privacy reasons. Moreover multiple failures are usually responsible of a machine's life degradation, making it challenging to correctly label each one of them. As a result researchers and practitioners lack of standard guidelines on the implementation of such prognostics systems. Moreover, the lack of publicly available datasets decreases the chances of being able to carry out a fair comparison between different RUL estimation methods making most of the RUL prediction approaches to be case-dependent [482].

1) *RUL Estimation for Lithium-Ion Batteries:* A common field of application of PdM methodologies is the one of lithium-ion batteries that is becoming increasingly popular in the recent years thanks to the development of electric vehicles. The development of prognostic applications in this field is of crucial importance because of the risks associated to the usage of these devices outside of their operating limits: fire, toxic gas release and electric shock [419].

One of the main challenges for the application of RUL estimation on these kind of devices is that the degradation process may also be inverted, with batteries capable of restoring some of their capacity. This phenomenon is called Capacity Regeneration Phenomena (CRP) [406] and dramatically increases the difficulty of the task since most of the RUL estimation approaches are based on the assumption that the degradation phenomena leading an equipment to the end of its life is a monotonic process.

Considering the application of RUL based prognostic methods in the field of lithium-ion batteries it is possible to see the usage of all the different RUL estimation approaches listed in the previous sections.

An hybrid based approach is employed in [406] to deal with CRP. In particular, the battery capacity degradation signal is decomposed into a seasonal, residual and trend component.

The latter component is the one mostly correlated with the RUL and thus estimated through a Gaussian model. On the other hand the local features (i.e. seasonal and residual components), which represent the Capacity Regeneration Phenomena (CPR) phenomena, are estimated through a Bidirectional GRU data-driven model.

A similar approach is employed in [396] where the global deterioration trend and the local capacity regeneration are decoupled using EMD and then estimated through an Auto Regressive Integrated Moving Average (ARIMA) based model. As already described in IV-B3, another hybrid approach for RUL estimation consists in enriching the knowledge of data-driven models with domain knowledge on the batteries degradation process [391].

In [382] a purely data-driven approach is considered. In fact, the hierarchical spatial characteristics together with the temporal dependencies typical of the prognostics scenario are learned through the so called *Convo-LSTM* network which combines the 1DCNN (i.e. 1 dimensional CNN) architecture with the LSTM one. In particular, an LSTM cell is used to model the temporal patterns while convolutional operations are used instead of the classical multiplication to exploit their weight-sharing strategy and locally focused representation abilities. The combination of CNNs and LSTM networks is also exploited in [420] with the addition of an Autoencoder that deals with the lack of data by extracting additional features from the input space. Moreover a post prediction smoothing filter is applied on the RUL signal to solve the discontinuity problem in the prediction result. In [421] BO is used in the hyperparameter tuning of these hybrid models in order to maximize their predictive accuracy.

LSTM model is employed also in [418] where the focus is on the estimation of the SOH of the battery. Inaccurate SOH estimation is known to lead to spontaneous combustion accidents so it's crucial to be able to provide an estimate of this quantity which is defined as the ratio of a fully charged battery in the current state on the capacity of a fully charged on as it is when initially bought (i.e Nominal Capacity):

$$SOH \approx \frac{\text{Current actual capacity (Ah)}}{\text{Nominal Capacity (Ah)}} \times 100\%$$

Conventionally the end of life is determined when a device capacity is lower than the APT defined as 70-80 % of the rated capacity. This quantity is correlated to other features in batteries data such as the number of discharging cycles, the time required to reach a certain voltage level or the time required to reach a certain temperature. This fact is exploited in [418] where Support Vector Regression (SVR) and LSTM models where trained on those three extracted features to predict the SOH.

Another commonly used architecture in data-driven temporal modelling is the Transformer [500] which is however also known for its quadratic time complexity that makes its use unpractical when dealing with long degradation life patterns. This challenge is dealt in [422] with the introduction of an Exponential Transformer where an exponential decaying measure is employed in the attention mechanism to focus

more on nearby data points. Moreover, similarly to the works cited above, the input data are decoupled into their trend and fluctuation components to produce highly interpretable results. A Transformer based architecture is also used in [423] where a Patch-wise Attention Network (DPAN) correlates patches of the signal to capture both global features (i.e. intra-patch attention) and local ones (i.e. inter-patch attention). Successively the relationships among these features are modelled through a Feature-wise Attention Network (FAN) employing the traditional self-attention.

2) *RUL Estimation for Rolling Element Bearings*: Rolling bearings have an huge impact in the performances of rotating machinery equipment and have been recognized as one of the main sources of failures. For this reason the RUL estimation task for rolling element bearings is crucial in the field of machine prognostics to avoid unexpected failure and to enhance efficiency and safety.

Feature extraction techniques, as the ones outlined in IV-B2, are highly beneficial in the field of bearing diagnostics and prognostics, in particular if frequency domain features are extracted from vibration signals. In fact, as outlined in [517], ground truth knowledge is available since some specific harmonic frequency are directly related to specific faults in bearings: Ball Pass Frequency Outer (BPFO), Ball Pass Frequency Inner (BPFI), Ball Spin Frequency (BFS).

These features are exploited in [446] where a complete PdM pipeline is implemented: firstly incipient bearing faults are detected through the IF AD algorithm, successively Un-supervised Fault Classification is performed thanks to the usage of the XAI SHAP algorithm which ranks the input features in terms of importance in the AD task and the most important fault related feature is used to detect the fault affecting the equipment. Finally RUL prediction is conducted by estimating the Weibull Distribution on the most important variable according to SHAP. Weibull-based failure probability modelling is also studied on its own in [518], for both individual machines and entire systems and across different dataset sizes.

A more complex method for the extraction of fault related features is explored in [390] called ASME which decomposes the original vibration signal into multiple simple Intrinsic Mode Functions (IMF)s. Successively the FFG method is used to adaptively calculate the amplitude of each fault frequency which is a crucial components for the estimation of RUL in bearings. Finally, the study considers the combination of a GRU architecture for temporal modelling and a BNN architecture to enrich the RUL prediction with a quantification of its uncertainty, which is a crucial factor in the field of PHM as detailed in IV-B2.

Feature extraction is addressed in an original way in [403] where a Dynamic Complexity Characteristic Entropy (DCCE) method is used to fully characterize the degradation trend of rolling bearing's useful life. Successively the HI is constructed using PCA and VAE. Finally, features extracted in this way are fed into a Deep Quantum Neural Network (DQNN) to obtain the final RUL prediction.

Even though feature extraction based approaches are naturally more suitable for rolling elements PdM, end-to-end

DL solutions are also explored. For example, in [447] the Recurrent Convolutional Network with Wide First Layer Kernels and Residual Shrinkage Building Unit (MSWR-LRCN) architecture is presented. In this model soft thresholding and attention are incorporated in a CNN model to form an Residual Shrinkage Building Unit (RSBU) unit whose aim is to remove noisy and redundant feature from the input and provide the model just with the features relevant for the task at hand. This is the first path of the feature extraction pipeline which focuses on local features. In parallel an Attention LSTM block is used to extract information spanning large number of time steps. Finally a BiLSTM block is used for the final RUL prediction.

The combination of soft thresholding and attention to eliminate redundant information is also employed in [448] in order to improve the prediction accuracy of a TCN based model.

A fundamental requirement for a task so deeply connected to industrial and real-world scenarios is the ability of producing solutions able to transfer knowledge between different scenarios (from the test rig to the real equipment) and working conditions. Transfer Learning techniques represent a viable solution in this direction. In [374] the Multisource Domain Meta Transfer Learning (MD-MTL) transfer learning method is used to migrate the estimation capabilities of a model composed of a Convolutional Encoder and ProbSparse Multihead Attention Decoder (CE-PSAD) from the test rig to real wind turbine bearings.

Another example of the utility of transfer learning is represented by [400]. In this work firstly a detailed construction of HI indexes is performed. Next, a dual-task (i.e. health assessment and RUL prediction) feature extractor is used to extract useful features exploiting a Convolutional Block Attention Module (CBAM) and Bidirectional Gated Recurrent Unit (BiGRU) architecture. Successively a Model Migration module is responsible to perform Domain Adaptation to move from the training (i.e. test rig) to the target domain (i.e. real-world scenario). Moreover the Weibull Proportional Hazards Model (WPHM) physics-based model is integrated into the data-driven model loss function, thus creating a PINN model as the ones presented in section IV-D1

3) *IoT Papers*: In this sections we consider works whose main focus is posed on the IoT architecture used to provide the monitoring data to the prognostic system. As presented in I, IoT infrastructure are nowadays the standard for PdM systems because of the large amount of benefits associated with them. Other than potential benefits, several challenges have also to be considered, as outlined in [476]. These include integrating diverse sensor systems, handling high volumes of data, ensuring data security and addressing costs and scalability of the implementation of such systems. Moreover, it is also crucial to consider that the transition from traditional maintenance practices to IoT-based ones will come with technical and organizational difficulties.

The typical pipeline of an IoT-based PdM system involves deploying multiple sensors on industrial machines to monitor critical variables, such as vibration or temperature, transmit these data through a cloud infrastructure where data are pre processed to deal with noisy or missing recordings and to perform predictive analysis through ML or DL models.

An example of such pipeline is provided in [477] where a complete PdM infrastructure is presented: temperature, humidity and vibration data are acquired with an IoT system composed of an Arduino Uno and ESP-32 sensors. Successively a cloud infrastructure is employed for data storage, data preprocessing and for identification of incipient faults through the usage of a RF model for Feature Selection and an initial classification of the machine's status as normal or fault. Successively an LSTM model is used to detect anomalous and faulty patterns in the data by providing alarms to maintenance personell through SMS in case a fault is detected.

A similar system is implemented in [478] where communication protocols like Message Queuing Telemetry Transport (MQTT), Hypertext Transfer Protocol (HTTP), OPC Unified Architecture (OPC-UA) are employed for the transmission of data from sensors to the cloud infrastructure. Differently from [477] in this case notification of potential issues in the monitored equipment are communicated through a web-based dashboard.

[479] deals with two crucial aspects in the application of PdM to the industry: data privacy which is handled by employing a FL framework for distributed and secure model training and XAI to equip model predictions with interpretable insights for the maintenance operators.

[376] proposed an Artificial Intelligence of Things (AIoT) system for PdM of the wheel and axle systems of high-speed trains. Differently from typical approaches employing IoT in this case quite complex DL algorithms are considered for the predictive part: Multi Scale Transformer (MST) and Deep Wavelet Transform (DWT). Considering the complexity and computational power required by these models some edge computing solutions were adopted for the deployment of this solution on a railway infrastructure. The predictions produced by these models are used by a Maintenance Decision-Making Layer which can generate maintenance schedules based on RUL predictions, raise alarms in case anomalies are detected and it is also integrated with the train management system to synchronize maintenance actions with train schedules to avoid disruptions of the transportation system.

[480] proposed a ML-based PdM system for a circular knitting machine. Machine speed and stops data are collected through an IoT infrastructure which uses a Printed Circuit Board (PCB), a ESP32 microcontroller and several sensors whose recordings are collected and managed through the *mongoBD* database management system. For what concerns the ML component, the Ada Boost model is used to classify different classes of machine's faults in advance in order to provide target interventions.

As it can be noted from the examples above in IoT-based PdM, predictions are not done on the RUL of the equipment but sensor recordings are directly analysed to detect failures in advance and consequently perform maintenance operations before taking place.

The advent of 5G cellular technologies has an impact on IoT-based systems which can be furtherly enhanced improving real-time data collection. [481] studies the application of this technology in the case study of a factory floor where Automated Guided Vehicles (AGV) are used to transport hazardous

liquid that, if leaked, can endanger workers. Thus, exploiting a 5G based IIoT network for data collection a DL model is trained for RUL prediction. RUL is predicted on the server side and an alert is sent to the AGV in case a failure is forecasted to trigger maintenance intervention.

4) *Business related metrics for the application of PdM in the industry:* As detailed in IV-B2 an aspect of paramount importance in PdM is the trade-off between the two extreme conditions of *unexpected breaks* and *unexploited lifetime* which assume a great importance when PdM systems have to be deployed in an industrial environment where premature breakages or maintenance actions can translate in significant economical losses. For this reason there are several applications of PdM in which the economical aspect is embedded in the RUL estimation model in the form of so called **business metrics**. In [6] Susto et al. proposed a business oriented metric in which, assuming that underestimating the RUL is the preferred error type from an economical point of view, a more cautious approach is embedded in the model by subtracting a constant quantity m (labeled *maintenance window*) from the model output: $\hat{RUL} = \bar{RUL} - m$ where $\bar{RUL}$ is the original model prediction and $\hat{RUL}$ is the adjusted value after the application of the *maintenance window* m . The hyperparameter m needs to be tuned in order to find the correct balance between *unexpected breaks* and *unexploited lifetime*. The following loss function is proposed:

$$J = \rho_{UB} C_{UB} + \rho_{UL} C_{UL} \quad (2)$$

where ρ_{UB}, ρ_{UL} represent the percentage of *unexpected breaks/unexploited lifetime* encountered in the test set and C_{UB}, C_{UL} are the costs associated to those situations, which have to be defined by the users based on the application needs. By optimizing 2 the optimal m window can be defined.

5) *PdM Industrial Case Studies:* In this section some case studies on the integration of PdM in smart manufacturing are presented with the aim of showing the benefits and challenges that this maintenance approach provides.

[483] proposes a hybrid framework for the integration of PdM in manufacturing. As usual the starting point is represented by data collection, performed by IoT-enabled sensors. Successively CNNs networks are employed for time-series feature extraction, followed by the combination of LSTM for sequence modelling and ARIMA model for forecasting. AD algorithms, such as IF and One-Class SVM, can also be considered for an early detection of potentially anomalous conditions. Finally, after a fault is detected, maintenance is scheduled following and Adaptive Maintenance Scheduling approach realized through the usage of an RL agent whose rewards coincide with the operators feedback.

The work presented in [484] is a great example of the computational challenges in the deployment of complex DL models in an industrial scenario. The case study focuses on the implementation of a web application for RUL prediction to guide maintenance interventions for cutting tool machines in milling industries. Due to the computational constraints usually faced in the deployment stage the RUL prediction

methods are all ML-based (i.e. DT, RF, Multi Layer Perceptron (MLP), ...) with MLP being the better performing one.

[498] presents a PdM strategy to substitute the conventional three-stage maintenance strategy employed in nuclear power plants, where maintenance plays a crucial role considering the devastating consequences that an incident in such a plant can cause to the environment. The developed prognostics system is implemented on the circulating water pump of the power plant. The system is composed of different stages: first a sensor selection strategy is used to reduce the number of input features to the RUL prediction models and thus guarantee higher efficiency. Because of the hardness in the collection of degradation data, pseudo-labels are created to train a *LassoNet* model to discard redundant sensors and select solely the ones relevant to for the RUL estimation task. Successively degradation-related sensors are used to construct HI which are used to establish a degradation trend prediction model to perform a short-term prognosis. Considering the fact that multiple degradation sensors are considered for the construction of the HI, an ensemble approach is used for trend prediction. It is worth noting how among the base models for the ensemble Structured State Space Sequence model (S4) is considered. This model is part of the family of SSM models, which are rarely used [465], [519] in the field of RUL estimation. Finally, a long-term RUL prediction is obtained through a model-based approach which is selected based on the degradation trend prediction performed at the previous stage. The RUL prediction is defined as the time when a specific threshold value (determined) is overcome by the HI. For what concerns the industrial deployment, an IoT infrastructure is considered composed by an end device (i.e. the sensors), an edge device represented by a computer which processes real-time monitoring data and perform data cleaning, AD and HI construction. Successively the elaborated data are sent to the cloud server whose task is to train the models for sensor selection, trend degradation and RUL prediction.

Another application of RUL estimation in a nuclear power plant is introduced in [485] where the focus is on Class II components (i.e. critical piping systems that can experience the Flow Accelerated Corrosion (FAC) fault). Due to the scarcity of real data the proposed models for RUL prediction were validated on a synthetic dataset produced by a Finite Element Model (FEM) model of the components under analysis. The synthetic dataset reproduces the operating conditions experienced in a catastrophic pipe rupture incident (caused by the FAC fault) in Surry Unit 2 in December 1986. For what concerns the methodology employed for RUL estimation an AE was used for feature selection and the reconstruction error was used to represent the machine's health status: the smaller the reconstruction error the healthier the machine. Finally an ARIMA model was considered to perform time-series forecasting on the learned HI to predict equipment's degradation trend and produce a RUL estimation. This case study represents an example on how challenging data collection can be in the field of PdM, in particular in cases in which such complicated physical degradation phenomena are considered. In these cases there is the need to exploit model-based approaches (FEM analysis in this case) to simulate

the FAC fault which normally expands at a rate of a few millimeters per year, making it impractical to be monitored through sensing technologies.

A novel PdM approach is presented in [486] for prognostics on marine engines. An LSTM based model is used for failure prediction and it is trained in an online fashion by using data collected in a three-week period to predict failure in the subsequent week and by continuously discarding old data as new ones are available, making it possible for the model to adapt to new environmental conditions that a marine engine can face during the year. Another important contribution is represented by an accurate selection of relevant Key Performance Indicators (KPIs) for model training. Since historical data are not considered in this online-based approach the selection of KPI is performed by analysing common faults in previously designed marine engines and by simulating them to confirm the correlation with the selected KPI.

Another case study is described in [487] where a PdM system is implemented by performing a regression analysis on the sensor data produced by the Alarm Monitoring System (AMS) of large cargo boats. Data Augmentation is applied to produce additional abnormal conditions for model training using a 1D simulation model of the monitored engine. The PdM algorithm is based on the method proposed in [488] where the Revision Generator Engine Condition Criterion (RGCCV) index is introduced to take the maintenance decisions. This index is based on the analysis of the exhaust gas temperature, one of the variables measured with the IoT sensors which is highly correlated with engine failures.

Traditionally, PdM is solved through RUL estimation as a regression task. However there are some case studies in which PdM is posed as a classification problem [6], [489]. In particular in [489] the aim is to predict whether a distribution transformer will fail within a given time horizon. The authors of this case study specify several challenges encountered in dealing with real world data on distribution transformers: missing data, high feature dimensionality, presence of both numerical and categorical features. Finally, when considering PdM as a classification problem dataset imbalance is guaranteed considering that equipment failures are rare events. In order to deal with these challenges, that are typical of real-world datasets, the authors used a tree based method (i.e. RF) to deal with mixed feature types and employed random under-sampling with boosting to create a more balanced dataset. Moreover the authors developed a specific metric called *match in top N* (MITN) in order to avoid a high number of false positives predictions. In fact the costs of premature replacements of transformers may exceed the repairment cost for a sudden failure. This final consideration highlights another crucial factor to take into account when deploying a PdM solution in the industrial world: the economical consequences of prediction errors. Depending on the specific applications a pessimistic or optimistic prediction may have different economical impacts, creating a trade-off of difficult solution. A possible approach to deal with challenge, presented for the first time in [6], is represented by the usage of business-related metrics (IV-D4).

V. INDUSTRIAL USE CASES FROM THE AIMS5.0 PROJECT

This section focuses on the AIMS5.0 European project by introducing its scopes and missions and reporting its current state on some of its Use Case (UC) strictly related to the topics of this chapter.

A. Overview of the AIMS5.0 Project

The AIMS5.0 project is a European research and innovation initiative aimed at accelerating the transition from Industry 4.0 to Industry 5.0 by enabling the systematic integration of AI into industrial processes. The project addresses key technological, organizational, and human-centric challenges to foster sustainable, resilient, and trustworthy manufacturing ecosystems.

The final aim of the project is to develop and validate a comprehensive framework for AI-enabled industrial transformation, supporting the full lifecycle of AI adoption — from use case specification and data management to algorithm development, deployment, validation, and standardization. Through the development of AI methods, architectures, and reference solutions, AIMS5.0 seeks to empower industrial stakeholders to deploy scalable, explainable, and human-centric AI systems that enhance productivity, sustainability, and decision-making in alignment with the Industry 5.0 paradigm.

AIMS5.0 integrates twenty industrial use cases (UCs) from globally renowned companies across sectors such as semiconductor manufacturing, machine tools, robotics, logistics, and production systems. These UCs serve as real-world validation environments to demonstrate how AI-driven solutions can improve operational efficiency, sustainability KPIs, human-machine collaboration, and resilience across complex industrial value chains.

Among the twenty UCs, four are directly aligned with the frameworks presented in this chapter, namely AD, PdM, and FDD:

- Anomaly Detection on Production Processes;
- Edge Anomaly Detection and Predictive Maintenance;
- Tool Breakage Detection (Fault Detection);
- Industrial Production-Ready Anomaly Detection.

The following subsections describe the objectives and operational principles of these four UCs, highlighting their relevance to diagnostic and prognostic AI frameworks in Smart Manufacturing.

B. AI-related Challenges in AIMS5.0 Use Cases

Before delving into a detailed description of the four UCs listed above, this subsection describes the AI-related challenges faced in AIMS5.0 UCs. The most commonly faced challenges can be divided into the following groups:

- **Data Collection** is the most frequent obstacle when dealing with industrial AI integration. In particular issues regard the presence of unbalanced or low-quality data (i.e. defects are rare), difficult connection between ML or DL based tools when data are stored in Manufacturing Execution Systems (MES) or ERP systems, privacy issues due to datasets confidential or owned by third-parties and

data distribution issues as a result of the post COVID pandemic era.

- **Algorithmic problems** regard the lack of best practices to train and validate AI models. Interestingly in some UC the so-called Fear of Missing Out (FOMO) and *fast time to market* pressure can hinder AI adoption. Other challenges regard the lack of computing power, available AI and ML platforms.
- **Human-centric problems** regard the lack of explainability and trustworthiness of some AI solution.

C. Anomaly Detection on Production Processes

This use case addresses anomaly detection within industrial production processes. The primary objective is to identify abnormal behaviours or deviations from expected operational patterns that may affect product quality or process efficiency.

The UC focuses on the development of ML models capable of detecting anomalous conditions in manufacturing workflows. These anomalies may correspond to unusual production states, deviations in process parameters, or unexpected variations in output characteristics. The approach involves collecting process data, defining relevant operational indicators, and designing models that distinguish between normal and abnormal operating regimes.

At its current stage, this use case is characterized by an exploratory development phase. The main efforts are directed toward defining appropriate problem formulations, identifying meaningful data sources, and selecting suitable anomaly detection techniques. The challenge lies not only in model development but also in clearly specifying validation metrics and performance indicators that reflect industrial requirements.

From a methodological perspective, this use case exemplifies the early stages of industrial anomaly detection deployment, where problem definition, data strategy, and KPI specification are critical enablers for successful AI integration.

D. Edge Anomaly Detection and Predictive Maintenance

This use case focuses on PdM and AD executed directly at the edge level. The objective is to enable real-time monitoring of industrial assets without relying on continuous cloud connectivity, thereby addressing latency, bandwidth, and data privacy constraints.

The UC leverages lightweight and interpretable ML models deployed on industrial edge devices such as Programmable Logic Controllers (PLCs), embedded systems, or gateways. Typical algorithms include decision trees, rule-based models, and TinyML neural networks optimized for constrained hardware environments. Gradient-boosted decision trees are used as core models, fine-tuned on self-collected industrial datasets.

A key design choice in this use case is the development of specialized models tailored to specific operational contexts rather than a single generalized solution. This specialization allows the models to capture equipment-specific behaviour and degradation patterns more accurately. The system architecture supports incremental updates and continuous refinement as new operational data becomes available.

This UC represents a concrete implementation of PdM in distributed industrial environments, where computational constraints, model interpretability, and deployment robustness are central design requirements.

E. Tool Breakage Detection in Machine Tool Operations

This use case addresses FDD in precision machining processes, with a specific focus on automatic tool breakage detection. The objective is to detect tool failures during machining operations using high-frequency vibration signals and other sensor data, enabling timely intervention and reducing scrap and downtime.

The approach combines signal processing and supervised ML. First, frequency-domain features are extracted using FFT analysis to capture harmonic patterns associated with tool condition. These features are then used as inputs to binary classification models, including RF, SVM, k-NN, and Logistic Regression, to determine whether the tool is in a *broken* or *not broken* state.

In addition to supervised learning, unsupervised techniques such as motif discovery and change-point detection are used to segment repetitive machining cycles automatically. This segmentation step is crucial for isolating comparable operational patterns and improving classification reliability.

A central challenge in this UC is the transition from experimental datasets to real-world production data, which are often unlabelled and more heterogeneous. Therefore, the use case highlights common industrial issues such as limited labelled data, variability across machining conditions, and the need for robust model generalization.

This UC constitutes a representative example of data-driven FDD in manufacturing systems and illustrates the practical steps required to industrialize supervised diagnostic models.

F. Industrial Production-Ready Anomaly Detection

This use case represents a mature implementation of anomaly detection deployed in a production environment. The objective is to develop and operationalize an application-specific ML model capable of identifying abnormal production conditions at scale.

In contrast to earlier-stage UCs, this use case is characterized by fully specified targets, validation metrics, and KPIs. The developed model has been deployed within the production infrastructure, and a pilot phase is being conducted to evaluate its performance under real operational conditions.

The system follows a complete industrial AI lifecycle: customization to the specific production context, model training on industrial data, deployment within the production environment, and continuous monitoring during operation. Particular attention is given to scalability, robustness, and integration with existing manufacturing systems.

This use case demonstrates the transition from research-oriented anomaly detection methods to production-grade AI solutions. It exemplifies how well-defined objectives, structured validation procedures, and close collaboration between AI developers and industrial stakeholders enable successful deployment in real manufacturing settings.

Overall, these four UCs collectively illustrate different maturity stages and application contexts of AD, PdM, and FDD within Smart Manufacturing. They provide practical insights into how AI-based diagnostic and prognostic techniques are designed, validated, and integrated into industrial environments under the Industry 5.0 paradigm.

VI. COMPARISON, DISCUSSION AND RESEARCH GAP

This section is devoted to perform a comparison between the different methods and applications surveyed in Sections II, III, IV. The comparison will not be targeting the quantitative performances of different diagnostic and prognostic approaches but will rather discuss the main difference and similarities between those and potentially highlight some common research trends in the field and identify potential gaps.

Among data-driven models there is a division in different domains of application of the RUL estimation task. Turbofan engine is the most popular domain, probably because of the popularity of the C-MAPSS based datasets which contain high-quality data and represent a standard for the benchmarking of RUL estimation methods. Given the sequential nature of the C-MAPSS data most of the proposed methods employ the LSTM, CNNs architectures to exploit their excellent temporal and spatial feature extraction capabilities. A lot of solutions decide to merge together both these qualities by combining the LSTM and CNNs architectures. Additionally, the inclusion of attention based architectures for an importance weighting of the time samples is also explored. The high-quality of the C-MAPSS data pushes researchers to design solutions that may be excessively complicated and too focused on the performances showed on this dataset. In this context, it is important to note that the data contained in C-MAPSS and N-CMAPSS are the result of a simulation that, albeit precise, does not guarantee to be a good enough representation of real-world conditions. Consequently, comparable performances once these solutions are applied to real cases are not guaranteed. Moreover the elevated computational complexity associated to the DL architectures employed requires extra work to develop edge computing solutions, as in [376], to make the deployment of such architectures sustainable.

Another negative aspect of approaches benchmarked on the C-MAPSS dataset is reported in [370]: the evaluation criteria used to validate the different approaches presented differs from work to work making impractical to conduct a full comparison in terms of quantitative metrics.

The elevated computational complexity associated to DL based models makes the solution of enriching simpler ML models with engineered input features still a viable choice. This approach is in fact the one with the highest amount of review articles in Table X.

In this context the topic of feature engineering becomes of crucial importance. As explained in IV-B2 pre-defined statistical features in time or frequency domain are typically used. These features usually are able to extract the degradation process of the machine assisting the predictive models in the prediction of the RUL signal. However in some cases this

is not enough and domain knowledge should be employed by pairing data modelling with physical knowledge on the machine deterioration. This is where hybrid RUL methods, grouped in Table IX, should be used. Among these approaches a valid class of models are PINN, in which prior knowledge on the equipment can be used to craft highly informative input features or as a regularization term in the loss function which guides the training process of the underlying NN model.

Inside the manufacturing process PHM constitutes the final step where the fundamental decision on how to face the incoming failure on the machine is taken. Thus is crucial to provide accurate results that should be able to deal with the multiple operating conditions and different environments and domains where the same machine may be employed. As described in previous sections numerous contributions were presented in this direction but this is a challenging tasks and some research gaps have still to be faced. One of them is the transferability of RUL prediction models from simulated environments (i.e. C-MAPSS) to real industrial processes. Developing overly complex models on a synthetic datasets like C-MAPSS, designing specific architectural choices to deal with a very unique and peculiar setting may produce unsatisfiable results when applying said architectures to completely different scenarios. For this reason it's important to employ transfer learning or domain adaptation solutions such as the ones presented in [445], [374], [494], [12], [514], [449] which represent a small fraction of the current state of the art on RUL estimation.

Other than the difficulty on transferring the knowledge between different domains complex DL approaches, normally associated with demanding computational requirements, are not satisfying the requirements of operational efficiency and transparency required by Industry 4.0 and 5.0 requiring the need to develop more efficient and trustworthy solutions. A solution in this direction can be represented by hybrid-based RUL models where the computational load of DL model is reduced by the presence of prior physical knowledge that can help in modelling some less complex components/features of the degradation process leaving to the DL component to explain what cannot be efficiently modelled by model-based approaches to refine the predictions.

Finally, to deal with the human-inspired principles required by Industry 5.0 additional focus should be addressed towards the integration of XAI approaches and UQ methodologies in order to equip the final model prediction with auxiliary information such as a list of the features more closely related to machine degradation and confidence intervals on the RUL estimate. These information can promote informed decision and increase trust in maintenance personell.

VII. CONCLUSION

This survey has comprehensively reviewed AD, FD, and PdM within the context of Industry 4.0 and 5.0. The critical role of AI, IoT, and IIoT in enabling Smart Factories was established, alongside the advancements in diagnostic and prognostic capabilities. State-of-the-art methodologies, encompassing model-based, data-driven, and hybrid approaches for

RUL estimation, were thoroughly examined. The increasing importance of UQ and XAI for building trustworthy and transparent systems, aligning with Industry 5.0's human-centric principles, was also highlighted.

Industrial applications across various sectors demonstrated tangible benefits, though significant real-world challenges persist. These include data scarcity, computational demands of complex AI models, integration complexities and legal or privacy issues related to the data collection process.

To fully realize the potential of these frameworks, several research gaps require addressing: enhance model transferability to real world applications and different domains through transfer learning and domain adaptation approaches, employ edge computing solutions to enable a sustainable application of complex DL based architectures, and increase the application of XAI and UQ methodologies to form human-friendly predictions in order to fulfill the requirements of the Industry 4.0 and 5.0 revolutions. Another key challenge to address is the one related to data privacy, mainly caused by private companies concerned about protecting their resources, which can be faced with FL based solutions.

In conclusion, AD, FD, and PdM are pivotal for intelligent industrial operations. Addressing these identified gaps will enable the industrial sector to fully leverage the transformative capabilities of Industry 4.0 and 5.0, leading to enhanced efficiency, reduced downtime and sustainability.

ACKNOWLEDGMENT

The research leading to these results was conducted in the AIMS5.0 project, which received funding from the Chips Joint Undertaking under grant agreement No. 101112089 and from the partners' national funding programmes.

REFERENCES

- [1] P. Adolphs, H. Bedenbender, D. Dirzus, M. Ehlich, U. Epple, M. Hankel, R. Heidele, M. Hoffmeister, H. Huhle, B. Kärcher, H. Koziol, R. Pichler, S. Pollmeier, F. Schewe, A. Walter, B. Waser, and M. Wollschlaeger, "Reference architecture model industrie 4.0 (RAMI4.0)," VDI/VDE Society for Measurement and Automatic Control (GMA) and ZVEI, Status Report, Jul. 2015.
- [2] H. Lasi, P. Fietke *et al.*, "Industry 4.0," *Business & Inf. systems Eng.*, vol. 6, pp. 239–242, 2014.
- [3] A. Keshar, "Advancing industrial iot and industry 4.0 through digital twin technologies: A comprehensive framework for intelligent manufacturing, real-time analytics and predictive maintenance," *World Journal of Advanced Engineering Technology and Sciences*, vol. 14, pp. 228–240, 01 2025.
- [4] Deloitte, "Internet of things (iot) the rise of the connected world," 2021. [Online]. Available: https://www.deloitte.com/content/dam/Deloitte/in/Documents/technology-media-telecommunications/int-tmt-IoT_Theriseoftheconnectedworld-28aug-noexp.pdf
- [5] A. Frankó, G. Hollósi, D. Ficzer, and P. Varga, "Applied machine learning for IIoT and smart production—methods to improve production quality, safety and sustainability," *Sensors*, vol. 22, no. 23, p. 9148, 2022.
- [6] G. A. Susto, A. Schirru, S. Pampuri, S. McLoone, and A. Beghi, "Machine learning for predictive maintenance: A multiple classifier approach," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 3, pp. 812–820, 2015.
- [7] J. Leng, W. Sha, B. Wang, P. Zheng, C. Zhuang, Q. Liu, T. Wuest, D. Mourtzis, and L. Wang, "Industry 5.0: Prospect and retrospect," *Journal of Manufacturing Systems*, vol. 65, pp. 279–295, 2022. [Online]. Available: <https://www.sciencedirect.com/journal/journal-of-manufacturing-systems/s092658072200017&url=2-s2.0-85139280032&doi=10.1016%2fj.jmsy.2022.09.017&partnerID=40&md5=8a4363e335263e3ed3af9a4c83987f1b>
- [8] E. Valette, H. B. El-Haouzi, and G. Demesure, "Industry 5.0 and its technologies: A systematic literature review upon the human place into iot and cps-based industrial systems," *Computers & Ind. Eng.*, vol. 184, p. 109 426, 2023.
- [9] M. C. Zizic, M. Mladineo *et al.*, "From industry 4.0 towards industry 5.0: A review and analysis of paradigm shift for the people, organization and technology," *Energies*, vol. 15, no. 14, p. 5221, 2022.
- [10] G. Dimitrakopoulos, P. Varga, T. Gutt, G. Schneider, H. Ehm, A. Hoess *et al.*, "Industry 5.0: Research areas and challenges with artificial intelligence and human acceptance," *IEEE Industrial Electronics Magazine*, vol. 18, no. 4, pp. 43–54, 2024.
- [11] A. Saxena and K. Goebel, "Turbopfan engine degradation simulation data set," NASA Ames Prognostics Data Repository, 2008. [Online]. Available: <https://c3.ndc.nasa.gov/dashlink/resources/139/>
- [12] X. Du, J. Chen, J. Wang, H. Zhang, and J. Wen, "Fusion-based dual-task architecture for predicting the remaining useful life of an aeroengine," *Journal of Aerospace Engineering*, vol. 38, no. 3, p. 04 025 004, 2025. [Online]. Available: <https://ascelibrary.org/doi/abs/10.1061/JAEEZ.ASENG-5887>
- [13] S. Chatterjee and A. Keprate, "Exploratory data analysis of the ncmapps dataset for prognostics," in *2021 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, 2021, pp. 1114–1121.
- [14] M. Arias Chao, C. Kulkarni, K. Goebel, and O. Fink, "Aircraft engine run-to-failure dataset under real flight conditions for prognostics and diagnostics," *Data*, vol. 6, no. 1, 2021. [Online]. Available: <https://www.mdpi.com/2306-5729/6/1/5>
- [15] A. C. J. Bonatakis and N. Propeas, "Phm data challenge 2018," PHM Society, 2018. [Online]. Available: <https://c3.ndc.nasa.gov/dashlink/resources/139/>
- [16] B. Saha and K. Goebel, "Battery data set," NASA Ames Prognostics Data Repository, 2007. [Online]. Available: <https://www.nasa.gov/intelligent-systems-division/discovery-and-systems-health/pcoc/pcoc-data-set-repository/>
- [17] P. Nectoux, R. Gouriveau, K. Medjaher, E. Ramasso, B. Chebel-Morello, N. Zerhouni, and C. Varnier, "PRONOSTIA: An experimental platform for bearings accelerated degradation tests," in *Conference on Prognostics and Health Management*, vol. sur CD ROM. Denver, Colorado, United States: IEEE Catalog Number: CPE12PHM-CDR, Jun. 2012, pp. 1–8. [Online]. Available: <https://hal.science/hal-00719503>
- [18] NASA, "Ims bearings — dataset," <https://data.nasa.gov/dataset/ims-bearings>.
- [19] C. W. R. U. Case School of Engineering, "Bearing data center," <https://engineering.case.edu/bearingdatacenter>.
- [20] C. A. Rieth, B. D. Amsel, R. Tran, and M. B. Cook, "Additional tennessee eastman process simulation data for anomaly detection evaluation," *Harvard Dataverse*, vol. 1, p. 2017, 2017.
- [21] J. Downs and E. Vogel, "A plant-wide industrial process control problem," *Computers & Chemical Eng.*, vol. 17, no. 3, pp. 245–255, 1993, industrial challenge problems in process control.
- [22] V. Zaccaria, C. Masiero, D. Dandolo, and G. A. Susto, "Enabling efficient and flexible interpretability of data-driven anomaly detection in industrial processes with acme-ad," in *2024 10th International Conference on Control, Decision and Information Technologies (CoDIT)*. IEEE, 2024, pp. 1375–1380.
- [23] D. Frizzo, F. Borsatti, A. Arcudi, A. De Moliner, R. Oboe, and G. A. Susto, "Interpretable data-driven anomaly detection in industrial processes with exiffi," in *2024 IEEE 8th Forum on Research and Technologies for Society and Industry Innovation (RTSI)*. IEEE, 2024, pp. 595–600.
- [24] D. Neupane, M. R. Bouadjene, R. Dazeley, and S. Aryal, "Data-driven machinery fault diagnosis: A comprehensive review," *Neurocomputing*, vol. 627, p. 129 588, 2025. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S09252321225002607>
- [25] D. M. Hawkins, *Identification of outliers*. Springer, 1980, vol. 11.
- [26] R. Foorthuis, "A typology of data anomalies," in *Information Processing and Management of Uncertainty in Knowledge-Based Systems. Theory and Foundations*, J. Medina, M. Ojeda-Aciego, J. L. Verdegay, D. A. Pelta, I. P. Cabrera, B. Bouchon-Meunier, and R. R. Yager, Eds. Cham: Springer International Publishing, 2018, pp. 26–38.
- [27] —, "On the nature and types of anomalies: a review of deviations in data," *International Journal of Data Science and Analytics*, vol. 12, no. 4, pp. 297–331, Oct. 2021. [Online]. Available: <https://doi.org/10.1007/s41060-021-00265-1>

- [28] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, Jul. 2009. [Online]. Available: <https://doi.org/10.1145/1541880.1541882>
- [29] V. Hodge and J. Austin, "A Survey of Outlier Detection Methodologies," *Artificial Intelligence Review*, vol. 22, no. 2, pp. 85–126, Oct. 2004. [Online]. Available: <https://doi.org/10.1023/B:AIRE.0000045502.10941.a9>
- [30] M. A. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko, "A review of novelty detection," *Signal Processing*, vol. 99, pp. 215–249, 2014. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S016516841300515X>
- [31] L. Ruff, J. R. Kauffmann, R. A. Vandermeulen, G. Montavon, W. Samek, M. Kloft, T. G. Dietterich, and K.-R. Müller, "A unifying review of deep and shallow anomaly detection," *Proceedings of the IEEE*, vol. 109, no. 5, pp. 756–795, 2021.
- [32] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: identifying density-based local outliers," in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, 2000, pp. 93–104.
- [33] S. Papadimitriou, H. Kitagawa, P. B. Gibbons, and C. Faloutsos, "Loc: Fast outlier detection using the local correlation integral," in *Proceedings 19th international conference on data engineering (Cat. No. 03CH37405)*. IEEE, 2003, pp. 315–326.
- [34] L. J. Latecki, A. Lazarevic, and D. Pokrajac, "Outlier detection with kernel density functions," in *International workshop on machine learning and data mining in pattern recognition*. Springer, 2007, pp. 61–75.
- [35] W. Hu, J. Gao, B. Li, O. Wu, J. Du, and S. Maybank, "Anomaly detection using local kernel density estimation and context-based regression," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 2, pp. 218–233, 2018.
- [36] E. Schubert, A. Zimek, and H.-P. Kriegel, "Generalized outlier detection with flexible kernel density estimates," in *Proceedings of the 2014 SIAM international conference on data mining*. SIAM, 2014, pp. 542–550.
- [37] K. Olsen, R. M. H. Lindrup, and M. Mørup, "Think global, adapt local: Learning locally adaptive k-nearest neighbor kernel density estimators," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2024, pp. 4114–4122.
- [38] M. Goldstein and A. Dengel, "Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm," *KI-2012: poster and demo track*, vol. 1, pp. 59–63, 2012.
- [39] M. Gebski and R. K. Wong, "An efficient histogram method for outlier detection," in *International Conference on Database Systems for Advanced Applications*. Springer, 2007, pp. 176–187.
- [40] T. Pevn' y, "Loda: Lightweight on-line detector of anomalies," *Machine Learning*, vol. 102, no. 2, pp. 275–304, 2016.
- [41] Z. Li, Y. Zhao, N. Botta, C. Ionescu, and X. Hu, "Copod: copulabased outlier detection," in *2020 IEEE international conference on data mining (ICDM)*. IEEE, 2020, pp. 1118–1123.
- [42] W. Liu, D. Cui, Z. Peng, and J. Zhong, "Outlier detection algorithm based on gaussian mixture model," in *2019 IEEE international conference on power, intelligent computing and systems (ICPICS)*. IEEE, 2019, pp. 488–492.
- [43] S. Guan, Z. He, S. Ma, and M. Gao, "Conditional normalizing flow for multivariate time series anomaly detection," *ISA transactions*, vol. 143, pp. 231–243, 2023.
- [44] B. Nachman and D. Shih, "Anomaly detection with density estimation," *Physical Review D*, vol. 101, no. 7, p. 075 042, 2020.
- [45] A. Ryzhikov, M. Borisyak, A. Ustyuzhanin, and D. Derkach, "Nfad: fixing anomaly detection using normalizing flows," *PeerJ Computer Science*, vol. 7, p. e757, 2021.
- [46] B. Rosenhahn and C. Hirche, "Quantum normalizing flows for anomaly detection," *Physical Review A*, vol. 110, no. 2, p. 022 443, 2024.
- [47] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen, "Deep autoencoding gaussian mixture model for unsupervised anomaly detection," in *International conference on learning representations*, 2018.
- [48] G. Zamberg, M. Shihov, O. Lindenbaum, and A. Averbuch, "Tabadm: Unsupervised tabular anomaly detection with diffusion models," *arXiv preprint arXiv:2307.12336*, 2023.
- [49] Y. Huang, Z. Zhang, M. Liao, Y. Tan, and S. Zhou, "A hybrid distanced-based outlier detection approach," in *2012 International Conference on Systems and Informatics (ICSAI2012)*. IEEE, 2012, pp. 2212–2216.
- [50] J. Mei, M. Liu, Y.-F. Wang, and H. Gao, "Learning a mahalanobis distance-based dynamic time warping measure for multivariate time series classification," *IEEE transactions on Cybernetics*, vol. 46, no. 6, pp. 1363–1374, 2015.
- [51] J. R. Wells, K. M. Ting, and T. Washio, "Linearn: A new approach to nearest neighbour density estimator," *Pattern Recognition*, vol. 47, no. 8, pp. 2702–2720, 2014.
- [52] K. M. Ting, Y. Zhu, M. Carman, Y. Zhu, and Z.-H. Zhou, "Overcoming key weaknesses of distance-based neighbourhood methods using a data dependent dissimilarity measure," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1205–1214.
- [53] H.-P. Kriegel, M. Schubert, and A. Zimek, "Angle-based outlier detection in high-dimensional data," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2008, pp. 444–452.
- [54] A. Lohrer, D. Kazempour, M. Hünemörder, and P. Kröger, "Comadout—a robust outlier detection algorithm based on comad," *Machine Learning*, vol. 113, no. 10, pp. 8061–8135, 2024.
- [55] Y. Hou, Z. Chen, M. Wu, C.-S. Foo, X. Li, and R. M. Shubair, "Mahalanobis distance based adversarial network for anomaly detection," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 3192–3196.
- [56] M. Sun, Y. Su, S. Zhang, Y. Cao, Y. Liu, D. Pei, W. Wu, Y. Zhang, X. Liu, and J. Tang, "Ctf: Anomaly detection in high-dimensional time series with coarse-to-fine model transfer," in *IEEE INFOCOM 2021-IEEE conference on computer communications*. IEEE, 2021, pp. 1–10.
- [57] C. Qiu, T. Pfommer, M. Kloft, S. Mandt, and M. Rudolph, "Neural transformation learning for deep anomaly detection beyond images," in *International conference on machine learning*. PMLR, 2021, pp. 8703–8714.
- [58] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 eighth IEEE international conference on data mining*. IEEE, 2008, pp. 413–422.
- [59] J. Lesouple, C. Baudoin, M. Spigai, and J.-Y. Tournet, "Generalized isolation forest for anomaly detection," *Pattern Recognition Letters*, vol. 149, pp. 109–119, 2021.
- [60] J. Choudhury, P. Ky, Y. Ren, and C. Shi, "Hypersphere for branching node for the family of isolation forest algorithms," in *2021 IEEE International Conference on Smart Computing (SMARTCOMP)*. IEEE, 2021, pp. 418–423.
- [61] T. R. Bandaragoda, K. M. Ting, D. Albrecht, F. T. Liu, Y. Zhu, and J. R. Wells, "Isolation-based anomaly detection using nearest-neighbor ensembles," *Computational Intelligence*, vol. 34, no. 4, pp. 968–998, 2018.
- [62] T. Liu, Z.-Y. Zhou, and L. Yang, "Layered isolation forest: A multi-level subspace algorithm for improving isolation forest," *Neurocomputing*, vol. 581, p. 127 525, 2024. [Online]. Available: <https://api.semanticscholar.org/CorpusID:268334315>
- [63] F. Leveni, G. W. Cassales, B. Pfahringer, A. Bifet, and G. Boracchi, "Online isolation forest," *ArXiv*, vol. abs/2505.09593, 2025. [Online]. Available: <https://api.semanticscholar.org/CorpusID:278602380>
- [64] A. Putina, M. Sozio, D. Rossi, and J. M. Navarro, "Random histogram forest for unsupervised anomaly detection," *2020 IEEE International Conference on Data Mining (ICDM)*, pp. 1226–1231, 2020. [Online]. Available: <https://api.semanticscholar.org/CorpusID:228195076>
- [65] G. Pang, K. M. Ting, and D. W. Albrecht, "Lesinn: Detecting anomalies by identifying least similar nearest neighbours," *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, pp. 623–630, 2015. [Online]. Available: <https://api.semanticscholar.org/CorpusID:18178448>
- [66] Y. Zhao, M. K. Hryniewicki, Z. Nasrullah, and Z. Li, "Lscp: Locally selective combination in parallel outlier ensembles," *ArXiv*, vol. abs/1812.01528, 2018. [Online]. Available: <https://api.semanticscholar.org/CorpusID:54445467>
- [67] H. Sarvari, C. Domeniconi, and G. Stilo, "Graph-based selective outlier ensembles," *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, 2018. [Online]. Available: <https://api.semanticscholar.org/CorpusID:4903507>
- [68] B. Wang and Z. Mao, "A dynamic ensemble outlier detection model based on an adaptive k-nearest neighbor rule," *Inf. Fusion*, vol. 63, pp. 30–40, 2020. [Online]. Available: <https://api.semanticscholar.org/CorpusID:219499716>

- [69] H. Xu, G. Pang, Y. Wang, and Y. Wang, "Deep isolation forest for anomaly detection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, pp. 12 591–12 604, 2022. [Online]. Available: <https://api.semanticscholar.org/CorpusID:249642184>
- [70] D. Chakraborty, V. Narayanan, and A. Ghosh, "Integration of deep feature extraction and ensemble learning for outlier detection," *Pattern Recognit.*, vol. 89, pp. 161–171, 2019. [Online]. Available: <https://api.semanticscholar.org/CorpusID:126415389>
- [71] A. Iqbal and R. Amin, "Time series forecasting and anomaly detection using deep learning," *Computers & Chemical Engineering*, vol. 182, p. 108 560, 2023.
- [72] H. Hoffmann, "Kernel pca for novelty detection," *Pattern Recognition*, vol. 40, no. 3, pp. 863–874, 2007. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0031320306003414>
- [73] Y.-J. Lee, Y.-R. Yeh, and Y.-C. F. Wang, "Anomaly detection via online oversampling principal component analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 7, pp. 1460–1470, 2013.
- [74] A. Adler, M. Elad, Y. Hel-Or, and E. Rivlin, "Sparse coding with anomaly detection," *Journal of Signal Processing Systems*, vol. 79, no. 2, pp. 179–188, 2015.
- [75] L. Gjorgiev and S. Gievska, "Time series anomaly detection with variational autoencoder using mahalanobis distance," in *ICT Innovations 2020. Machine Learning and Applications*, V. Dimitrova and I. Dimitrovski, Eds. Cham: Springer International Publishing, 2020, pp. 42–55.
- [76] Y. Su, Y. Zhao, C. Niu, R. Liu, W. Sun, and D. Pei, "Robust anomaly detection for multivariate time series through stochastic recurrent neural network," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, ser. KDD '19*. New York, NY, USA: Association for Computing Machinery, 2019, p. 2828–2837. [Online]. Available: <https://doi.org/10.1145/3292500.33030672>
- [77] S. Guan, B. Zhao, Z. Dong, M. Gao, and Z. He, "Gtad: Graph and temporal neural network for multivariate time series anomaly detection," *Entropy*, vol. 24, 2022. [Online]. Available: <https://api.semanticscholar.org/CorpusID:249166131>
- [78] J. Audibert, P. Michiardi, F. Guyard, S. Marti, and M. A. Zuluaga, "Usad: Unsupervised anomaly detection on multivariate time series," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, ser. KDD '20*. New York, NY, USA: Association for Computing Machinery, 2020, p. 3395–3404. [Online]. Available: <https://doi.org/10.1145/3394486.3403392>
- [79] X. Chen, L. Deng, F. Huang, C. Zhang, Z. Zhang, Y. Zhao, and K. Zheng, "Daemon: Unsupervised anomaly detection and interpretation for multivariate time series," in *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, 2021, pp. 2225–2230.
- [80] P. Reddy and A. Singh, "Anogan for tabular data: A novel approach to anomaly detection," in *HCI International 2024 – Late Breaking Posters*, C. Stephanidis, M. Antona, S. Ntoa, and G. Salvendy, Eds. Cham: Springer Nature Switzerland, 2025, pp. 319–330.
- [81] L. Chen, H. Jiang, L. Wang, J. Li, M. Yu, Y. Shen, and X. Du, "Generative adversarial synthetic neighbors-based unsupervised anomaly detection," *Scientific Reports*, vol. 15, no. 1, p. 16, 2025.
- [82] Y. F. Wang, Y. Jiong, G. P. Su, and Y. R. Qian, "A new outlier detection method based on optics," *Sustainable Cities and Society*, vol. 45, pp. 197–212, 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2210670718318328>
- [83] P. K. Jain, M. S. Bajpai, and R. Pamula, "A modified dbscan algorithm for anomaly detection in time-series data with seasonality," *Int. Arab J. Inf. Technol.*, vol. 19, no. 1, pp. 23–28, 2022.
- [84] D. Krleža, B. Vrdoljak, and M. Brčić, "Statistical hierarchical clustering algorithm for outlier detection in evolving data streams," *Machine Learning*, vol. 110, no. 1, pp. 139–184, 2021.
- [85] R. Chitrakar and H. Chuanhe, "Anomaly detection using support vector machine classification with k-medoids clustering," in *2012 Third Asian Himalayas International Conference on Internet*, 2012, pp. 1–5.
- [86] U. Rashid, M. F. Saleem, S. Rasool, A. Abdullah, H. Mustafa, and A. Iqbal, "Anomaly detection using clustering (k-means with dbscan) and smo," *Journal of Computing & Biomedical Informatics*, vol. 7, no. 02, 2024.
- [87] J. Li, H. Izakian, W. Pedrycz, and I. Jamal, "Clustering-based anomaly detection in multivariate time series data," *Applied Soft Computing*, vol. 100, p. 106919, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1568494620308577>
- [88] C. Aytekin, X. Ni, F. Cricri, and E. Aksu, "Clustering and unsupervised anomaly detection with l2 normalized deep auto-encoder representations," in *2018 International Joint Conference on Neural Networks (IJCNN)*, 2018, pp. 1–6.
- [89] G. S. Chadha, I. Islam, A. Schwung, and S. X. Ding, "Deep convolutional clustering-based time series anomaly detection," *Sensors*, vol. 21, no. 16, p. 5488, 2021.
- [90] H. Song, P. Li, and H. Liu, "Deep clustering based fair outlier detection," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, ser. KDD '21*. New York, NY, USA: Association for Computing Machinery, 2021, pp. 1481–1489. [Online]. Available: <https://doi.org/10.1145/3447548.3467225>
- [91] A. G. Roselin, P. Nanda, S. Nepal, and X. He, "Intelligent anomaly detection for large network traffic with optimized deep clustering (odc) algorithm," *IEEE Access*, vol. 9, pp. 47 243–47 251, 2021.
- [92] J. Ma and S. Perkins, "Time-series novelty detection using one-class support vector machines," in *Proceedings of the International Joint Conference on Neural Networks*, 2003., vol. 3, 2003, pp. 1741–1745 vol. 3.
- [93] M. Ji and H.-J. Xing, "Adaptive-weighted one-class support vector machine for outlier detection," in *2017 29th Chinese Control And Decision Conference (CCDC)*, 2017, pp. 1766–1771.
- [94] S. Goyal, A. Raghunathan, M. Jain, H. V. Simhadri, and P. Jain, "DROCC: Deep robust one-class classification," in *Proceedings of the 37th International Conference on Machine Learning, ser. Proceedings of Machine Learning Research*, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13–18 Jul 2020, pp. 3711–3721. [Online]. Available: <https://proceedings.mlr.press/v119/goyal20c.html>
- [95] L. Shen, Z. Li, and J. Kwok, "Timeseries anomaly detection using temporal hierarchical one-class network," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 13 016–13 026. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/97e401a02082021fd24957f852e0e475-Paper.pdf
- [96] H. Xu, Y. Wang, S. Jian, Q. Liao, Y. Wang, and G. Pang, "Calibrated one-class classification for unsupervised time series anomaly detection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 11, pp. 5723–5736, 2024.
- [97] Z. Yang, X. Wang, Y. Chen, H. Dou, and H. Sang, "Robust deep one-class classification time series anomaly detection," *Computers, Materials & Continua*, vol. 83, no. 3, pp. 5181–5197, 2025. [Online]. Available: <http://www.techscience.com/cmc/v83n3/60982>
- [98] G. Zhong, F. Liu, J. Jiang, B. Wang, and C. P. Chen, "Refining one-class representation: A unified transformer for unsupervised time-series anomaly detection," *Information Sciences*, vol. 656, p. 119 914, 2024. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0020025523014998>
- [99] X. Yang and X. Li, "Atdad: One-class adversarial learning for tabular data anomaly detection," *Computers & Security*, vol. 134, p. 103 449, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0167404823003590>
- [100] J. Ye, Z. Tan, Y. Hu, X. Yang, G. Cheng, and K. Huang, "Disentangling tabular data towards better one-class anomaly detection," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 12, pp. 13 061–13 068, Apr. 2025. [Online]. Available: <https://ojs.aaai.org/index.php/AAAI/article/view/33425>
- [101] E. H. M. Pena, M. V. O. de Assis, and M. L. Proença, "Anomaly detection using forecasting methods arima and hwsd," in *2013 32nd International Conference of the Chilean Computer Science Society (SCCC)*, 2013, pp. 63–66.
- [102] A. M. Bianco, M. García Ben, E. J. Martínez, and V. J. Yohai, "Outlier detection in regression models with arima errors using robust estimates," *Journal of Forecasting*, vol. 20, no. 8, pp. 565–579, 2001. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/for.768>
- [103] V. Kozitsin, I. Katser, and D. Lakontsev, "Online forecasting and anomaly detection based on the arima model," *Applied Sciences*, vol. 11, no. 7, 2021. [Online]. Available: <https://www.mdpi.com/2076-3417/11/7/3194>
- [104] P. Malhotra, L. Vig, G. M. Shroff, and P. Agarwal, "Long short term memory networks for anomaly detection in time series," in *The European Symposium on Artificial Neural Networks*, 2015. [Online]. Available: <https://api.semanticscholar.org/CorpusID:43680425>

- [105] M. Munir, S. A. Siddiqui, A. Dengel, and S. Ahmed, "Deepant: A deep learning approach for unsupervised anomaly detection in time series," *IEEE Access*, vol. 7, pp. 1991–2005, 2019.
- [106] S. Ahmad, A. Lavin, S. Purdy, and Z. Agha, "Unsupervised real-time anomaly detection for streaming data," *Neurocomputing*, vol. 262, pp. 134–147, 2017, online Real-Time Learning Strategies for Data Streams. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0925231217309864>
- [107] A. Deng and B. Hooi, "Graph neural network-based anomaly detection in multivariate time series," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, 2021, pp. 4027–4035.
- [108] B. Koltai, A. Gazdag, and G. Ács, "Improving CAN anomaly detection with correlation-based signal clustering," *Infocommunications Journal*, vol. 15, no. 4, pp. 17–25, 2023.
- [109] S. Tuli, G. Casale, and N. R. Jennings, "Tranad: Deep transformer networks for anomaly detection in multivariate time series data," *Proc. VLDB Endow.*, vol. 15, pp. 1201–1214, 2022. [Online]. Available: <https://api.semanticscholar.org/CorpusID:246035402>
- [110] M. Zhao, H. Peng, L. Li, and Y. Ren, "Graph attention network and informer for multivariate time series anomaly detection," *Sensors*, vol. 24, no. 5, 2024. [Online]. Available: <https://www.mdpi.com/1424-8220/24/5/1522>
- [111] T. Mathonsi and T. L. v. Zyl, "Multivariate anomaly detection based on prediction intervals constructed using deep learning," *Neural Computing and Applications*, vol. 37, no. 2, pp. 707–721, 2025.
- [112] Z. Li, Y. Zhu, and M. Van Leeuwen, "A survey on explainable anomaly detection," *ACM Trans. Knowl. Discov. Data*, vol. 18, no. 1, Sep. 2023. [Online]. Available: <https://doi.org/10.1145/3609333>
- [113] E. Brusa, L. Cibrario, C. Delprete, and L. G. Di Maggio, "Explainable ai for machine fault diagnosis: understanding features' contribution in machine learning models for industrial condition monitoring," *Applied Sciences*, vol. 13, no. 4, p. 2038, 2023.
- [114] K. Jang, K. E. S. Pilario, N. Lee, I. Moon, and J. Na, "Explainable artificial intelligence for fault diagnosis of industrial processes," *IEEE Transactions on Industrial Informatics*, vol. 21, no. 1, pp. 4–11, 2025.
- [115] M. T. Ribeiro, S. Singh, and C. Guestrin, "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*, ser. KDD '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 1135–1144. [Online]. Available: <https://doi.org/10.1145/2939672.2939778>
- [116] A. K. M. Nor, S. R. Pedapati, M. Muhammad, and V. Leiva, "Abnormality detection and failure prediction using explainable bayesian deep learning: Methodology and case study with industrial data," *Mathematics*, vol. 10, no. 4, 2022. [Online]. Available: <https://www.mdpi.com/2227-7390/10/4/554>
- [117] K. Patra, R. N. Sethi, and D. K. Behera, "Anomaly detection in rotating machinery using autoencoders based on bidirectional lstm and gru neural networks," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 30, no. 4, pp. 1637–1653, 2022.
- [118] S. Ahmad, K. Styp-Rekowski, S. Nedelkoski, and O. Kao, "Autoencoder-based condition monitoring and anomaly detection method for rotating machines," in *2020 IEEE International Conference on Big Data (Big Data)*. IEEE, 2020, pp. 4093–4102.
- [119] L. C. Brito, G. A. Susto, J. N. Brito, and M. A. Duarte, "An explainable artificial intelligence approach for unsupervised fault detection and diagnosis in rotating machinery," *Mechanical Systems and Signal Processing*, vol. 163, p. 108 105, 2022.
- [120] T. Reeber, J. Henninger, N. Weingarz, P. M. Simon, M. Berndt, M. Glatt, B. Kirsch, R. Eisseler, J. C. Aurich, and H.-C. Möhring, "Tool condition monitoring in drilling processes using anomaly detection approaches based on control internal data," *Procedia Cirp*, vol. 121, pp. 216–221, 2024.
- [121] C. Shi, B. Luo, H. Li, B. Li, X. Mao, and F. Peng, "Anomaly detection via unsupervised learning for tool breakage monitoring," *International Journal of Machine Learning and Computing*, vol. 6, no. 5, p. 256, 2016.
- [122] J. Shim, T. Joung, S. Lee, and J.-I. Ha, "Audio data-driven anomaly detection for induction motor based on generative adversarial networks," in *2022 IEEE Energy Conversion Congress and Exposition (ECCE)*. IEEE, 2022, pp. 1–5.
- [123] J. Shim, G. C. Lim, and J.-I. Ha, "Unsupervised anomaly detection for electric drives based on variational auto-encoder," in *2022 IEEE applied power electronics conference and exposition (appec)*. IEEE, 2022, pp. 1703–1708.
- [124] J. M. Barrera, A. Reina, A. Mate, and J. C. Trujillo, "Fault detection and diagnosis for industrial processes based on clustering and autoencoders: a case of gas turbines," *International Journal of Machine Learning and Cybernetics*, vol. 13, no. 10, pp. 3113–3129, 2022.
- [125] M. Carletti, C. Masiero, A. Beghi, and G. A. Susto, "A deep learning approach for anomaly detection with industrial time series data: a refrigerators manufacturing case study," *Procedia Manufacturing*, vol. 38, pp. 233–240, 2019.
- [126] J. Goh, S. Adepu, M. Tan, and Z. S. Lee, "Anomaly detection in cyber physical systems using recurrent neural networks," in *2017 IEEE 18th international symposium on high assurance systems engineering (HASE)*. IEEE, 2017, pp. 140–145.
- [127] M. M. Aboelwafa, K. G. Seddik, M. H. Eldefrawy, Y. Gadallah, and M. Gidlund, "A machine-learning-based technique for false data injection attacks detection in industrial iot," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 8462–8471, 2020.
- [128] W.-H. Choi and J. Kim, "Unsupervised learning approach for anomaly detection in industrial control systems," *Applied System Innovation*, vol. 7, no. 2, p. 18, 2024.
- [129] D. Benka, D. Horváth, L. Špendla, G. Gašpar, and M. Strémy, "Machine learning-based detection of anomalies, intrusions and threats in industrial control systems," *IEEE Access*, 2025.
- [130] C. Goetz and B. Humm, "Decentralized real-time anomaly detection in cyber-physical production systems under industry constraints," *Sensors*, vol. 23, no. 9, p. 4207, 2023.
- [131] C. Sánchez-Zas, X. Larriva-Novo, V. A. Villagrà, M. S. Rodrigo, and J. I. Moreno, "Design and evaluation of unsupervised machine learning models for anomaly detection in streaming cybersecurity logs," *Mathematics*, vol. 10, no. 21, p. 4043, 2022.
- [132] D. Leite, E. Andrade, D. Rativa, and A. M. Maciel, "Fault detection and diagnosis in industry 4.0: a review on challenges and opportunities," *Sensors (Basel, Switzerland)*, vol. 25, no. 1, p. 60, 2024.
- [133] H. Li, M. Jia, and Z. Mao, "Dynamic reconstruction principal component analysis for process monitoring and fault detection in the cold rolling industry," *Journal of Process Control*, vol. 128, p. 103 010, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0959152423000951>
- [134] O. AlShorman, F. Alkahatni, M. Masadeh, M. Irfan, A. Glowacz, F. Althobiani, J. Kozik, and W. Glowacz, "Sounds and acoustic emission-based early fault diagnosis of induction motor: A review study," *Advances in Mechanical Engineering*, vol. 13, no. 2, p. 1687814021996915, 2021.
- [135] D. Zhang, E. Stewart, M. Entezami, C. Roberts, and D. Yu, "Intelligent acoustic-based fault diagnosis of roller bearings using a deep graph convolutional network," *Measurement*, vol. 156, p. 107585, 2020.
- [136] G. Niu, X. Dong, and Y. Chen, "Motor fault diagnostics based on current signatures: A review," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–19, 2023.
- [137] D. Ding, Q.-L. Han, Y. Xiang, X. Ge, and X.-M. Zhang, "A survey on security control and attack detection for industrial cyber-physical systems," *Neurocomputing*, vol. 275, pp. 1674–1683, 2018.
- [138] A. Garcés-Jiménez, A. Rodrigues, J. M. Gomez-Pulido, D. Raposo, J. A. Gomez-Pulido, J. S. Silva, and F. Boavida, "Industrial internet of things embedded devices fault detection and classification. a case study," *Internet of Things*, vol. 25, p. 101 042, 2024.
- [139] M. E. H. Benbouzid, "A review of induction motors signature analysis as a medium for faults detection," *IEEE transactions on industrial electronics*, vol. 47, no. 5, pp. 984–993, 2002.
- [140] J. Sim, S. Kim, H. J. Park, and J.-H. Choi, "A tutorial for feature engineering in the prognostics and health management of gears and bearings," *Applied Sciences*, vol. 10, no. 16, 2020. [Online]. Available: <https://www.mdpi.com/2076-3417/10/16/5639>
- [141] C. Zhang, W. Yin, C. Zhang, K. Peng, and X. Zhang, "A kpi-related fault diagnosis method for multimode manufacturing processes based on supervised minimal gated unit and sparse broad learning system," *Applied Soft Computing*, p. 113 679, 2025.
- [142] D. Leite, A. Martins Jr, D. Rativa, J. F. De Oliveira, and A. M. Maciel, "An automated machine learning approach for real-time fault detection and diagnosis," *Sensors*, vol. 22, no. 16, p. 6138, 2022.
- [143] J. Yao, C. Liu, K. Song, C. Feng, and D. Jiang, "Fault diagnosis of planetary gearbox based on acoustic signals," *Applied Acoustics*, vol. 181, p. 108 151, 2021.
- [144] S. Sekiou, A. Behloul, R. Nait-Said, and Z. Chiremsel, "Alarms prediction and classification in industrial processes using supervised machine learning techniques: A case study in an algerian gas plant," *Computers & Chemical Engineering*, p. 109 378, 2025.

- [145] L. Arya and G. P. Gupta, "Ensemble filter-based feature selection model for cyber attack detection in industrial internet of things," in *2023 9th International conference on advanced computing and communication systems (ICACCS)*, vol. 1. IEEE, 2023, pp. 834–840.
- [146] L. Wan, K. Gong, G. Zhang, X. Yuan, C. Li, and X. Deng, "An efficient rolling bearing fault diagnosis method based on spark and improved random forest algorithm," *Ieee Access*, vol. 9, pp. 37 866–37 882, 2021.
- [147] W. Wang, F. Harrou, B. Bouyeddou, S.-M. Senouci, and Y. Sun, "Cyber-attacks detection in industrial systems using artificial intelligence-driven methods," *International journal of critical infrastructure protection*, vol. 38, p. 100542, 2022.
- [148] A. N. Saberi, A. Belahcen, J. Sobra, and T. Vaimann, "Lightgbmbased fault diagnosis of rotating machinery under changing working conditions using modified recursive feature elimination," *IEEE Access*, vol. 10, pp. 81 910–81 925, 2022.
- [149] M. C. Gomes, L. C. Brito, M. B. da Silva, and M. A. V. Duarte, "Tool wear monitoring in micromilling using support vector machine with vibration and sound sensors," *Precision Engineering*, vol. 67, pp. 137–151, 2021.
- [150] Y. E. Karabacak, N. G. Özmen, and L. Gümüşel, "Intelligent worm gearbox fault diagnosis under various working conditions using vibration, sound and thermal features," *Applied Acoustics*, vol. 186, p. 108 463, 2022.
- [151] J. Wei, H. Huang, L. Yao, Y. Hu, Q. Fan, and D. Huang, "New imbalanced bearing fault diagnosis method based on sample-characteristic oversampling technique (scote) and multi-class ls-svm," *Applied Soft Computing*, vol. 101, p. 107 043, 2021.
- [152] K. Ravikumar, C. Madhusudana, H. Kumar, and K. Gangadharan, "Classification of gear faults in internal combustion (ic) engine gearbox using discrete wavelet transform features and k star algorithm," *Engineering Science and Technology, an International Journal*, vol. 30, p. 101 048, 2022.
- [153] G. Rayjade, A. Bhagure, P. B. Kushare, R. Bhandare, V. Matsagar, and A. Chaudhari, "Performance evaluation of machine learning algorithms and impact of activation functions in artificial neural network classifier for bearing fault diagnosis," *Journal of Vibration and Control*, vol. 31, no. 9–10, pp. 1859–1873, 2025.
- [154] S.-K. S. Fan, C.-Y. Hsu, D. ming Tsai, F. He, and C.-C. Cheng, "Data-driven approach for fault detection and diagnostic in semiconductor manufacturing," *IEEE Transactions on Automation Science and Engineering*, vol. 17, pp. 1925–1936, 2020. [Online]. Available: <https://api.semanticscholar.org/CorpusID:218776407>
- [155] S. Gawde, S. Patil, S. Kumar, P. Kamat, and K. Kotecha, "An explainable predictive maintenance strategy for multi-fault diagnosis of rotating machines using multi-sensor data fusion," *Decision Analytics Journal*, vol. 10, p. 100 425, 2024.
- [156] J. Jiang, W. Li, Z. Wen, Y. Bie, H. Schwarz, and C. Zhang, "Series arc fault detection based on random forest and deep neural network," *IEEE Sensors Journal*, vol. 21, no. 15, pp. 17 171–17 179, 2021.
- [157] M. Hosseinpour-Zarnaq, M. Omid, and E. Biabani-Aghdam, "Fault diagnosis of tractor auxiliary gearbox using vibration analysis and random forest classifier," *Information Processing in Agriculture*, vol. 9, no. 1, pp. 60–67, 2022.
- [158] Z. Wang, J. Guo, P. Xia, L. Wang, C. Zhang, Q. Leng, and K. Zheng, "Feature selection for chillers fault diagnosis from the perspectives of machine learning and field application," *Energy and Buildings*, vol. 307, p. 113 937, 2024.
- [159] G. Xu, M. Liu, Z. Jiang, D. Söffker, and W. Shen, "Bearing fault diagnosis method based on deep convolutional neural network and random forest ensemble learning," *Sensors*, vol. 19, no. 5, p. 1088, 2019.
- [160] W. Xie, Z. Li, Y. Xu, P. Gardoni, and W. Li, "Evaluation of different bearing fault classifiers in utilizing cnn feature extraction ability," *Sensors*, vol. 22, no. 9, 2022. [Online]. Available: <http://www.mdpi.com/1424-8220/22/9/3314>
- [161] S. Esakimuthu Pandarakone, Y. Mizuno, and H. Nakamura, "A comparative study between machine learning algorithm and artificial intelligence neural network in detecting minor bearing fault of induction motors," *Energies*, vol. 12, no. 11, p. 2105, 2019.
- [162] H. Ruan, B. Dorneanu, H. Arellano-Garcia, P. Xiao, and L. Zhang, "Deep learning-based fault prediction in wireless sensor network embedded cyber-physical systems for industrial processes," *IEEE Access*, vol. 10, pp. 10 867–10 879, 2022.
- [163] Z. Jin, D. Chen, D. He, Y. Sun, and X. Yin, "Bearing fault diagnosis based on vmd and improved cnn," *Journal of Failure Analysis and Prevention*, vol. 23, no. 1, pp. 165–175, 2023.
- [164] Z. Dong, D. Zhao, and L. Cui, "An intelligent bearing fault diagnosis framework: One-dimensional improved self-attention-enhanced cnn and empirical wavelet transform," *Nonlinear Dynamics*, vol. 112, no. 8, pp. 6439–6459, 2024.
- [165] S. Zhao, Y. Duan, N. Roy, and B. Zhang, "A deep learning methodology based on adaptive multiscale cnn and enhanced highway lstm for industrial process fault diagnosis," *Reliability engineering & system safety*, vol. 249, p. 110 208, 2024.
- [166] T. Li, C. Sun, O. Fink, Y. Yang, X. Chen, and R. Yan, "Filter-informed spectral graph wavelet networks for multiscale feature extraction and intelligent fault diagnosis," *IEEE Transactions on Cybernetics*, vol. 54, no. 1, pp. 506–518, 2023.
- [167] B. A. Tama, M. Vania, S. Lee, and S. Lim, "Recent advances in the application of deep learning for fault diagnosis of rotating machinery using vibration signals," *Artificial Intelligence Review*, vol. 56, no. 5, pp. 4667–4709, 2023.
- [168] Y. Hou, H. Huang, Z. Chen, X. Cai, Y. Han, and X. Zhai, "A novel multi-scale adaptive feature fusion framework for accurate and efficient bearing fault diagnosis," *Signal, Image and Video Processing*, vol. 19, no. 15, pp. 1–12, 2025.
- [169] X. Zhao, J. Yao, W. Deng, P. Ding, Y. Ding, M. Jia, and Z. Liu, "Intelligent fault diagnosis of gearbox under variable working conditions with adaptive intra-class and inter-class convolutional neural network," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 9, pp. 6339–6353, 2022.
- [170] B. Guan, X. Bao, H. Qiu, and D. Yang, "Enhancing bearing fault diagnosis using motor current signals: A novel approach combining time shifting and causal convnets," *Measurement*, vol. 226, p. 114 049, 2024.
- [171] X. Wang, D. Mao, and X. Li, "Bearing fault diagnosis based on vibroacoustic data fusion and 1d-cnn network," *Measurement*, vol. 173, p. 108 518, 2021.
- [172] D.-T. Hoang and H.-J. Kang, "Rolling element bearing fault diagnosis using convolutional neural network and vibration image," *Cognitive Systems Research*, vol. 53, pp. 42–50, 2019.
- [173] C.-F. Chien, W.-T. Hung, and E. T.-Y. Liao, "Redefining monitoring rules for intelligent fault detection and classification via cnn transfer learning for smart manufacturing," *IEEE Transactions on Semiconductor Manufacturing*, vol. 35, no. 2, pp. 158–165, 2022.
- [174] J. Jiao, M. Zhao, J. Lin, and K. Liang, "A comprehensive review on convolutional neural network in machine fault diagnosis," *Neurocomputing*, vol. 417, pp. 36–63, 2020.
- [175] Y. Cheng, M. Lin, J. Wu, H. Zhu, and X. Shao, "Intelligent fault diagnosis of rotating machinery based on continuous wavelet transform-local binary convolutional neural network," *Knowledge-Based Systems*, vol. 216, p. 106 796, 2021.
- [176] Y.-M. Hsueh, V. R. Ittangihai, W.-B. Wu, H.-C. Chang, and C.-C. Kuo, "Fault diagnosis system for induction motors by cnn using empirical wavelet transform," *Symmetry*, vol. 11, no. 10, p. 1212, 2019.
- [177] B. Peng, H. Xia, X. Lv, M. Annor-Nyarko, S. Zhu, Y. Liu, and J. Zhang, "An intelligent fault diagnosis method for rotating machinery based on data fusion and deep residual neural network," *Applied Intelligence*, vol. 52, no. 3, pp. 3051–3065, 2022.
- [178] Q. Zhang and L. Deng, "An intelligent fault diagnosis method of rolling bearings based on short-time fourier transform and convolutional neural network," *Journal of Failure Analysis and Prevention*, vol. 23, no. 2, pp. 795–811, 2023.
- [179] A. Mohammad-Alikhani, E. Jamshidpour, S. Dhale, M. Akrami, S. Pardhan, and B. Nahid-Mobarakeh, "Fault diagnosis of electric motors by a channel-wise regulated cnn and differential of stft," *IEEE Transactions on Industry Applications*, 2025.
- [180] H. Wang, Z. Liu, D. Peng, and Z. Cheng, "Attention-guided joint learning cnn with noise robustness for bearing fault diagnosis and vibration signal denoising," *ISA transactions*, vol. 128, pp. 470–484, 2022.
- [181] G. Niu, E. Liu, X. Wang, P. Ziehl, and B. Zhang, "Enhanced discriminate feature learning deep residual cnn for multitask bearing fault diagnosis with information fusion," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 1, pp. 762–770, 2022.
- [182] J. Miao, C. Deng, H. Zhang, and Q. Miao, "Interactive channel attention for rotating component fault detection with strong noise and limited data," *Applied Soft Computing*, vol. 138, p. 110 171, 2023.
- [183] B. Hu, J. Liu, and Y. Xu, "A novel multi-scale convolutional neural network incorporating multiple attention mechanisms for bearing fault diagnosis," *Measurement*, vol. 242, p. 115 927, 2025.

- [184] Y. Chen and R. Zhang, "Hybrid dual-channel attention cnn and extreme gradient boosting for industrial process model development and fault diagnosis," *IEEE Internet of Things Journal*, 2025.
- [185] K. Lu, C. Chen, T. Wang, L. Cheng, and J. Qin, "Fault diagnosis of industrial robot based on dual-module attention convolutional neural network," *Autonomous Intelligent Systems*, vol. 2, no. 1, p. 12, 2022.
- [186] Y. Keshun, W. Puzhou, and G. Yingkui, "Toward efficient and interpretative rolling bearing fault diagnosis via quadratic neural network with bi-lstm," *IEEE Internet of Things Journal*, vol. 11, no. 13, pp. 23 002–23 019, 2024.
- [187] F. Dao, Y. Zeng, and J. Qian, "Fault diagnosis of hydro-turbine via the incorporation of bayesian algorithm optimized cnn-lstm neural network," *Energy*, vol. 290, p. 130326, 2024.
- [188] H. Bilal, M. S. Obaidat, M. S. Aslam, J. Zhang, B. Yin, and K. Mahmood, "Online fault diagnosis of industrial robot using iort and hybrid deep learning techniques: An experimental approach," *IEEE Internet of Things Journal*, vol. 11, no. 19, pp. 31 422–31 437, 2024.
- [189] G. Fu, Q. Wei, Y. Yang, and C. Li, "Bearing fault diagnosis based on cnn-bilstm and residual module," *Measurement Science and Technology*, vol. 34, no. 12, p. 125050, 2023.
- [190] X. Xu, Z. Tao, W. Ming, Q. An, and M. Chen, "Intelligent monitoring and diagnostics using a novel integrated model based on deep learning and multi-sensor feature fusion," *Measurement*, vol. 165, p. 108 086, 2020.
- [191] J. Li, Y. Liu, and Q. Li, "Intelligent fault diagnosis of rolling bearings under imbalanced data conditions using attention-based deep learning method," *Measurement*, vol. 189, p. 110 500, 2022.
- [192] Y. Jin, L. Hou, and Y. Chen, "A time series transformer based method for the rotating machinery fault diagnosis," *Neurocomputing*, vol. 494, pp. 379–395, 2022.
- [193] S. Han, H. Shao, J. Cheng, X. Yang, and B. Cai, "Convformer-nse: A novel end-to-end gearbox fault diagnosis framework under heavy noise using joint global and local information," *IEEE/ASME Transactions on Mechatronics*, vol. 28, no. 1, pp. 340–349, 2022.
- [194] Y. Han, F. Zhang, Z. Li, Q. Wang, C. Li, P. Lai, T. Li, F. Teng, and Z. Jin, "Mt-convformer: a multi-task bearing fault diagnosis method using a combination of cnn and transformer," *IEEE Transactions on Instrumentation and Measurement*, 2024.
- [195] Y. Hou, J. Wang, Z. Chen, J. Ma, and T. Li, "Diagnosisformer: An efficient rolling bearing fault diagnosis method based on improved transformer," *Engineering Applications of Artificial Intelligence*, vol. 124, p. 106 507, 2023.
- [196] X. Luo, H. Wang, T. Han, and Y. Zhang, "Fft-trans: Enhancing robustness in mechanical fault diagnosis with fourier transform-based transformer under noisy conditions," *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1–12, 2024.
- [197] Y. Ding, M. Jia, Q. Miao, and Y. Cao, "A novel time-frequency transformer based on self-attention mechanism and its application in fault diagnosis of rolling bearings," *Mechanical Systems and Signal Processing*, vol. 168, p. 108 616, 2022.
- [198] X. Tang, Z. Xu, and Z. Wang, "A novel fault diagnosis method of rolling bearing based on integrated vision transformer model," *Sensors*, vol. 22, no. 10, p. 3878, 2022.
- [199] C. Li, J. Chen, C. Yang, J. Yang, Z. Liu, and P. Davari, "Convolutional neural network-based transformer fault diagnosis using vibration signals," *Sensors*, vol. 23, no. 10, p. 4781, 2023.
- [200] S. Wang, Q. Xu, S. Zhu, and B. Wang, "Making transformer hear better: Adaptive feature enhancement based multi-level supervised acoustic signal fault diagnosis," *Expert Systems with Applications*, vol. 264, p. 125 736, 2025.
- [201] W. Sun, R. Yan, R. Jin, J. Xu, Y. Yang, and Z. Chen, "Liteformer: a lightweight and efficient transformer for rotating machine fault diagnosis," *IEEE Transactions on Reliability*, vol. 73, no. 2, pp. 1258–1269, 2023.
- [202] Y. Xiao, H. Shao, J. Wang, S. Yan, and B. Liu, "Bayesian variational transformer: A generalizable model for rotating machinery fault diagnosis," *Mechanical Systems and Signal Processing*, vol. 207, p. 110 936, 2024.
- [203] L.-z. Wang, J.-f. Chi, Y.-q. Ding, H.-y. Yao, Q. Guo, and H.-q. Yang, "Transformer fault diagnosis method based on smote and ngo-gbdt," *Scientific Reports*, vol. 14, no. 1, p. 7179, 2024.
- [204] H. Chen, J. Wei, H. Huang, L. Wen, Y. Yuan, and J. Wu, "Novel imbalanced fault diagnosis method based on generative adversarial networks with balancing serial cnn and transformer (bcrgan)," *Expert Systems with Applications*, vol. 258, p. 125 171, 2024.
- [205] Z. Chen, J. Xu, C. Alippi, S. X. Ding, Y. Shardt, T. Peng, and C. Yang, "Graph neural network-based fault diagnosis: a review," *arXiv preprint arXiv:2111.08185*, 2021.
- [206] M. Jin, H. Y. Koh, Q. Wen, D. Zambon, C. Alippi, G. I. Webb, I. King, and S. Pan, "A survey on graph neural networks for time series: Forecasting, classification, imputation, and anomaly detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.
- [207] T. Li, Z. Zhao, C. Sun, R. Yan, and X. Chen, "Multireceptive field graph convolutional networks for machine fault diagnosis," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 12, pp. 12 739–12 749, 2020.
- [208] C. Li, L. Mo, and R. Yan, "Rolling bearing fault diagnosis based on horizontal visibility graph and graph neural networks," in *2020 international conference on sensing, measurement & data analytics in the era of artificial intelligence (ICSMD)*. IEEE, 2020, pp. 275–279.
- [209] K. Wen, R. Huang, D. Li, Z. Chen, and W. Li, "Gradient-based interpretable graph convolutional network for bearing fault diagnosis," in *2023 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*. IEEE, 2023, pp. 1–6.
- [210] W. Lu, W. Wang, X. Qin, and Z. Cai, "Enhancing fault diagnosis in mechanical systems with graph neural networks addressing class imbalance," *Mathematics*, vol. 12, no. 13, p. 2064, 2024.
- [211] M. Jia, J. Hu, Y. Liu, Z. Gao, and Y. Yao, "Topology-guided graph learning for process fault diagnosis," *Industrial & Engineering Chemistry Research*, vol. 62, no. 7, pp. 3238–3248, 2023.
- [212] D. Wu and J. Zhao, "Process topology convolutional network model for chemical process fault diagnosis," *Process Safety and Environmental Protection*, vol. 150, pp. 93–109, 2021.
- [213] D. Chen, R. Liu, Q. Hu, and S. X. Ding, "Interaction-aware graph neural networks for fault diagnosis of complex industrial processes," *IEEE Transactions on neural networks and learning systems*, vol. 34, no. 9, pp. 6015–6028, 2021.
- [214] R. Liu, Q. Zhang, D. Lin, W. Zhang, and S. X. Ding, "Causal intervention graph neural network for fault diagnosis of complex industrial processes," *Reliability Engineering & System Safety*, vol. 251, p. 110 328, 2024.
- [215] Y. Xu, C. Zhang, Y. Luo, W. Ke, Q.-X. Zhu, Y.-L. He, Y. Zhang, and M.-Q. Zhang, "Multi-attention spatio-temporal fusion graph neural network for chemical process fault diagnosis," *IEEE Transactions on Instrumentation and Measurement*, 2025.
- [216] G. Jiang, K. Shen, X. Liu, X. Cheng, and P. Xie, "Hierarchical spatiotemporal graph network for fault diagnosis of industrial processes," *IEEE Internet of Things Journal*, 2024.
- [217] Z. Wang, Z. Wu, X. Li, H. Shao, T. Han, and M. Xie, "Attention-aware temporal-spatial graph neural network with multi-sensor information fusion for fault diagnosis," *Knowledge-Based Systems*, vol. 278, p. 110 891, 2023.
- [218] J. Xu, S. Mo, Z. Jiang, Z. Chen, W. Gui, and H. Wang, "A novel positive-negative graph convolutional network-based fault diagnosis method with application to complex systems," *Neurocomputing*, vol. 600, p. 128 145, 2024.
- [219] W. Cai, D. Zhao, and T. Wang, "Multi-scale dynamic graph mutual information network for planet bearing health monitoring under imbalanced data," *Advanced Engineering Informatics*, vol. 64, p. 103 096, 2025.
- [220] T. Li, Z. Zhou, S. Li, C. Sun, R. Yan, and X. Chen, "The emerging graph neural networks for intelligent fault diagnostics and prognostics: A guideline and a benchmark study," *Mechanical Systems and Signal Processing*, vol. 168, p. 108 653, 2022.
- [221] Z. Zhang and L. Wu, "Graph neural network-based bearing fault diagnosis using granger causality test," *Expert Systems with Applications*, vol. 242, p. 122 827, 2024.
- [222] Z. Chen, J. Xu, T. Peng, and C. Yang, "Graph convolutional networkbased method for fault diagnosis using a hybrid of measurement and prior knowledge," *IEEE transactions on cybernetics*, vol. 52, no. 9, pp. 9157–9169, 2021.
- [223] Z. Yu, C. Zhang, and C. Deng, "An improved gnn using dynamic graph embedding mechanism: A novel end-to-end framework for rolling bearing fault diagnosis under variable working conditions," *Mechanical Systems and Signal Processing*, vol. 200, p. 110 534, 2023.
- [224] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.
- [225] T. Kipf, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

Anomaly and Fault Detection and Predictive Maintenance: a Review on Industrial Applications

- [226] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in neural information processing systems*, vol. 30, 2017.
- [227] C. Li, L. Mo, and R. Yan, "Fault diagnosis of rolling bearing based on whvg and gcn," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–11, 2021.
- [228] C. Yang, K. Zhou, and J. Liu, "Supergraph: Spatial-temporal graph-based feature extraction for rotating machinery diagnosis," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 4, pp. 4167–4176, 2021.
- [229] B. Zhou, X. Li, T. Liu, K. Xu, W. Liu, and J. Bao, "Causalkgpt: Industrial structure causal knowledge-enhanced large language model for cause analysis of quality problems in aerospace product manufacturing," *Advanced Engineering Informatics*, vol. 59, p. 102 333, 2024.
- [230] S. Marandi, Y.-S. Hu, and M. Modarres, "Complex system diagnostics using a knowledge graph-informed and large language model-enhanced framework," *Applied Sciences*, vol. 15, no. 17, p. 9428, 2025.
- [231] C. Xiao, X. Liu, A. Wulamu, and D. Zhang, "Kg-sr-llm: Knowledge-guided semantic representation and large language model framework for cross-domain bearing fault diagnosis," *Sensors*, vol. 25, no. 18, p. 5758, 2025.
- [232] L. Tao, H. Liu, G. Ning, W. Cao, B. Huang, and C. Lu, "Llm-based framework for bearing fault diagnosis," *Mechanical Systems and Signal Processing*, vol. 224, p. 112 127, 2025.
- [233] J. Wang, T. Li, Y. Yang, S. Chen, and W. Zhai, "Diagllm: multimodal reasoning with large language model for explainable bearing fault diagnosis," *Science China Information Sciences*, vol. 68, no. 6, p. 160 103, 2025.
- [234] L. Lin, S. Zhang, S. Fu, and Y. Liu, "Fd-llm: Large language model for fault diagnosis of complex equipment," *Advanced Engineering Informatics*, vol. 65, p. 103 208, 2025.
- [235] S. Zheng, K. Pan, J. Liu, and Y. Chen, "Empirical study on finetuning pre-trained large language models for fault diagnosis of complex systems," *Reliability Engineering & System Safety*, vol. 252, p. 110 382, 2024.
- [236] X. Liao, C. Chen, Z. Wang, Y. Liu, T. Wang, and L. Cheng, "Large language model assisted fine-grained knowledge graph construction for robotic fault diagnosis," *Advanced Engineering Informatics*, vol. 65, p. 103 134, 2025.
- [237] Q. Zhang, C. Xu, J. Li, Y. Sun, J. Bao, and D. Zhang, "Llm-tsfd: An industrial time series human-in-the-loop fault diagnosis method based on a large language model," *Expert Systems with Applications*, vol. 264, p. 125 861, 2025.
- [238] X. Chen, Y. Lei, Y. Li, S. Parkinson, X. Li, J. Liu, F. Lu, H. Wang, Z. Wang, B. Yang et al., "Large models for machine monitoring and fault diagnostics: Opportunities, challenges, and future direction," *Journal of Dynamics, Monitoring and Diagnostics*, vol. 4, no. 2, pp. 76–90, 2025.
- [239] M. W. Hopwood, J. S. Stein, J. L. Braid, and H. P. Seigneur, "Physics-based method for generating fully synthetic iv curve training datasets for machine learning classification of pv failures," *Energies*, vol. 15, no. 14, p. 5085, 2022.
- [240] P. Seventekidis, D. Giagopoulos, A. Arailopoulos, and O. Markogiannaki, "Structural health monitoring using deep learning with optimal finite element model generated data," *Mechanical Systems and Signal Processing*, vol. 145, p. 106972, 2020.
- [241] W. Hao, L. Tan, X. Yang, D. Shi, M. Wang, G. Miao, and Y. Fan, "A physics-informed machine learning approach for notch fatigue evaluation of alloys used in aerospace," *International Journal of Fatigue*, vol. 170, p. 107 536, 2023.
- [242] Y. Wu, B. Sicard, and S. A. Gadsden, "Physics-informed machine learning: A comprehensive review on applications in anomaly detection and condition monitoring," *Expert Systems with Applications*, vol. 255, p. 124 678, 2024.
- [243] Z. Wang, Z. Zhou, W. Xu, C. Sun, and R. Yan, "Physics informed neural networks for fault severity identification of axial piston pumps," *Journal of Manufacturing Systems*, vol. 71, pp. 421–437, 2023.
- [244] C. Lai, I. Ahmed, E. Zio, W. Li, Y. Zhang, W. Yao, and J. Chen, "A multistage physics-informed neural network for fault detection in regulating valves of nuclear power plants," *Energies*, vol. 17, no. 11, p. 2647, 2024.
- [245] M. Parziale, L. Lomazzi, M. Giglio, and F. Cadini, "Physics-informed neural networks for the condition monitoring of rotating shafts," *Sensors*, vol. 24, no. 1, p. 207, 2023.
- [246] J. Shen, J. Chowdhury, S. Banerjee, and G. Terejanu, "Machine fault classification using hamiltonian neural networks," *arXiv preprint arXiv:2301.02243*, 2023.
- [247] S. Shen, H. Lu, M. Sadoughi, C. Hu, V. Nemani, A. Thelen, K. Webster, M. Darr, J. Sidon, and S. Kenny, "A physics-informed deep learning approach for bearing fault detection," *Engineering Applications of Artificial Intelligence*, vol. 103, p. 104 295, 2021.
- [248] Z. Xu, K. Zhao, J. Wang, and M. Bashir, "Physics-informed probabilistic deep network with interpretable mechanism for trustworthy mechanical fault diagnosis," *Adv. Eng. Informatics*, vol. 62, p. 102 806, 2024. [Online]. Available: <https://api.semanticscholar.org/CorpusID:272638356>
- [249] C. Dong, J. Tao, H. Sun, Q. Wei, H. Tan, and C. Liu, "Innovative fault diagnosis for axial piston pumps: A physics-informed neural network framework predicting pump flow ripple," *Mechanical Systems and Signal Processing*, vol. 225, p. 112 274, 2025.
- [250] C. Gao, Z. Wang, Y. Guo, H. Wang, and H. Yi, "Mpinet: multiscale physics-informed network for bearing fault diagnosis with small samples," *IEEE Transactions on Industrial Informatics*, 2024.
- [251] C. He, H. Shi, J. Si, and J. Li, "Physics-informed interpretable wavelet weight initialization and balanced dynamic adaptive threshold for intelligent fault diagnosis of rolling bearings," *Journal of Manufacturing Systems*, vol. 70, pp. 579–592, 2023.
- [252] Z. Shang, Z. Zhao, and R. Yan, "Denoising fault-aware wavelet network: A signal processing informed neural network for fault diagnosis," *Chinese Journal of Mechanical Engineering*, vol. 36, no. 1, p. 9, 2023.
- [253] C. He, H. Shi, X. Liu, and J. Li, "Interpretable physics-informed domain adaptation paradigm for cross-machine transfer diagnosis," *Knowledge-Based Systems*, vol. 288, p. 111 499, 2024.
- [254] T. Gao, H. Zhu, J. Wu, Q. Yuan, and Z. Li, "A fully interpretable convolutional neural network for intelligent fault diagnosis of rotating machinery," *Applied Soft Computing*, vol. 180, p. 113 472, 2025.
- [255] T. Li, Z. Zhao, C. Sun, L. Cheng, X. Chen, R. Yan, and R. X. Gao, "Waveletkernelnet: An interpretable deep neural network for industrial intelligent diagnosis," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 4, pp. 2302–2312, 2021.
- [256] Q. Ni, J. Ji, B. Halkon, K. Feng, and A. K. Nandi, "Physics-informed residual network (piresnet) for rolling element bearing fault diagnostics," *Mechanical Systems and Signal Processing*, vol. 200, p. 110544, 2023.
- [257] H. Lu, V. P. Nemani, V. Barzegar, C. Allen, C. Hu, S. Laflamme, S. Sarkar, and A. T. Zimmerman, "A physics-informed feature weighting method for bearing fault diagnostics," *Mechanical Systems and Signal Processing*, vol. 191, p. 110 171, 2023.
- [258] Y. Chen, M. Rao, K. Feng, and M. J. Zuo, "Physics-informed lstm hyperparameters selection for gearbox fault detection," *Mechanical Systems and Signal Processing*, vol. 171, p. 108 907, 2022.
- [259] Y. Ding, L. Ma, J. Ma, M. Suo, L. Tao, Y. Cheng, and C. Lu, "Intelligent fault diagnosis for rotating machinery using deep q-network based health state classification: A deep reinforcement learning approach," *Advanced Engineering Informatics*, vol. 42, p. 100 977, 2019.
- [260] H. Wang, J. Xu, C. Sun, R. Yan, and X. Chen, "Intelligent fault diagnosis for planetary gearbox using time-frequency representation and deep reinforcement learning," *IEEE/ASME Transactions On Mechatronics*, vol. 27, no. 2, pp. 985–998, 2021.
- [261] G. Qian and J. Liu, "Development of deep reinforcement learning-based fault diagnosis method for rotating machinery in nuclear power plants," *Progress in Nuclear Energy*, vol. 152, p. 104 401, 2022.
- [262] R. Wang, H. Jiang, K. Zhu, Y. Wang, and C. Liu, "A deep feature-enhanced reinforcement learning method for rolling bearing fault diagnosis," *Advanced Engineering Informatics*, vol. 54, p. 101 750, 2022.
- [263] Z. Li, H. Jiang, and X. Wang, "A novel reinforcement learning agent for rotating machinery fault diagnosis with data augmentation," *Reliability Engineering & System Safety*, vol. 253, p. 110 570, 2025.
- [264] R. Wang, H. Jiang, X. Li, and S. Liu, "A reinforcement neural architecture search method for rolling bearing fault diagnosis," *Measurement*, vol. 154, p. 107 417, 2020.
- [265] L. Wen, X. Li, and L. Gao, "A new reinforcement learning based learning rate scheduler for convolutional neural network in fault classification," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 12, pp. 12 890–12 900, 2020.
- [266] X. Zhang, S. Fan, and Z. Song, "Reinforcement learning-based consensative classifier for imbalanced fault classification," *Science China Information Sciences*, vol. 66, no. 11, p. 212 201, 2023.
- [267] S. Fan, X. Zhang, and Z. Song, "Imbalanced sample selection with deep reinforcement learning for fault diagnosis," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 4, pp. 2518–2527, 2021.

- [268] K. Yan, C. Lu, X. Ma, Z. Ji, and J. Huang, "Intelligent fault diagnosis for air handling units based on improved generative adversarial network and deep reinforcement learning," *Expert Systems with Applications*, vol. 240, p. 122 545, 2024.
- [269] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.
- [270] J. Tian, Y. Jiang, J. Zhang, H. Luo, and S. Yin, "A novel data augmentation approach to fault diagnosis with class-imbalance problem," *Reliability Engineering & System Safety*, vol. 243, p. 109 832, 2024.
- [271] Y. Liu, H. Jiang, C. Liu, W. Yang, and W. Sun, "Data-augmented wavelet capsule generative adversarial network for rolling bearing fault diagnosis," *Knowledge-Based Systems*, vol. 252, p. 109 439, 2022.
- [272] H. Gao, X. Zhang, X. Gao, F. Li, and H. Han, "Icot-gan: Integrated convolutional transformer gan for rolling bearings fault diagnosis under limited data condition," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–14, 2023.
- [273] S. Dahdal, L. Colombi, M. Brina, A. Gilli, M. Tortonesi, M. Vignoli, and C. Stefanelli, "An MLOps framework for GAN-based fault detection in Bonfiglioli's EVO plant," *Infocommunications Journal*, vol. 16, no. 2, pp. 2–10, 2024.
- [274] C. Hegedűs and P. Varga, "Tailoring MLOps techniques for industry 5.0 needs," in *2023 19th International Conference on Network and Service Management (CNSM)*, 2023, pp. 1–7.
- [275] I. Misbah, C. K. Lee, and K. L. Keung, "Fault diagnosis in rotating machines based on transfer learning: Literature review," *Knowledge-Based Systems*, vol. 283, p. 111 158, 2024.
- [276] L. Wen, X. Li, and L. Gao, "A transfer convolutional neural network for fault diagnosis based on resnet-50," *Neural Computing and Applications*, vol. 32, no. 10, pp. 6111–6124, 2020.
- [277] J. Caçao, J. Santos, and M. Antunes, "Explainable ai for industrial fault diagnosis: A systematic review," *Journal of Industrial Information Integration*, p. 100 905, 2025.
- [278] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf
- [279] F. Kakavandi, P. Han, R. D. Reus, P. G. Larsen, and H. Zhang, "Interpretable fault detection approach with deep neural networks to industrial applications," *2023 International Conference on Control, Automation and Diagnosis (ICCAD)*, pp. 1–7, 2023.
- [280] H. Tang, H. Wang, and C. Li, "Explainable fault diagnosis method based on statistical features for gearboxes," *Eng. Appl. Artif. Intell.*, vol. 148, p. 110 503, 2025.
- [281] M. J. Hasan, M. Sohaib, and J.-M. Kim, "An explainable ai-based fault diagnosis model for bearings," *Sensors*, vol. 21, no. 12, p. 4070, 2021.
- [282] M. Baek and S. B. Kim, "Failure detection and primary cause identification of multivariate time series data in semiconductor equipment," *IEEE Access*, vol. 11, pp. 54 363–54 372, 2023.
- [283] J. Hu, Y. Zhang, W. Li, X. Zheng, and Z. Tian, "Trustworthy artificial intelligence based on an explicable temporal feature network for industrial fault diagnosis," *Cognitive Computation*, vol. 16, no. 2, pp. 534–545, 2024.
- [284] A. N. Gummadi, J. C. Napier, and M. Abdallah, "Xai-iot: an explainable ai framework for enhancing anomaly detection in iot systems," *IEEE Access*, vol. 12, pp. 71 024–71 054, 2024.
- [285] M. Keshk, N. Koroniotis, N. Pham, N. Moustafa, B. Turnbull, and A. Y. Zomaya, "An explainable deep learning-enabled intrusion detection framework in iot networks," *Information Sciences*, vol. 639, p. 119 000, 2023.
- [286] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626.
- [287] H.-Y. Chen and C.-H. Lee, "Vibration signals analysis by explainable artificial intelligence (xai) approach: Application on bearing faults diagnosis," *IEEE Access*, vol. 8, pp. 134 246–134 256, 2020.
- [288] L. Guo, X. Gu, Y. Yu, A. Duan, and H. Gao, "An analysis method for interpretability of convolutional neural network in bearing fault diagnosis," *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1–12, 2024.
- [289] M. S. Kim, J. P. Yun, and P. Park, "An explainable convolutional neural network for fault diagnosis in linear motion guide," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 6, pp. 4036–4045, 2020.
- [290] L. C. Brito, G. A. Susto, J. N. Brito, and M. A. V. Duarte, "Fault diagnosis using explainable ai: A transfer learning-based approach for rotating machinery exploiting augmented synthetic data," *Expert Systems with Applications*, vol. 232, p. 120 860, 2023.
- [291] M. Gwak, M. S. Kim, J. P. Yun, and P. Park, "Robust and explainable fault diagnosis with power-perturbation-based decision boundary analysis of deep learning models," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 5, pp. 6982–6992, 2022.
- [292] M. S. Kim, J. P. Yun, and P. Park, "Deep learning-based explainable fault diagnosis model with an individually grouped 1-d convolution for three-axis vibration signals," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 12, pp. 8807–8817, 2022.
- [293] J. Lee, I. Noh, J. Lee, and S. W. Lee, "Development of an explainable fault diagnosis framework based on sensor data imification: A case study of the robotic spot-welding process," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 10, pp. 6895–6904, 2021.
- [294] S. Li, T. Li, C. Sun, R. Yan, and X. Chen, "Multilayer grad-cam: An effective tool towards explainable deep neural networks for intelligent fault diagnosis," *Journal of manufacturing systems*, vol. 69, pp. 20–30, 2023.
- [295] G. Li, Q. Yao, C. Fan, C. Zhou, G. Wu, Z. Zhou, and X. Fang, "An explainable one-dimensional convolutional neural networks based fault diagnosis method for building heating, ventilation and air conditioning systems," *Building and Environment*, vol. 203, p. 108 057, 2021.
- [296] A. Binder, G. Montavon, S. Lapuschkin, K.-R. Müller, and W. Samek, "Layer-wise relevance propagation for neural networks with local renormalization layers," in *International conference on artificial neural networks*. Springer, 2016, pp. 63–71.
- [297] J. Grezma, P. Wang, C. Sun, and R. X. Gao, "Explainable convolutional neural network for gearbox fault diagnosis," *Procedia CIRP*, vol. 80, pp. 476–481, 2019.
- [298] P. Agarwal, M. Tamer, and H. Budman, "Explainability: Relevance based dynamic deep learning algorithm for fault detection and diagnosis in chemical processes," *Computers & Chemical Engineering*, vol. 154, p. 107 467, 2021.
- [299] Y. Gao, S. Miyata, and Y. Akashi, "How to improve the application potential of deep learning model in hvac fault diagnosis: Based on pruning and interpretable deep learning method," *Applied Energy*, vol. 348, p. 121591, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0306261923009558>
- [300] P. Peng, Y. Zhang, H. Wang, and H. Zhang, "Towards robust and understandable fault detection and diagnosis using denoising sparse autoencoder and smooth integrated gradients," *ISA transactions*, vol. 125, pp. 371–383, 2022.
- [301] J. Du, X. Li, Y. Gao, and L. Gao, "Integrated gradient-based continuous wavelet transform for bearing fault diagnosis," *Sensors*, vol. 22, no. 22, p. 8760, 2022.
- [302] M. R. Santos, A. Guedes, and I. Sanchez-Gendriz, "Shapley additive explanations (shap) for efficient feature selection in rolling bearing fault diagnosis," *Machine Learning and Knowledge Extraction*, vol. 6, no. 1, pp. 316–341, 2024.
- [303] H. Pu, K. Zhang, and Y. An, "Restricted sparse networks for rolling bearing fault diagnosis," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 11, pp. 11 139–11 149, 2023.
- [304] T. Yan, D. Wang, and Y. Wang, "Discrimination-and sparsity-driven weight-oriented optimization model for interpretable initial fault detection and fault diagnosis," *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1–13, 2023.
- [305] G. Chen, J. Yuan, Y. Zhang, H. Zhu, R. Huang, F. Wang, and W. Li, "Enhancing reliability through interpretability: A comprehensive survey of interpretable intelligent fault diagnosis in rotating machinery," *IEEE Access*, 2024.
- [306] L. Ma, Y. Ding, Z. Wang, C. Wang, J. Ma, and C. Lu, "An interpretable data augmentation scheme for machine fault diagnosis based on a sparsity-constrained generative adversarial network," *Expert Systems with Applications*, vol. 182, p. 115 234, 2021.
- [307] R. Tian, M. Cui, and G. Chen, "A neural-symbolic network for interpretable fault diagnosis of rolling element bearings based on temporal logic," *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1–14, 2024.

- [308] G. Chen, M. Liu, and J. Chen, "Frequency-temporal-logic-based bearing fault diagnosis and fault interpretation using bayesian optimization with bayesian neural networks," *Mechanical Systems and Signal Processing*, vol. 145, p. 106 951, 2020.
- [309] G. Chen, P. Wei, H. Jiang, and M. Liu, "Formal language generation for fault diagnosis with spectral logic via adversarial training," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 1, pp. 119–129, 2020.
- [310] G. Chen, P. Lu, and Y. Tang, "Interpretable fault diagnosis of rolling element bearings with temporal logic neural network," in *2024 18th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. IEEE, 2024, pp. 290–296.
- [311] G. Chen, Y. Lu, and R. Su, "Interpretable fault diagnosis with shapelet temporal logic: Theory and application," *Automatica*, vol. 142, p. 110 350, 2022.
- [312] F. Rao, M. Zeng, and Y. Cheng, "A novel interpretable model via algorithm unrolling for intelligent fault diagnosis of machinery," *IEEE Sensors Journal*, vol. 24, no. 1, pp. 495–505, 2023.
- [313] Z. Zhao, T. Li, B. An, S. Wang, B. Ding, R. Yan, and X. Chen, "Model-driven deep unrolling: Towards interpretable deep learning against noise attacks for intelligent fault diagnosis," *ISA transactions*, vol. 129, pp. 644–662, 2022.
- [314] B. An, S. Wang, Z. Zhao, F. Qin, R. Yan, and X. Chen, "Interpretable neural network via algorithm unrolling for mechanical fault diagnosis," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–11, 2022.
- [315] Y. Li, Z. Zhou, C. Sun, X. Chen, and R. Yan, "Variational attention-based interpretable transformer network for rotary machine fault diagnosis," *IEEE transactions on neural networks and learning systems*, vol. 35, no. 5, pp. 6180–6193, 2022.
- [316] D. Liu, L. Cui, G. Wang, and W. Cheng, "Interpretable domain adaptation transformer: a transfer learning method for fault diagnosis of rotating machinery," *Structural Health Monitoring*, vol. 24, no. 2, pp. 1187–1200, 2025.
- [317] D. Chen, Z. Xie, R. Liu, W. Yu, Q. Hu, X. Li, and S. X. Ding, "Bayesian hierarchical graph neural networks with uncertainty feedback for trustworthy fault diagnosis of industrial processes," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.
- [318] R. Mishra, A. Choudhary, S. Fatima, A. Mohanty, and B. Panigrahi, "A systematic review on advancement and challenges in multi-fault diagnosis of rotating machines," *Engineering Applications of Artificial Intelligence*, vol. 156, p. 111 306, 2025.
- [319] Y. Zhou and W. Sun, "Tool wear condition monitoring in milling process based on current sensors," *IEEE access*, vol. 8, pp. 95 491–95 502, 2020.
- [320] P. Twardowski, M. Tabaszewski, M. Wiciak-Pikuła, and A. Felusiak-Czyrca, "Identification of tool wear using acoustic emission signal and machine learning methods," *Precision Engineering*, vol. 72, pp. 738–744, 2021.
- [321] H. Miao, Z. Zhao, C. Sun, B. Li, and R. Yan, "A u-net-based approach for tool wear area detection and identification," *IEEE Transactions on instrumentation and measurement*, vol. 70, pp. 1–10, 2020.
- [322] C.-W. Hung, C.-H. Lee, C.-C. Kuo, and S.-X. Zeng, "Soc-based early failure detection system using deep learning for tool wear," *Ieee Access*, vol. 10, pp. 70 491–70 501, 2022.
- [323] Y. Zhou, G. Zhi, W. Chen, Q. Qian, D. He, B. Sun, and W. Sun, "A new tool wear condition monitoring method based on deep learning under small samples," *Measurement*, vol. 189, p. 110 622, 2022.
- [324] R. Munaro, A. Attanasio, and A. Del Prete, "Tool wear monitoring with artificial intelligence methods: a review," *Journal of Manufacturing and Materials Processing*, vol. 7, no. 4, p. 129, 2023.
- [325] M. Cheng, L. Jiao, P. Yan, H. Jiang, R. Wang, T. Qiu, and X. Wang, "Intelligent tool wear monitoring and multi-step prediction based on deep learning model," *Journal of Manufacturing Systems*, vol. 62, pp. 286–300, 2022.
- [326] Z. Huang, J. Zhu, J. Lei, X. Li, and F. Tian, "Tool wear predicting based on multi-domain feature fusion by deep convolutional neural network in milling operations," *Journal of intelligent manufacturing*, vol. 31, no. 4, pp. 953–966, 2020.
- [327] Q. Wang, H. Wang, L. Hou, and S. Yi, "Overview of tool wear monitoring methods based on convolutional neural network," *Applied Sciences*, vol. 11, no. 24, p. 12041, 2021.
- [328] Y. Zhao, A. Jiang, W. Jiang, Y. Zhang, and S. Zhang, "Hydraulic pump fault diagnosis by modified slime mold algorithm optimized support vector machine," *Measurement*, vol. 254, p. 117 878, 2025.
- [329] D. Ma, Z. Liu, Q. Gao, and Y. Ding, "Fault diagnosis of multi-step electromagnetic hydraulic valve group based on localized current signal cs-svm," *Measurement*, vol. 245, p. 116 632, 2025.
- [330] N. Liang, F. Zhang, Z. Yuan, H. Wang, J. Zhang, Z. Fan, and X. Yu, "Fault diagnosis in hydraulic systems via multi-channel multi-modal fusion," *Measurement Science and Technology*, vol. 36, no. 5, p. 055 023, 2025.
- [331] R. Jiang, Z. Yuan, H. Wang, Z. Fan, Y. Zhang, N. Liang, and X. Yu, "Mcfpred: A novel multichannel signal adaptive fusion framework for fault diagnosis in hydraulic systems," *IEEE Transactions on Instrumentation and Measurement*, 2025.
- [332] S. Liu, J. Yin, Y. Zhang, and P. Liang, "Mlift: Multi-scale linear interaction fusion transformer for fault diagnosis of hydraulic pumps," *Measurement*, p. 117892, 2025.
- [333] G. Zhang, K. Zhang, W. Shen, G. Xu, and Y.-J. Lin, "Intelligent fault diagnosis of multi-way directional valves in hydraulic systems using digital twin and deep learning approaches," *Mechanical Systems and Signal Processing*, vol. 230, p. 112 579, 2025.
- [334] S. Tang, Z. Yan, Y. Xue, K. M. Lim, and Y. Zhu, "Deep transferable model with multi-head attention for small sample fault diagnosis of hydraulic pumps," *Engineering Applications of Artificial Intelligence*, vol. 160, p. 111 808, 2025.
- [335] K.-J. Shih, M.-F. Hsieh, B.-J. Chen, and S.-F. Huang, "Machine learning for inter-turn short-circuit fault diagnosis in permanent magnet synchronous motors," *IEEE Transactions on Magnetics*, vol. 58, no. 8, pp. 1–7, 2022.
- [336] C. Zachariades and V. Xavier, "A review of artificial intelligence techniques in fault diagnosis of electric machines," *Sensors (Basel, Switzerland)*, vol. 25, no. 16, p. 5128, 2025.
- [337] P. Pietrzak, M. Wolkiewicz, and T. Orlowska-Kowalska, "Pmsm stator winding fault detection and classification based on bispectrum analysis and convolutional neural network," *IEEE Transactions on Industrial Electronics*, vol. 70, no. 5, pp. 5192–5202, 2022.
- [338] F. Husari and J. Seshadrinath, "Incipient interturn fault detection and severity evaluation in electric drive system using hybrid hcn-svm based model," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 3, pp. 1823–1832, 2021.
- [339] M.-C. Kim, J.-H. Lee, D.-H. Wang, and I.-S. Lee, "Induction motor fault diagnosis using support vector machine, neural networks, and boosting methods," *Sensors*, vol. 23, no. 5, p. 2585, 2023.
- [340] M.-Q. Tran, M.-K. Liu, Q.-V. Tran, and T.-K. Nguyen, "Effective fault diagnosis based on wavelet and convolutional attention neural network for induction motors," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–13, 2021.
- [341] R. F. R. Junior, I. A. dos Santos Areias, M. M. Campos, C. E. Teixeira, L. E. B. da Silva, and G. F. Gomes, "Fault detection and diagnosis in electric motors using 1d convolutional neural networks with multi-channel vibration signals," *Measurement*, vol. 190, p. 110 759, 2022.
- [342] A. Choudhary, R. K. Mishra, S. Fatima, and B. K. Panigrahi, "Multinput cnn based vibro-acoustic fusion for accurate fault diagnosis of induction motor," *Engineering Applications of Artificial Intelligence*, vol. 120, p. 105 872, 2023.
- [343] Y. Tang, X. Zhang, G. Qin, Z. Long, S. Huang, D. Song, and H. Shao, "Graph cardinality preserved attention network for fault diagnosis of induction motor under varying speed and load condition," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 6, pp. 3702–3712, 2021.
- [344] W. Lang, Y. Hu, C. Gong, X. Zhang, H. Xu, and J. Deng, "Artificial intelligence-based technique for fault detection and diagnosis of ev motors: A review," *IEEE Transactions on Transportation Electrification*, vol. 8, no. 1, pp. 384–406, 2021.
- [345] T.-T. Vo, M.-K. Liu, and M.-Q. Tran, "Harnessing attention mechanisms in a comprehensive deep learning approach for induction motor fault diagnosis using raw electrical signals," *Engineering Applications of Artificial Intelligence*, vol. 129, p. 107643, 2024.
- [346] A. A. Nuhu, Q. Zeeshan, B. Safaei, and M. A. Shahzad, "Machine learning-based techniques for fault diagnosis in the semiconductor manufacturing process: a comparative study," *The Journal of Supercomputing*, vol. 79, no. 2, pp. 2031–2081, 2023.
- [347] C.-Y. Hsu and W.-C. Liu, "Multiple time-series convolutional neural network for fault detection and diagnosis and empirical study in semiconductor manufacturing," *Journal of Intelligent Manufacturing*, vol. 32, no. 3, pp. 823–836, 2021.
- [348] S. Lou, C. Yang, P. Wu, L. Kong, and Y. Xu, "Fault diagnosis of blast furnace iron-making process with a novel deep stationary kernel learning support vector machine approach," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–13, 2022.

- [349] H. Nandanwar and R. Katarya, "Deep learning enabled intrusion detection system for industrial iot environment," *Expert Systems with Applications*, vol. 249, p. 123 808, 2024.
- [350] M. Mohy-Eddine, A. Guezaz, S. Benkirane, M. Azrou, and Y. Farhaoui, "An ensemble learning based intrusion detection model for industrial iot security," *Big Data Mining and Analytics*, vol. 6, no. 3, pp. 273–287, 2023.
- [351] S. Soliman, W. Oudah, and A. Aljuhani, "Deep learning-based intrusion detection approach for securing industrial internet of things," *Alexandria Engineering Journal*, vol. 81, pp. 371–383, 2023.
- [352] B. Tang, Y. Lu, Q. Li, Y. Bai, J. Yu, and X. Yu, "A diffusion model based on network intrusion detection method for industrial cyberphysical systems," *Sensors*, vol. 23, no. 3, p. 1141, 2023.
- [353] S. T. Mehedi, A. Anwar, Z. Rahman, K. Ahmed, and R. Islam, "Dependable intrusion detection system for iot: A deep transfer learning based approach," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 1, pp. 1006–1017, 2022.
- [354] M.-Q. Tran, M. Amer, A. Y. Abdelaziz, H.-J. Dai, M.-K. Liu, M. Elsis, et al., "Robust fault recognition and correction scheme for induction motors using an effective iot with deep learning approach," *Measurement*, vol. 207, p. 112 398, 2023.
- [355] M. A. Alohal, F. N. Al-Wesabi, A. M. Hilal, S. Goel, D. Gupta, and A. Khanna, "Artificial intelligence enabled intrusion detection systems for cognitive cyber-physical systems in industry 4.0 environment," *Cognitive Neurodynamics*, vol. 16, no. 5, pp. 1045–1057, 2022.
- [356] A. Abusitta, G. H. de Carvalho, O. A. Wahab, T. Halabi, B. C. Fung, and S. Al Mamoori, "Deep learning-enabled anomaly detection for iot systems," *Internet of Things*, vol. 21, p. 100 656, 2023.
- [357] F. Khan, R. Alturki, M. A. Rahman, S. Mastorakis, I. Razzak, and S. T. Shah, "Trustworthy and reliable deep-learning-based cyberattack detection in industrial iot," *IEEE transactions on industrial informatics*, vol. 19, no. 1, pp. 1030–1038, 2022.
- [358] R. Canonico and G. Sperli, "Industrial cyber-physical systems protection: A methodological review," *Computers & Security*, vol. 135, p. 103 531, 2023.
- [359] Z. Zhang, G. Peng, W. Wang, and Y. Chen, "Real-time human fault detection in assembly tasks, based on human action prediction using a spatio-temporal learning model," *Sustainability*, vol. 14, no. 15, p. 9027, 2022.
- [360] A. M. Vukicevic, M. N. Petrovic, N. M. Knezevic, and K. M. Jovanovic, "Deep learning-based recognition of unsafe acts in manufacturing industry," *IEEE Access*, vol. 11, pp. 103 406–103 418, 2023.
- [361] L. Wang, J. Yuan, S. Zhou, and M. König, "Automatic detection of unsafe behaviors resulting from the improper use of ppe during human-robot collaboration," *Journal of Computing in Civil Engineering*, vol. 40, no. 2, p. 04 025 147, 2026.
- [362] C. Zhang, Y. Zhao, Y. Zhao, T. Li, and X. Zhang, "Causal discovery and inference-based fault detection and diagnosis method for heating, ventilation and air conditioning systems," *Building and Environment*, vol. 212, p. 108 760, 2022.
- [363] Z. Du, S. Chen, P. Li, K. Chen, X. Liang, X. Zhu, and X. Jin, "Knowledge-extracted deep learning diagnosis and its cloud-based management for multiple faults of chiller," *Building and Environment*, vol. 235, p. 110 228, 2023.
- [364] Z. Chen, Z. O'Neill, J. Wen, O. Pradhan, T. Yang, X. Lu, G. Lin, S. Miyata, S. Lee, C. Shen et al., "A review of data-driven fault detection and diagnostics for building hvac systems," *Applied Energy*, vol. 339, p. 121 030, 2023.
- [365] L. Eren, T. Ince, and S. Kiranyaz, "A generic intelligent bearing fault diagnosis system using compact adaptive 1d cnn classifier," *Journal of signal processing systems*, vol. 91, no. 2, pp. 179–189, 2019.
- [366] Q. Zhang and Z. Ju, "Rolling bearing fault diagnosis based on 2d cnn and hybrid kernel fuzzy svm," *Advanced Theory and Simulations*, p. 2 400 793, 2025.
- [367] K. S. Siraskar R. and P. S., "Reinforcement learning for predictive maintenance: A systematic technical review," *Artif Intell Rev*, vol. 56, p. 12885–12947, 2023. [Online]. Available: <https://doi.org/10.1007/s10462-023-10468-6>
- [368] M. L. Ruiz Rodríguez, S. Kubler, A. de Giorgio, M. Cordy, J. Robert, and Y. Le Traon, "Multi-agent deep reinforcement learning based predictive maintenance on parallel machines," *Robotics and Computer-Integrated Manufacturing*, vol. 78, p. 102 406, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0736584522000928>
- [369] J. Kairo, "Machine learning algorithms for predictive maintenance in manufacturing," *Journal of Technology and Systems*, vol. 6, pp. 66–79, 08 2024.
- [370] S. Vollert and A. Theissler, "Challenges of machine learning-based rul prognosis: A review on nasa's c-mapss data set," in *2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2021, pp. 1–8.
- [371] S. S. Sharma, V. V. and A. Malviya, "Ai-enhanced predictive maintenance in intelligent systems for industries," in *2024 International Conference on Advances in Computing Research on Science Engineering and Technology (ACROSET)*, 2024, pp. 1–6.
- [372] A. E. Frankó and P. Varga, "A survey on machine learning based smart maintenance and quality control solutions," *Infocommunications Journal*, vol. 13, no. 4, pp. 28–35, 2021.
- [373] F. Wang, A. Liu, C. Qu, R. Xiong, and L. Chen, "A deep-learning method for remaining useful life prediction of power machinery via dual-attention mechanism," *Sensors*, vol. 25, no. 2, 2025. [Online]. Available: <https://www.mdpi.com/1424-8220/25/2/497>
- [374] L. Cao, X. Wang, H. Zhang, Z. Meng, J. Li, and M. Liu, "A novel crossscenario transferable rul prediction network with multisource domain meta transfer learning for wind turbine bearings," *IEEE Transactions on Instrumentation and Measurement*, pp. 1–1, 2025.
- [375] X. Liu, Z. Zhang, Z. Li, J. Wang, Y. Zhu, and H. Ma, "Advancements in bearing health monitoring and remaining useful life prediction: Techniques, challenges, and future directions," *Measurement Science and Technology*, 2025. [Online]. Available: <http://iopscience.iop.org/article/10.1088/1361-6501/adafc8>
- [376] D. Li, Y.-F. Li, and H. Wang, "Next-gen aiot-driven predictive maintenance and health management system for high-speed train wheel and axle systems," in *2024 IEEE Global Conference on Artificial Intelligence and Internet of Things (GCAIoT)*, 2024, pp. 1–6.
- [377] P. Suebsuwong, "Integrating horizons: A holistic view of predictive maintenance in aviation maintenance practices," *Journal of Science and Technology Kasetsart University*, vol. 13, no. 3, pp. 38–60, 2024. [Online]. Available: <https://ph03.tci-thaijo.org/index.php/JSTKU/article/view/3495>
- [378] G. Bianchi, C. Fanelli, F. Freddi, F. Giuliani, and A. L. Placa, "Systematic review railway infrastructure monitoring: From classic techniques to predictive maintenance," *Advances in Mechanical Engineering*, vol. 17, no. 1, p. 16878132241285631, 2025. [Online]. Available: <https://doi.org/10.1177/16878132241285631>
- [379] G. Papa, Š. Poklukar, A. Frankó, A. Sillitti, A. Kancilija, M. Šterk, C. Hegedűs et al., "Improving the maintenance of railway switches through proactive approach," *Electronics*, vol. 9, no. 8, p. 1260, 2020.
- [380] C. Hegedűs, P. Ciancarini, A. Frankó, A. Kancilija, I. Moldován, G. Papa et al., "Proactive maintenance of railway switches," in *2018 5th International Conference on Control, Decision and Information Technologies (CoDIT)*, 2018.
- [381] S. R. Katta, "Optimizing pharmaceutical manufacturing through ai-powered calibration and maintenance workflow management," *Journal of Artificial Intelligence, Machine Learning and Data Science*, vol. 1, no. 4, pp. 2175–2181, 2023.
- [382] S. Gupta, J. Maiti, and A. Kumar, "Prognostic modeling of li-ion battery using convolutional long short term memory (convlstm)," *International Journal of System Assurance Engineering and Management*, 2025. [Online]. Available: <https://doi.org/10.1007/s13198-025-02706-7>
- [383] J. Heredia and E. Ayala, "IoT and AI-Based Predictive Maintenance System Design for Express Auto Repair Shops," *Revista Técnica energía*, vol. 21, pp. 81–86, 06 2025. [Online]. Available: http://scielo.senescyt.gob.ec/scielo.php?script=sci_arttext&pid=S2602-84922025000100081&nrm=iso
- [384] F. Battisti, A. V. B. da Silva, L. F. A. Pereira, T. B. A. de Carvalho, J. C. T. de Araujo, E. Choi, T. A. Nguyen, and D. Min, "hLSTMAging: A hybrid LSTM model for software aging forecast," *Applied Sciences*, vol. 12, no. 13, p. 6412, 2022.
- [385] C. Chen, H. Fu, Y. Zheng, F. Tao, and Y. Liu, "The advance of digital twin for predictive maintenance: The role and function of machine learning," *Journal of Manufacturing Systems*, vol. 71, pp. 581–594, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S027861252300211X>
- [386] M. Arias Chao, C. Kulkarni, K. Goebel, and O. Fink, "Fusing physicsbased and deep learning models for prognostics," *Reliability Engineering & System Safety*, vol. 217, p. 107 961, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0951832021004725>

Anomaly and Fault Detection and Predictive Maintenance: a Review on Industrial Applications

- [387] R. G. Nascimento and F. A. C. Viana, "Fleet prognosis with physics-informed recurrent neural networks," *ArXiv*, vol. abs/1901.05512, 2019. [Online]. Available: <https://api.semanticscholar.org/CorpusID:58014195>
- [388] Y. Yucesan and F. Viana, "Wind turbine main bearing fatigue life estimation with physics-informed neural networks," *Annual Conference of the PHM Society*, vol. 11, 09 2019.
- [389] A. Dourado and F. Viana, "Physics-informed neural networks for corrosion-fatigue prognosis," *Annual Conference of the PHM Society*, vol. 11, 09 2019.
- [390] N. Yang, W. Zhang, J. Zhang, K. Wang, Y. Su, and Y. Liu, "A method for remaining useful life prediction and uncertainty quantification of rolling bearings based on fault feature gain," *IEEE Transactions on Instrumentation and Measurement*, pp. 1–1, 2025.
- [391] R. Xiong, Y. He, Y. Sun, Y. Jia, and W. Shen, "Enhanced electrode-level diagnostics for lithium-ion battery degradation using physics-informed neural networks," *Journal of Energy Chemistry*, 2025. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2095495625000609>
- [392] Z. Lu, C. Guo, M. Liu, and R. Shi, "Remaining useful lifetime estimation for discrete power electronic devices using physics-informed neural network," *Scientific Reports*, vol. 13, no. 1, p. 101 67, jun 2023. [Online]. Available: <https://doi.org/10.1038/s41598-023-37154-5>
- [393] L. E. J. Wang, R. Yang, C. Wang, H. Li, and R. Xiong, "A physics-informed neural network-based method for predicting degradation trajectories and remaining useful life of supercapacitors," *Green Energy and Intelligent Transportation*, vol. 4, no. 3, p. 100 291, 2025. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2773153725000416>
- [394] X. Liao, S. Chen, P. Wen, and S. Zhao, "Remaining useful life with self-attention assisted physics-informed neural network," *Advanced Engineering Informatics*, vol. 58, p. 102 195, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1474034623003233>
- [395] S. Xie, W. Cheng, Z. Nie, Q. Huang, J. Xing, X. Chen, R. Zhang, and Y. Yang, "Bayesian physics-informed neural networks with iterative ensemble kalman inversion for rul prediction and uncertainty quantification," *Advanced Engineering Informatics*, vol. 69, p. 103 907, 2026. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1474034625008006>
- [396] Y. Zhou and M. Huang, "Lithium-ion batteries remaining useful life prediction based on a mixture of empirical mode decomposition and arima model," *Microelectronics Reliability*, vol. 65, pp. 265–273, 2016. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0026271416303018>
- [397] K. Javed, R. Gouriveau, N. Zerhouni, and P. Nectoux, "A feature extraction procedure based on trigonometric functions and cumulative descriptors to enhance prognostics modeling," in *2013 IEEE Conference on Prognostics and Health Management (PHM)*, 2013, pp. 1–7.
- [398] J. Coble and J. Baalis, "Merging data sources to predict remaining useful life – an automated method to identify prognostic parameters." PhD Diss, University of Tennessee, 2010. [Online]. Available: https://trace.tennessee.edu/utk_graddiss/683
- [399] M. Rigamonti, P. Baraldi, E. Zio, I. Roychoudhury, K. Goebel, and S. Poll, "Echo state network for the remaining useful life prediction of a turbofan engine," *PHM Society European Conference*, 2016. [Online]. Available: <https://doi.org/10.36001/phme.2016.v3i1.1623>
- [400] X. Zhu, P. Yu, and S. Su, "A deep transfer network based on dual-task learning for predicting the remaining useful life of rolling bearings," *Measurement Science and Technology*, vol. 36, no. 3, p. 036108, feb 2025. [Online]. Available: <https://dx.doi.org/10.1088/1361-6501/adafd2>
- [401] X. W. Pandit R, "Data-driven models for predicting remaining useful life of high-speed shaft bearings in wind turbines using vibration signal analysis and sparrow search algorithm," *Energy Science & Engineering*, 2023. [Online]. Available: <https://misclibrary.wiley.com/doi/10.1002/ese3.1075>
- [402] W. Zhang, D. Yang, and H. Wang, "Data-driven methods for predictive maintenance of industrial equipment: A survey," *IEEE Systems Journal*, vol. 13, no. 3, pp. 2213–2227, 2019.
- [403] Y. Liu, Y. Wang, Y. Wang, S. Xue, Z. Wang, and Z. Gao, "Method for predicting remaining useful life of rolling bearings based on dynamic complexity characteristic entropy and quantum neural networks," *Engineering Failure Analysis*, vol. 170, p. 109 315, 2025. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1350630725000561>
- [404] Y. Su, Z. Lei, G. Wen, and X. Chen, "A novel multi-sensor data fusion enabled health indicator construction and remaining useful life prediction of aero-engine," *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, vol. 0, no. 0, p. 09544054241310485, 0. [Online]. Available: <https://doi.org/10.1177/09544054241310485>
- [405] Y. Qin, J. Zhou, and D. Chen, "Unsupervised health indicator construction by a novel degradation-trend-constrained variational autoencoder and its applications," *IEEE/ASME Transactions on Mechatronics*, vol. PP, pp. 1–1, 07 2021.
- [406] M. Kim, S. Yoo, S. Son, S. Y. Chang, and K.-Y. Oh, "Physics-informed deep learning framework for explainable remaining useful life prediction," *Engineering Applications of Artificial Intelligence*, vol. 143, p. 110072, 2025. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0952197625000727>
- [407] L. Wen, G. Yang, L. Hu, C. Yang, and K. Feng, "A new unsupervised health index estimation method for bearings early fault detection based on gaussian mixture model," *Engineering Applications of Artificial Intelligence*, vol. 128, p. 107562, 2024. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0952197623017463>
- [408] T. Sun, C. Yin, H. Zheng, and Y. Dong, "An unsupervised framework for dynamic health indicator construction and its application in rolling bearing prognostics," 2025. [Online]. Available: <https://arxiv.org/abs/2506.05438>
- [409] P. Malhotra, V. TV, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, "Multi-sensor prognostics using an unsupervised health index based on lstm encoder-decoder," 2016. [Online]. Available: <https://arxiv.org/abs/1608.06154>
- [410] X. Ye, G. Li, L. Meng, and G. Lu, "Dynamic health index extraction for incipient bearing degradation detection," *ISA Transactions*, vol. 128, pp. 535–549, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0019057821006005>
- [411] C. Lim, S. Kim, Y.-H. Seo, and J.-H. Choi, "Feature extraction for bearing prognostics using weighted correlation of fault frequencies over cycles," *Structural Health Monitoring*, vol. 19, no. 6, pp. 1808–1820, 2020. [Online]. Available: <https://doi.org/10.1177/1475921719900917>
- [412] X. Zhang, P. Tengyi, S. Sun, and Y. Zhou, "New multifeature information health index (mihi) based on a quasi-orthogonal sparse algorithm for bearing degradation monitoring," *Computational Intelligence and Neuroscience*, vol. 2021, pp. 1–14, 08 2021.
- [413] D. An, J.-H. Choi, and N. H. Kim, "Remaining useful life prediction of rolling element bearings using degradation feature based on amplitude decrease at specific frequencies," *Structural Health Monitoring*, vol. 17, no. 5, pp. 1095–1109, 2018. [Online]. Available: <https://doi.org/10.1177/1475921717736226>
- [414] J. Park, S. Kim, J.-H. Choi, and S. H. Lee, "Frequency energy shift method for bearing fault prognosis using microphone sensor," *Mechanical Systems and Signal Processing*, vol. 147, p. 107 068, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0888327020304544>
- [415] X. Du, W. Jia, P. Yu, Y. Shi, and S. Cheng, "A remaining useful life prediction method based on time-frequency images of the mechanical vibration signals," *Measurement*, vol. 202, p. 111 782, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S026322412200985X>
- [416] X. Luo, Y. Song, W. Bu, H. Liang, and M. Zheng, "A joint prediction of the state of health and remaining useful life of lithium-ion batteries based on gaussian process regression and long short-term memory," *Processes*, vol. 13, no. 1, 2025. [Online]. Available: <https://www.mdpi.com/2227-9717/13/1/239>
- [417] M. Assafo and P. Langendoerfer, "Tool remaining useful life prediction using feature extraction and machine learning-based sensor fusion," *Results in Engineering*, vol. 28, p. 107 297, 2025. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2590123025033523>
- [418] I. Obisakin and E. C. Vanessa, "State of health estimation of lithium-ion batteries using support vector regression and long short-term memory," *Open Journal of Applied Sciences*, vol. 12, pp. 1366–1382, 2022.
- [419] M. Yenioglu, E. Ayçiçek, and O. Erdinc, "Remaining useful life estimation of lithium-ion batteries," in *2024 15th National Conference on Electrical and Electronics Engineering (ELECO)*, 2024, pp. 1–5.
- [420] L. Ren, J. Dong, X. Wang, Z. Meng, L. Zhao, and M. J. Deen, "A data-driven auto-cnn-lstm prediction model for lithium-ion battery remaining useful life," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 5, pp. 3478–3487, 2021.

- [421] A. Rastegarpanah, M. E. Asif, and R. Stolkin, "Hybrid neural networks for enhanced predictions of remaining useful life in lithium-ion batteries," *Batteries*, vol. 10, no. 3, 2024. [Online]. Available: <https://www.mdpi.com/2313-0105/10/3/106>
- [422] C. Wang, Z. Bao, H. Lin, Z. He, and M. Gao, "An exponential transformer for learning interpretable temporal information in remaining useful life prediction of lithium-ion battery," *IEEE Transactions on Transportation Electrification*, pp. 1–1, 2025.
- [423] L. Liu, J. Huang, H. Zhao, T. Li, and B. Li, "Patchformer: A novel patch-based transformer for accurate remaining useful life prediction of lithium-ion batteries," *Journal of Power Sources*, vol. 631, p. 236 187, 2025. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0378775325000230>
- [424] R. E and S. A., "Performance benchmarking and analysis of prognostic methods for cmappss datasets," *International Journal of Prognostics and Health Management*, 2020. [Online]. Available: <https://doi.org/10.36001/ijphm.2014.v5i2.2236>
- [425] F. O. Heimes, "Recurrent neural networks for remaining useful life estimation," in *2008 International Conference on Prognostics and Health Management*, 2008, pp. 1–6.
- [426] S. Zheng, K. Ristovski, A. Farahat, and C. Gupta, "Long short-term memory network for remaining useful life estimation," in *2017 IEEE International Conference on Prognostics and Health Management (ICPHM)*, 2017, pp. 88–95.
- [427] X. Li, Q. Ding, and J.-Q. Sun, "Remaining useful life estimation in prognostics using deep convolution neural networks," *Reliability Engineering and System Safety*, vol. 172, pp. 1–11, 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0951832017307779>
- [428] H. Li, W. Zhao, Y. Zhang, and E. Zio, "Remaining useful life prediction using multi-scale deep convolutional neural network," *Applied Soft Computing*, vol. 89, p. 106113, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1568494620300533>
- [429] R. Jin, M. Wu, K. Wu, K. Gao, Z. Chen, and X. Li, "Position encoding based convolutional neural networks for machine remaining useful life prediction," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 8, pp. 1427–1439, 2022.
- [430] J. Wang, G. Wen, S. Yang, and Y. Liu, "Remaining useful life estimation in prognostics using deep bidirectional lstm neural network," in *2018 Prognostics and System Health Management Conference (PHMChongqing)*, 2018, pp. 1037–1042.
- [431] S. Deng and J. Zhou, "Prediction of remaining useful life of aeroengines based on CNN-LSTM-Attention," *International Journal of Computational Intelligence Systems*, vol. 17, no. 1, 2024.
- [432] Y. He, C. Wen, and W. Xu, "Residual life prediction of saenn-bilstm aero-engine based on a multichannel hybrid network," *Applied Sciences*, vol. 15, no. 2, 2025. [Online]. Available: <https://www.mdpi.com/2076-3417/15/2/966>
- [433] J. Gao, Y. Wang, and Z. Sun, "An interpretable rul prediction method of aircraft engines under complex operating conditions using spatio-temporal features," *Measurement Science and Technology*, vol. 35, no. 7, p. 076003, apr 2024. [Online]. Available: <https://dx.doi.org/10.1088/1361-6501/ad3b2c>
- [434] B. Yang, R. Liu, and E. Zio, "Remaining useful life prediction based on a double-convolutional neural network architecture," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 12, pp. 9521–9530, 2019.
- [435] C. Zhang, P. Lim, A. K. Qin, and K. C. Tan, "Multiobjective deep belief networks ensemble for remaining useful life estimation in prognostics," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2306–2318, 2017.
- [436] A. Z. Ture, B. A. Akbulut, H. Ahmet et al., "Stacking-based ensemble learning for remaining useful life estimation," *Soft Computing*, vol. 28, no. 3, pp. 1337–1349, 2024.
- [437] Y. Liu, B. Xu, and Y.-a. Geng, "Multi-condition remaining useful life prediction based on mixture of encoders," *Entropy*, vol. 27, no. 1, 2025. [Online]. Available: <https://www.mdpi.com/1099-4300/27/1/79>
- [438] L. Jayasinghe, T. Samarasinghe, C. Yuen, J. C. N. Low, and S. S. Ge, "Temporal convolutional memory networks for remaining useful life estimation of industrial machinery," 2018. [Online]. Available: <https://arxiv.org/abs/1810.05644>
- [439] R. Li, Z. Chu, W. Jin, Y. Wang, and X. Hu, "Temporal convolutional network based regression approach for estimation of remaining useful life," in *2021 IEEE International Conference on Prognostics and Health Management (ICPHM)*, 2021, pp. 1–10.
- [440] J. Chen, D. Chen, and G. Liu, "Using temporal convolution network for remaining useful lifetime prediction," *Engineering Reports*, vol. 3, no. 3, p. e12305, 2021. [Online]. Available: <https://misclibrary.wiley.com/doi/abs/10.1002/eng2.12305>
- [441] F. Xiang, Y. Zhang, S. Zhang, Z. Wang, L. Qiu, and J.-H. Choi, "Bayesian gated-transformer model for risk-aware prediction of aeroengine remaining useful life," *Expert Systems with Applications*, vol. 238, p. 121859, 2024. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0957417423023618>
- [442] N. DeVol, C. Saldana, and K. Fu, "Inception based deep convolutional neural network for remaining useful life estimation of turbofan engines," *Annual Conference of the PHM Society*, 2021. [Online]. Available: <https://doi.org/10.36001/phmconf.2021.v13i1.3109>
- [443] T. Berghout, M.-D. Mouss, L.-H. Mouss, and M. Benbouzid, "Prognnet: A transferable deep network for aircraft engine damage propagation prognosis under real flight conditions," *Aerospace*, vol. 10, no. 1, 2023. [Online]. Available: <https://www.mdpi.com/2226-4310/10/1/10>
- [444] Z. Zhao and P. Zou, "Attention-based dual-channel deep neural network for aero-engine rul prediction under time-varying operating conditions," *Journal of Physics: Conference Series*, vol. 2386, no. 1, p. 012 027, dec 2022. [Online]. Available: <https://dx.doi.org/10.1088/1742-6596/2386/1/012027>
- [445] X. Li, J. Li, L. Zuo, L. Zhu, and H. T. Shen, "Domain adaptive remaining useful life prediction with transformer," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–13, 2022.
- [446] L. C. Brito, L. L. C. Araújo, G. A. Susto, J. N. Brito, and M. A. V. Duarte, "Rotating machinery health monitoring: Unsupervised anomaly detection and remaining useful life (RUL) prediction using explainable artificial intelligence (XAI) and statistical distribution," *IFACPapersOnLine*, vol. 59, no. 26, pp. 97–102, 2025.
- [447] Y. Chen, D. Zhang, and W. an Zhang, "Mswr-lrcn: A new deep learning approach to remaining useful life estimation of bearings," *Control Engineering Practice*, vol. 118, p. 104 969, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S096706612100246X>
- [448] K. Xia, Q. Li, L. Han, Z. Ren, and H. Luo, "Prediction of residual life of rolling bearings based on multi-scale improved temporal convolutional network (mitcn) model," *Machines*, vol. 13, no. 2, 2025. [Online]. Available: <https://www.mdpi.com/2075-1702/13/2/137>
- [449] T. Liu, Y. Li, Z. Hu, C. Fu, and G. Song, "Remaining useful life prediction of bearings based on small-sample enhanced and interpretable transfer learning," *Advanced Engineering Informatics*, vol. 69, p. 103 938, 2026. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1474034625008316>
- [450] A. Fentaye and K. Kyprianidis, "Gas turbine prognostics via temporal fusion transformer," *The Aeronautical Journal*, vol. 128, no. 1325, 2024.
- [451] B. Lim, S. O. Arık, N. Loeff, and T. Pfister, "Temporal fusion transformers for interpretable multi-horizon time series forecasting," *International Journal of Forecasting*, vol. 37, no. 4, pp. 1748–1764, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0169207021000637>
- [452] Y. Chang, F. Li, J. Chen, Y. Liu, and Z. Li, "Efficient temporal flow transformer accompanied with multi-head probparse self-attention mechanism for remaining useful life prognostics," *Reliability Engineering & System Safety*, vol. 226, p. 108 701, 10 2022.
- [453] Z. Chen, M. Wu, R. Zhao, F. Guretno, R. Yan, and X. Li, "Machine remaining useful life prediction via an attention-based deep learning approach," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 3, pp. 2521–2531, 2021.
- [454] Y. Mo, Q. Wu, X. Li, Y. Zhang, Y. Zhang, and Y. Zhang, "Remaining useful life estimation via transformer encoder enhanced by a gated convolutional unit," *Journal of Intelligent Manufacturing*, vol. 32, no. 8, pp. 1997–2006, 2021. [Online]. Available: <https://doi.org/10.1007/s10845-021-01750-x>
- [455] Y. Ding and M. Jia, "A convolutional transformer architecture for remaining useful life estimation," in *2021 Global Reliability and Prognostics and Health Management (PHM-Nanjing)*, 2021, pp. 1–7.
- [456] J. Zhu, Z. Long, X. Ma, and F. Luan, "Bearing remaining useful life prediction based on bert fine-tuning," in *2023 Global Reliability and Prognostics and Health Management Conference (PHM-Hangzhou)*, 2023, pp. 1–6.

- [457] S. Sankararaman, "Significance, interpretation, and quantification of uncertainty in prognostics and remaining useful life prediction," *Mechanical Systems and Signal Processing*, vol. 52-53, pp. 228-247, 2015. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0888327014002052>
- [458] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya, V. Makarenkov, and S. Nahavandi, "A review of uncertainty quantification in deep learning: Techniques, applications and challenges," *Information Fusion*, vol. 76, p. 243-297, Dec. 2021. [Online]. Available: <http://dx.doi.org/10.1016/j.inffus.2021.05.008>
- [459] M. Kraus and S. Feuerriegel, "Forecasting remaining useful life: Interpretable deep learning approach via variational bayesian inferences," *Decision Support Systems*, vol. 125, p. 113 100, 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0167923619301290>
- [460] B. Wang, Y. Lei, T. Yan, N. Li, and L. Guo, "Recurrent convolutional neural network: A new framework for remaining useful life prediction of machinery," *Neurocomputing*, vol. 379, pp. 117-129, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0925231219315024>
- [461] Z. Zhao, J. Wu, D. Wong, C. Sun, and R. Yan, "Probabilistic remaining useful life prediction based on deep convolutional neural network," in *9th International Conference on Through-life Engineering Service*, ser. SSRN Electronic Journal. Social Science Research Network, Oct. 2020.
- [462] L. D. Libera, J. Andreoli, D. D. Pezze, M. Ravanelli, and G. A. Susto, "Bayesian deep learning for remaining useful life estimation via stein variational gradient descent," 2024. [Online]. Available: <https://arxiv.org/abs/2402.01098>
- [463] M. Benker, L. Furtner, T. Semm, and M. F. Zaeh, "Utilizing uncertainty information in remaining useful life estimation via bayesian neural networks and hamiltonian monte carlo," *Journal of Manufacturing Systems*, vol. 61, pp. 799-807, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0278612520301928>
- [464] Liu, S., C. Lv, F. Song *et al.*, "Remaining useful life prediction integrating working conditions and uncertainty quantification based on multilayer graph neural networks," *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 2025. [Online]. Available: <https://doi.org/10.1007/s40430-025-05400-8>
- [465] D. Frizzo, F. Borsatti, and G. A. Susto, "A quantile regression approach for remaining useful life estimation with state space models," 2025. [Online]. Available: <https://arxiv.org/abs/2506.17018>
- [466] Y. Yin, J. Zhao, X. Zhang, and Y. Liu, "A novel remaining useful life prediction approach combined extreme gradient boosting and multiquantile recurrent neural network," *IEEE Access*, vol. 12, pp. 44 648-44 658, 2024.
- [467] R. Wen, K. Torkkola, and B. Narayanaswamy, "A multi-horizon quantile recurrent forecaster," 11 2017.
- [468] T. Khan, K. Ahmad, J. Khan, I. Khan, and N. Ahmad, "An explainable regression framework for predicting remaining useful life of machines," in *2022 27th International Conference on Automation and Computing (ICAC)*, 2022, pp. 1-6.
- [469] J. Figueroa Barraza, E. Drogue, and M. Ramos Martins, "Towards interpretable deep learning: A feature selection framework for prognostics and health management using deep neural networks," *Sensors*, vol. 21, p. 5888, 09 2021.
- [470] J. Grezmak, J. Zhang, P. Wang, K. Loparo, and R. Gao, "Interpretable convolutional neural network through layer-wise relevance propagation for machine fault diagnosis," *IEEE Sensors Journal*, vol. PP, pp. 1-1, 12 2019.
- [471] L. Rajaoarisoa, R. Randrianandraina, G. J. Nalepa, and J. a. Gama, "Decision-making systems improvement based on explainable artificial intelligence approaches for predictive maintenance," *Eng. Appl. Artif. Intell.*, vol. 139, no. PB, Feb. 2025. [Online]. Available: <https://doi.org/10.1016/j.engappai.2024.109601>
- [472] S. Vollert, M. Atzmueller, and A. Theissler, "Interpretable machine learning: A brief survey from the predictive maintenance perspective," in *2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2021, pp. 01-08.
- [473] D. Solís-Martín, J. Galán-Pérez, and J. Borrego-Díaz, "On the soundness of xai in prognostics and health management (phm)," 2023. [Online]. Available: <https://arxiv.org/abs/2303.05517>
- [474] R. K. Kundu and K. A. Hoque, "Explainable predictive maintenance is not enough: Quantifying trust in remaining useful life estimation," *Annual Conference of the PHM Society*, vol. 15, no. 1, 2023. [Online]. Available: <https://doi.org/10.36001/phmconf.2023.v15i1.3472>
- [475] L. Cummins, A. Sommers, S. B. Ramezani, S. Mittal, J. Jabour, M. Seale, and S. Rahimi, "Explainable predictive maintenance: A survey of current methods, challenges and opportunities," 2024. [Online]. Available: <https://arxiv.org/abs/2401.07871>
- [476] M. Achouch, M. Dimitrova, K. Ziane, S. Sattarpanah Karganroudi, R. Dhoub, H. Ibrahim, and M. Adda, "On predictive maintenance in Industry 4.0: Overview, models, and challenges," *Applied Sciences*, vol. 12, no. 16, p. 8081, 2022.
- [477] S. Arulnithika, D. R. Marur, and P. Taranalakshmi, "Iot based predictive maintenance in manufacturing," in *2025 International Conference on Recent Advances in Electrical, Electronics, Ubiquitous Communication, and Computational Intelligence (RAEEUCCI)*, 2025, pp. 1-5.
- [478] H. Singh, "Iot based predictive maintenance in industrial machinery," *International Journal for Modern Trends in Science and Technology*, vol. 11, no. 1, pp. 98-101, 07 2025. [Online]. Available: <https://feplb.com/index.php/csitrj/article/view/716>
- [479] D. Dhinakaran, S. Edwin Raja, R. Velselvi, S. Senthil Kumar, and S. Senthil Kumar, "Intelligent iot-driven advanced predictive maintenance system for industrial applications," *SN COMPUT. SCI.*, vol. 6, p. 151, 12 2025.
- [480] S. Elkateb, A. Méwalli, A. Shendy, and A. E. Abu-Elanien, "Machine learning and iot - based predictive maintenance approach for industrial applications," *Alexandria Engineering Journal*, vol. 88, pp. 298-309, 2024. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1110016823011572>
- [481] N. Longhi, L. M. Amorosa, S. Cavallero, E. Buracchini, and R. Verdene, "5g architectures enabling remaining useful life estimation for industrial iot: A holistic study," *IEEE Open Journal of the Communications Society*, pp. 1-1, 2025.
- [482] L. Lorenti, D. D. Pezze, J. Andreoli, C. Masiero, N. Gentner, Y. Yang, and G. A. Susto, "Predictive maintenance in the industry: A comparative study on deep learning-based remaining useful life estimation," in *2023 IEEE 21st International Conference on Industrial Informatics (INDIN)*, 2023, pp. 1-9.
- [483] N. Alshahrani, "Machine learning approaches for predictive maintenance in industrial operations," in *2024 IEEE 16th International Conference on Computational Intelligence and Communication Networks (CICN)*, 2024, pp. 365-372.
- [484] A. Al-Rafea, M. Al-atrash, and N. Lepkova, "Prediction of the remaining useful life of a milling machine using machine learning," *MethodsX*, vol. 14, p. 103 195, 2025. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2215016125000433>
- [485] S. Cancemi, M. Angelucci, A. Chierici, S. Paci, and R. Lo Frano, "Hybrid neural network and statistical forecasting methodology for predictive monitoring and residual useful life estimation in nuclear power plant components," *Nuclear Engineering and Design*, vol. 433, p. 113 900, 2025. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0029549325000779>
- [486] F. Maione, P. Lino, G. Maione, and G. Giannino, "Predictive maintenance of marine engines using a long short-term memory algorithm," in *2024 IEEE 8th Forum on Research and Technologies for Society and Industry Innovation (RTSI)*, 2024, pp. 578-583.
- [487] S. Lim, J. Oh, and J. Park, "Maintenance time prediction for predictive maintenance of ship engines," *Applied Sciences*, vol. 15, p. 4764, 04 2025.
- [488] J. Park and J. Oh, "A machine learning based predictive maintenance algorithm for ship generator engines using engine simulations and collected ship data," *Energy*, vol. 285, no. C, p. None, None 2023. [Online]. Available: <https://ideas.repec.org/a/eee/energy/v285y2023ics0360544223026634.html>
- [489] F. Kabir, B. Foggo, and N. Yu, "Data driven predictive maintenance of distribution transformers," in *2018 China International Conference on Electricity Distribution (CICED)*, 2018, pp. 312-316.
- [490] N. Bolander, H. Qiu, N. Eklund, E. Hindle, and T. Rosenfeld, "Physicsbased remaining useful life prediction for aircraft engine bearing prognosis," *Annual Conference of the Prognostics and Health Management Society*, 2009.
- [491] M. J. Daigle and K. Goebel, "A model-based prognostics approach applied to pneumatic valves," *Int J Progn Health Manag*, vol. 2, pp. 1-16, 2011.

- [492] —, “Model-based prognostics with concurrent damage progression processes,” *IEEE Trans Syst Man Cybern A*, vol. 43, no. 3, pp. 535–546, 2013.
- [493] D. Zhang, S. Dey, H. E. Perez, and S. J. Moura, “Remaining useful life estimation of lithium-ion batteries based on thermal dynamics,” in *2017 American Control Conference (ACC)*, 2017, pp. 4042–4047.
- [494] M. M. Özcan and U. Kutbay, “Improving predictive maintenance performance with limited data under different operating conditions with transfer learning,” in *2024 15th National Conference on Electrical and Electronics Engineering (ELECO)*, 2024, pp. 1–5.
- [495] X. Chen, Z. Chen, M. Zhang, Z. Wang, M. Liu, M. Fu, and P. Wang, “A remaining useful life estimation method based on long short-term memory and federated learning for electric vehicles in smart cities,” *PeerJ Computer Science*, 2023. [Online]. Available: <https://peerj.com/articles/cs-1652/>
- [496] L. Chen, H. Gao, L. Guo, J. Liang, and L. Peng, “Remaining useful life prediction of machinery using federated public feature representation in edge-cloud collaboration architecture,” *Engineering Applications of Artificial Intelligence*, vol. 143, p. 110 059, 2025. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0952197625000594>
- [497] M. L. Baptista, K. Goebel, and E. M. Henriques, “Relation between prognostics predictor evaluation metrics and local interpretability shap values,” *Artificial Intelligence*, vol. 306, p. 103 667, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0004370222000078>
- [498] X. Liu, W. Cheng, J. Xing, X. Chen, L. Li, Y. Guan, B. Ding, Z. Nie, R. Zhang, and Y. Zhi, “Predictive maintenance system for high-end equipment in nuclear power plant under limited degradation knowledge,” *Advanced Engineering Informatics*, vol. 61, p. 102 506, 2024. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S147403462400154X>
- [499] G. Sateesh Babu, P. Zhao, and X.-L. Li, “Deep convolutional neural network based regression approach for estimation of remaining useful life,” in *Database Systems for Advanced Applications*, S. B. Navathe, W. Wu, S. Shekhar, X. Du, X. S. Wang, and H. Xiong, Eds. Cham: Springer International Publishing, 2016, pp. 214–228.
- [500] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” 2023. [Online]. Available: <https://arxiv.org/abs/1706.03762>
- [501] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, “Inform: Beyond efficient transformer for long sequence time-series forecasting,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 11 106–11 115, 05 2021.
- [502] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *North American Chapter of the Association for Computational Linguistics*, 10 2018.
- [503] N. Tagasovska and D. Lopez-Paz, “Single-model uncertainties for deep learning,” 2019. [Online]. Available: <https://arxiv.org/abs/1811.00908>
- [504] J. Shang, D. Xu, H. Qiu, L. Gao, C. Jiang, and P. Yi, “A novel data augmentation framework for remaining useful life estimation with dense convolutional regression network,” *Journal of Manufacturing Systems*, vol. 74, pp. 30–40, 2024. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0278612524000347>
- [505] S. J. Julier and J. K. Uhlmann, “New extension of the Kalman filter to nmiscar systems,” in *Signal Processing, Sensor Fusion, and Target Recognition VI, ser. Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, I. Kadar, Ed., vol. 3068, Jul. 1997, pp. 182–193.
- [506] M. Raissi, P. Perdikaris, and G. Karniadakis, “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations,” *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0021999118307125>
- [507] A. Ucar, M. Karakose, and N. Kırımça, “Artificial intelligence for predictive maintenance applications: Key components, trustworthiness, and future trends,” *Applied Sciences*, vol. 14, no. 2, 2024. [Online]. Available: <https://www.mdpi.com/2076-3417/14/2/898>
- [508] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, “Playing atari with deep reinforcement learning,” 2013. [Online]. Available: <https://arxiv.org/abs/1312.5602>
- [509] V. Konda and J. Tsitsiklis, “Actor-critic algorithms,” in *Advances in Neural Information Processing Systems*, S.olla, T. Leen, and K. Müller, Eds., vol. 12. MIT Press, 1999. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/1999/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf
- [510] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” 2017. [Online]. Available: <https://arxiv.org/abs/1707.06347>
- [511] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, “Deterministic policy gradient algorithms,” *31st International Conference on Machine Learning, ICML 2014*, vol. 1, 06 2014.
- [512] M. Feng and Y. Li, “Predictive maintenance decision making based on reinforcement learning in multistage production systems,” *IEEE Access*, vol. 10, pp. 18 910–18 921, 2022.
- [513] K. S. H. Ong, W. Wang, D. Niyato, and T. Friedrichs, “Deepreinforcement-learning-based predictive maintenance model for effective resource management in industrial iot,” *IEEE Internet of Things Journal*, vol. 9, no. 7, pp. 5173–5188, 2022.
- [514] K. S. H. Ong, W. Wang, N. Q. Hieu, D. Niyato, and T. Friedrichs, “Predictive maintenance model for iiot-based manufacturing: A transferable deep reinforcement learning approach,” *IEEE Internet of Things Journal*, vol. 9, no. 17, pp. 15 725–15 741, 2022.
- [515] J. Lee and M. Mitici, “Deep reinforcement learning for predictive aircraft maintenance using probabilistic remaining-useful-life prognostics,” *Reliability Engineering & System Safety*, vol. 230, p. 108908, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0951832022005233>
- [516] A. N. Abbas, G. C. Chasparis, and J. D. Kelleher, “Hierarchical framework for interpretable and specialized deep reinforcement learning-based predictive maintenance,” *Data & Knowledge Engineering*, vol. 149, p. 102240, 2024. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0169023X23001003>
- [517] L. Brito, G. A. Susto, J. Brito, and M. Duarte, “Fault detection of bearing: An unsupervised machine learning approach exploiting feature extraction and dimensionality reduction,” *Informatics*, vol. 8, p. 85, 11 2021.
- [518] A. Agárdi and K. Nehéz, “Failure prediction with Weibull distribution,” *Infocommunications Journal*, vol. 17, no. 5, pp. 33–41, 2025.
- [519] H. K. Wang, X. Dai, Q. Ran, and et al., “Lithium-ion battery remaining useful life prediction based on dynamic filter frequency mixing learner and dual-stream mamba,” *Ionics*, 2025. [Online]. Available: <https://doi.org/10.1007/s11581-025-06715-1>

Reinforcement Learning, Optimization and Scheduling in Industry 5.0: Overview and Case studies

Alberto Sinigaglia³, Riccardo De Monte³, Patrick Stöckermann⁴, Alessandro Immordino⁴, Ilayda Celenk¹, Willem van Jaarsveld¹, Ivo Adan¹, Eric Weijers², Cas Rosman², Bibi de Jong², Kai Schelthoff², Bram Hoekstra^{1,2}, Laura Leja⁵, Gian Antonio Susto³, Oskars Teikmanis⁵, and Karlis Freivalds⁵

Abstract—This chapter reviews recent advances in the application of Deep Reinforcement Learning (DRL) to industrial scheduling and supply chain optimization. We first introduce foundational RL concepts and leading DRL algorithms, as well as various other machine learning algorithms. A central component of the chapter are various case studies, where DRL and other methods are applied to company use cases. The chapter concludes with perspectives on scalability, deployment, and research opportunities within Industry 4.0 and 5.0 contexts.

Index Terms—Reinforcement Learning, Scheduling, Optimization, Industry 4.0, Industry 5.0, Supply Chain, Logistics, Operational Research

I. INTRODUCTION

Recent years have witnessed a surge in interest around Reinforcement Learning (RL) and other machine learning (ML) techniques for solving decision-making problems in complex industrial environments. In this chapter, we explore the use of ML—particularly Deep Reinforcement Learning (DRL)—as a paradigm-shifting approach to operations planning and scheduling, where conventional methods struggle to meet scalability, adaptability, and real-time responsiveness requirements.

This review is structured to offer both theoretical insights and practical applications. After introducing foundational concepts in DRL and ML, we provide a comparative overview of state-of-the-art methods. A key case study is presented in the context of semiconductor supply chain planning, where DRL has been applied to address limitations of traditional optimization models. Finally, we highlight promising research directions and implementation considerations for deploying DRL in real-world industrial settings.

This work was funded by the European Union in the context of the Horizon Europe project “AIMS5.0 - Artificial Intelligence in Manufacturing leading to Sustainability and Industry 5.0” Grant agreement ID: 101112089.

¹ Eindhoven University of Technology, The Netherlands

² NXP Semiconductors N.V., The Netherlands

³ University of Padua, Italy

⁴ Infineon Technologies AG, Germany

⁵ EDI - Institute of Electronics and Computer Science, Latvia

II. OVERVIEW OF METHODS

A. Introduction to Reinforcement Learning

Reinforcement Learning deals with a two-entity framework, the agent and the environment. With reference to Fig. 1, the environment dictates the dynamics, and the agent acts on it to maximize a specific reward function. This setting is usually formalized under the umbrella of Markov Decision Process (MDP). An MDP consists of five main parts:

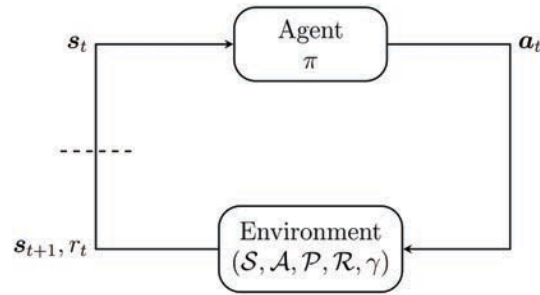


Fig. 1. Agent-environment interaction in a Markov Decision Process.

- 1) $\mathcal{S}$, called state space, is the set of states in which a reinforcement learning agent can be.
- 2) $\mathcal{A}$, called action space, is the set of actions that an agent can perform;
- 3) $\mathcal{P}$, called the transition function, rules the dynamics and is defined as $p(s_{t+1}|s_t, a_t)$. For the rest of the chapter, we will consider the discretized time RL setting, which transition from the current state and the new state will be dictated by the transition distribution;
- 4) $\mathcal{R}$, called the reward function, which acts as the supervision in the RL problem, and is defined as $p(r_t|s_t, a_t, s_{t+1})$. Poorly speaking, it defines the immediate reward an agent should receive given a certain decision;

- 5) γ , called the discount factor, which rules the trade-off between the importance of the immediate reward and the future rewards.

The behavior of an agent is identified in a policy that defines how the agent acts in a specific state. It is defined as a distribution $\pi(a_t|s_t)$. The support for such a distribution can change between states. Such policy aims at maximizing the return (expected cumulative sum of discounted rewards), defined as:

$$G(\pi) = \mathbb{E}_{s_0, a_0, s_1, \dots}^{\pi} \left[\sum_{i=0}^{\infty} \gamma^i r_i \right]$$

Given a specific policy π and discount factor γ , we can define two other quantities, the value function $V^{\pi} : \mathcal{S} \rightarrow \mathbb{R}$ and the action-value function $Q^{\pi} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$:

$$V^{\pi}(s_t) = \mathbb{E}_{a_t, s_{t+1}, a_{t+1}, s_{t+2}, \dots}^{\pi} \left[\sum_{i=0}^{\infty} \gamma^i r_{t+i} \mid s_t \right]$$

$$Q^{\pi}(s_t, a_t) = \mathbb{E}_{s_{t+1}, a_{t+1}, s_{t+2}, \dots}^{\pi} \left[\sum_{i=0}^{\infty} \gamma^i r_{t+i} \mid s_t, a_t \right]$$

All RL algorithms can be derived from such quantities, however, an additional one is defined since is extensively used in literature, which is the Advantage A , which is defined as:

$$A^{\pi}(s_t, a_t) = Q^{\pi}(s_t, a_t) - V^{\pi}(s_t)$$

Intuitively, it models the improvement obtained thanks to action a_t from the policy π compared to the average return in such state.

The challenge in RL is learning these functions in order to derive a policy improvement algorithm. In the naive setting with a finite set of possible states and actions, and these sets being reasonably small, a tabular setting is usually employed, where the table represents all possible combinations of states and actions $\mathcal{S} \times \mathcal{A}$. However, in real world scenario, the size of these sets makes such approach prohibitive, or impossible in the case of continuous settings. In order to solve this criticality, RL usually relies in approximations of these functions.

All the algorithms presented in the following sections, will take advantage on a subset of these tools ($\pi, \mathcal{A}, \mathcal{V}, \mathcal{Q}$) approximated with some function approximator, which usually are Neural Networks, in which case are referred as Deep Reinforcement Learning (DRL) algorithms.

Most of recent advancements in DRL fall under the umbrella of Q-learning algorithms, and Actor Critic algorithms. The former aims at distilling a policy directly from the Q-function, this requiring the learning of only that component. Actor Critic algorithms instead aim at learning the policy π , and exploit either $\mathcal{A}$ or $\mathcal{V}$ or $\mathcal{Q}$ to do so.

B. State of the Art of Deep Reinforcement Learning

Having introduced the fundamental concepts of Reinforcement Learning (RL)—value functions (V^{π}), action-value functions (Q^{π}), advantage functions (A^{π}), and stochastic or deterministic policies (π, μ)—we now focus on the principal deep RL algorithms that have driven recent progress in the field. Each of the following methods leverages neural networks to

approximate one or more of the core functions, and incorporates various techniques to stabilize learning and improve performance.

1) *Deep Q Networks*: Deep Q Networks (DQN) [1] were among the first successful attempts to combine deep learning with Q-learning, enabling agents to learn directly from high-dimensional inputs such as raw pixels in Atari games.

a) *Objective*.: Let $Q_{\theta}(s, a)$ be the parameterized action-value function (Q-function) approximated by a neural network with parameters θ . DQN minimizes the following loss function:

$$\mathcal{L}(\theta) = \mathbb{E}_{(s, a, s', r) \sim \mathcal{D}} \left[(r + \gamma \max_{a'} Q_{\theta^-}(s', a') - Q_{\theta}(s, a))^2 \right], \quad (1)$$

where θ^- are the parameters of a slowly updated target network and (s, a, s', r) are transitions sampled from a replay buffer $\mathcal{D}$, where agent's experiences are stored. The replay buffer improves data efficiency, namely, the same transition can be used for multiple gradient updates, and decorrelates samples.

b) *Target Network and Double DQN*.: The use of a second target network, namely keeping a distinct set of parameters for the Q-learning target in equation 1, has shown improved training stability. Double DQN [2] further improves DQN performance by proposing the following target:

$$r + \gamma Q_{\theta^-}(s', \arg\max_{a'} Q_{\theta}(s', a')) \quad (2)$$

This allows reducing overestimation of the target by decomposing the max operation in the target into action selection and action evaluation.

2) *REINFORCE*: Instead of learning the action-value functions (Q^{π}) and derive a greedy policy from it, Policy Gradient methods, as REINFORCE [3], directly optimize the parameters θ of the policy $\pi_{\theta}(a|s)$ by sampling entire trajectories from the environment. Such an approach is effective in high-dimensional or continuous action spaces, contrary to DQN-based methods.

To do so, REINFORCE seeks to maximize the expected return $J(\theta)$ defined as:

$$J(\theta) = G(\pi_{\theta}) = \mathbb{E}^{\pi_{\theta}} \left[\sum_{t=0}^{T-1} \gamma^t r_t \right] \quad (3)$$

where T is the length of an episode (assuming an episodic setting where for any trajectory ends in a final state S_T). The gradient of $J(\theta)$ with respect to θ is given by the so-called Policy Gradient (PG) Theorem [4]:

$$\nabla_{\theta} J(\theta) = \mathbb{E}^{\pi_{\theta}} \left[\sum_{t=0}^{T-1} \nabla_{\theta} \log \pi_{\theta}(a_t|s_t) G_t \right], \quad (4)$$

$$G_t = \sum_{k=t}^{T-1} \gamma^k r_{t+k}$$

with G_t representing the return (or discounted return) following time step t . In practice, to mitigate high variance, REINFORCE implementations subtract to G_t a learned baseline e.g. the value function $V^{\pi_{\theta}}(s_t)$.

3) *Advantage Actor Critic (A2C / A3C)*: Actor-Critic methods combine policy-gradient (the *actor*) with a value function estimator (the *critic*). Actor-Critic methods estimate the advantage function $A^{\pi_{\theta}}(s_t, \mathbf{a}_t)$ by bootstrapping. In particular, provided an estimate of the value function for any state $\hat{V}^{\pi_{\theta}}(s)$, an estimate of the advantage $A^{\pi_{\theta}}(s_t, \mathbf{a}_t)$ is given by the so-called k -steps Temporal Difference error (TD(k)) error:

$$\hat{A}^{\pi_{\theta}}(s_t, \mathbf{a}_t) = \sum_{i=0}^k \gamma^i r_{t+i} + \gamma^{k+1} \hat{V}^{\pi_{\theta}}(s_{t+k+1}) - \hat{V}^{\pi_{\theta}}(s_t) \quad (5)$$

Advantage Actor Critic methods usually estimate $V^{\pi_{\theta}}(s)$ by training a neural network as a function approximator. Therefore, by denoting with ϕ the learnable parameter of such a network, called *Critic network*, $\hat{V}_{\phi}^{\pi_{\theta}}(s)$ is used as an estimate of the value function. As consequence, Advantage Actor Critic approaches jointly update the Actor network (π_{θ}) and the Critic one. As for REINFORCE, policy parameters θ are updated according to PG:

$$\nabla_{\theta} J(\theta) = \mathbb{E} \left[\nabla_{\theta} \log \pi_{\theta}(\mathbf{a}_t | s_t) \hat{A}^{\pi_{\theta}}(s_t, \mathbf{a}_t) \right] \quad (6)$$

Regarding the Critic network, the parameters are updated by minimizing the Mean Squared Error (MSE) between the current estimate $\hat{V}^{\pi_{\theta}}(s)$ and the TD(k) target. In the simple case where $k = 0$, the loss for the Critic is given by:

$$\mathcal{L}(\phi) = \mathbb{E} \left[\left(r_t + \gamma \hat{V}^{\pi_{\theta}}(s_{t+1}) - \hat{V}^{\pi_{\theta}}(s_t) \right)^2 \right] \quad (7)$$

where $r_t + \gamma \hat{V}^{\pi_{\theta}}(s_{t+1})$ is the TD(1) target at given state s_t .

a) *Synchronization.*: In the literature, there are two popular implementations for Advantage Actor Critic: Asynchronous Advantage Actor Critic (A3C) [5] and the synchronous variant A2C. In A3C, multiple agents explore in parallel on different threads, asynchronously updating the shared parameters, while for A2C updates are collected from parallel actors and applied in a batch.

4) *Proximal Policy Optimization (PPO)*: In Reinforcement Learning, as in traditional Machine Learning, the choice of the learning rate is crucial. A small learning rate often requires multiple updates on the same samples. However, in RL, this can lead to large policy updates that are not justified, as the advantage function used in the Policy Gradient update depends on the policy itself.

To address this, TRPO [6] introduces a surrogate objective function along with a constraint to prevent excessively large updates. However, TRPO is challenging to implement in frameworks like PyTorch and is incompatible with techniques like Dropout or parameter sharing between the critic and the policy.

To overcome these limitations, PPO [7] introduces a clipped surrogate objective, making it both simpler to implement and more broadly applicable while still controlling policy updates. Defined the following likelihood ratio:

$$r_t(\theta) = \frac{\pi_{\theta}(\mathbf{a}_t | s_t)}{\pi_{\theta_{\text{old}}}(\mathbf{a}_t | s_t)}, \quad (8)$$

where $\pi_{\theta_{\text{old}}}$ denotes the policy used to sample, the clipped surrogate objective (the most used PPO objective) is:

$$\mathcal{L}^{\text{CLIP}}(\theta) = \mathbb{E} \left[\min(r_t(\theta) \hat{A}_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon) \hat{A}_t) \right], \quad (9)$$

where $\hat{A}_t = \hat{A}^{\pi_{\theta_{\text{old}}}}(s_t, \mathbf{a}_t)$ is the estimate of the advantage at time step t concerning the sampling policy $\pi_{\theta_{\text{old}}}$ (as done for A2C/A3C, a critic network is used for this purpose), and ϵ is a hyperparameter (typically small, e.g., 0.1 or 0.2). This *clipping* prevents excessively large policy updates while preserving incentive to improve on-policy actions.

5) *Deep Controlled Learning*: Deep Controlled Learning (DCL) [8] is a Deep Reinforcement Learning (DRL) algorithm tailored to high-variance environments such as inventory management and supply chain scheduling. It operates within the Approximate Policy Iteration (API) framework, where the goal is to iteratively improve a policy π by approximating the optimal value function and deriving improved policies.

Unlike conventional DRL algorithms, DCL decouples the learning of the value function from policy learning. It focuses on the policy learning step as a supervised classification task:

$$\pi(s) = \arg \min_{a \in \mathcal{A}} \mathbb{E}[c(s, a)]$$

where $c(s, a)$ is the estimated cost of taking action a in state s , obtained via simulation.

To estimate $\mathbb{E}[c(s, a)]$ efficiently and robustly, DCL combines two algorithmic techniques:

- **Sequential Halving (SH)**: An adaptive sampling method that allocates simulation budget to promising actions, discarding suboptimal ones early.
- **Common Random Numbers (CRN)**: A technique that reduces variance by using shared randomness across simulations for different actions, ensuring fair comparisons.

Given a set of actions $\mathcal{A}$, each action a is simulated under identical scenarios $\omega \in \Omega$, and the empirical mean cost $\hat{c}(s, a)$ is computed. The selected action minimizes this estimate:

$$\pi(s) = \arg \min_{a \in \mathcal{A}} \hat{c}(s, a)$$

This process is repeated for each state in a dataset, and a policy neural network is trained to predict the optimal action given any state.

DCL's design ensures scalability and stability. Empirical studies have shown that DCL performs consistently well across diverse inventory control scenarios—including lost sales, perishable goods, and random lead times—achieving average cost reductions with an optimality gap below 0.2% compared to known optimal policies.

The algorithm's generality, low variance, and robust performance with fixed hyperparameters make it a compelling choice for dynamic supply chain and inventory planning under uncertainty.

6) *Preconditioned Proximal Policy Optimization (P3O)*: While the clipping PPO objective aims to prevent large policy

updates, it might cause loss of exploration. For this reason P3O [9] proposes the following objective:

$$\mathcal{L}^{\text{p3o}}(\theta) = \mathbb{E} \left[\mathcal{L}^{\text{scp}}(\theta) - \beta \cdot \text{KL}(\pi_{\theta_{\text{old}}}(\cdot|s_t), \pi_{\theta}(\cdot|s_t)) \right] \quad (10)$$

$$\mathcal{L}^{\text{scp}}(\theta) = \sigma \left(\tau \cdot (r_t(\theta) - 1) \right) \cdot \frac{4}{\tau} \hat{A}_t \quad (11)$$

where τ and β are two hyperparameters, $\sigma(\cdot)$ denotes the sigmoid function, $\text{KL}(\cdot, \cdot)$ denotes the Kullback–Leibler (KL) divergence and $r_t(\theta)$ denotes the likelihood ratio defined in equation 8. $\mathcal{L}^{\text{scp}}(\theta)$ is called *Scopic objective* and, contrary to the clipped surrogate objective used in PPO, allows the policy to be optimized in a much larger space promoting exploration. On the other hand, the KL divergence keeps the importance sampling ration $r_t(\theta)$ close to one, to better balance exploration and exploitation.

7) *Deep Deterministic Policy Gradient (DDPG)*: DDPG [10] extends Q-learning to continuous actions by following the Actor-Critic paradigm, namely, maintaining both a parameterized critic (Q-function) $Q_{\phi}(s, a)$ and a parameterized, deterministic actor $\mu_{\theta}(s)$. As for DQN, DDPG exploits a replay buffer $\mathcal{D}$, where samples (s, a, r, s') are collected according to the exploration policy $\mu(s) = \mu_{\theta}(s) + \mathcal{N}$, where $\mathcal{N}$ is any random noise, e.g. Gaussian noise.

a) *Critic Update.*: Similar to DQN, DDPG uses a target network for both Q , Q_{ϕ^-} , and μ , μ_{θ^-} . These are updated with an Exponential Moving Average update, and allows to derive a target for the critic that guarantees stable updates:

$$\mathcal{L}(\phi) = \mathbb{E}_{(s,a,r,s') \sim \mathcal{D}} \left[(r + \gamma Q_{\phi^-}(s', \mu_{\theta^-}(s')) - Q_{\phi}(s, a))^2 \right]. \quad (12)$$

b) *Actor Update.*: Following the Deterministic Policy Gradient Theorem [11], the deterministic policy gradient is computed by backpropagating through the Q-function with respect to the policy:

$$\nabla_{\theta} J(\mu_{\theta}) = \mathbb{E}_{s \sim \mathcal{D}} \left[\nabla_a Q_{\phi}(s, a) \Big|_{a=\mu_{\theta}(s)} \nabla_{\theta} \mu_{\theta}(s) \right]. \quad (13)$$

8) *Twin Delayed Deep Deterministic Policy Gradient (TD3)*: TD3 [12] improves upon DDPG by mitigating overestimation bias in Q-learning. It uses two critics and delays policy updates, leading to more stable training.

a) *Overestimation Mitigation.*: When computing the target, TD3 takes the minimum of the two critic networks:

$$y = r + \gamma \min_{i=1,2} Q_{\phi_i^-}(s', \mu_{\theta^-}(s') + \epsilon), \quad (14)$$

where ϵ is a small noise added for smoothing (target policy smoothing). The critics are updated by minimizing the difference to this target, and the policy update is performed less frequently (e.g., every two critic updates).

9) *Soft Actor Critic (SAC)*: SAC [13] incorporates an entropy term to encourage broader exploration. Instead of simply maximizing the expected return, SAC optimizes a *soft* value function that balances return and policy entropy.

a) *Soft Q Function.*: The soft Q-function $Q_{\phi}(s, a)$ is learned by minimizing

$$L_Q(\phi) = \mathbb{E}_{(s,a,r,s') \sim \mathcal{D}} \left[(Q_{\phi}(s, a) - (r + \gamma V_{\bar{\psi}}(s')))^2 \right], \quad (15)$$

where $V_{\bar{\psi}}(s)$ is a parameterized value function network (or, in some variants, omitted in favor of a second Q-network).

b) *Entropy-Augmented Policy Update.*: The policy $\pi_{\theta}(a|s)$ aims to minimize the Kullback–Leibler divergence between itself and an exponential distribution defined by the soft Q-function. One way to write the objective is:

$$\nabla_{\theta} J(\theta) = \mathbb{E}_{s \sim \mathcal{D}, a \sim \pi_{\theta}} \left[\nabla_{\theta} \alpha \log \pi_{\theta}(a|s) - \nabla_{\theta} Q_{\phi}(s, a) \right], \quad (16)$$

where α is the temperature parameter that controls the trade-off between exploration (entropy) and exploitation (return). An adaptive scheme for α can also be employed to dynamically adjust the entropy bonus.

c) *Advantages of SAC.*: By emphasizing entropy in the objective, SAC avoids prematurely collapsing to a near-deterministic policy, often leading to more robust and stable learning across a wide range of continuous control environments.

Overall, these algorithms showcase the evolution of deep RL, illustrating how different design choices—on-policy vs. off-policy, deterministic vs. stochastic policies, clipping vs. entropy regularization, or single vs. multiple critics—can greatly affect training stability and performance. While each method has its own strengths and trade-offs, together they form the core toolkit for solving complex RL tasks in both simulation and real-world applications.

C. Introduction to Evolutionary Methods

Evolutionary methods are optimization techniques inspired by natural evolution, emulating concepts such as mutation, selection, and recombination. These methods iteratively evolve candidate solutions to optimize complex, noisy, or non-differentiable functions. Unlike gradient-based optimization, evolutionary strategies do not rely on gradient information, thus providing robustness in scenarios where gradient information is either unavailable or unreliable.

At their core, evolutionary algorithms maintain a population of candidate solutions. Through repeated cycles of evaluation and variation, they progressively improve these solutions by favoring those exhibiting superior performance, typically measured by a fitness function. A general evolutionary approach includes the following steps:

- 1) **Initialization**: Generate an initial population of candidate solutions.
- 2) **Evaluation**: Compute the fitness of each candidate solution.
- 3) **Selection**: Select a subset of candidates based on fitness to form a new generation.
- 4) **Variation**: Apply mutation and recombination to create offspring solutions.
- 5) **Replacement**: Replace less fit solutions with better-performing offspring.

Contrary to traditional Reinforcement Learning (RL) approaches, Evolutionary Strategies (ES) do not rely on gradient estimates. Instead they treat the optimization of policies as a black-box optimization problem. This makes ES particularly suited for scenarios where the reward signals are sparse, delayed, or noisy, conditions frequently encountered in industrial scheduling and real-world RL problems. Prominent evolutionary methods, such as the Cross-Entropy Method (CEM), Covariance Matrix Adaptation Evolution Strategy (CMA-ES), and Separable Natural Evolution Strategies (SNES), have demonstrated their efficacy in optimizing neural network policies directly, effectively handling high-dimensional and complex policy spaces.

D. State of the Art Evolutionary Methods

Modern evolutionary optimization methods have significantly advanced, providing powerful alternatives to classical optimization and reinforcement learning in complex industrial applications. Among these, three notable state-of-the-art methods include Cross-Entropy Method (CEM), Covariance Matrix Adaptation Evolution Strategy (CMA-ES), and Separable Natural Evolution Strategies (SNES).

Each of these evolutionary strategies addresses different optimization challenges. CMA-ES stands out for its robustness in noisy, correlated parameter spaces, CEM offers simplicity and interpretability, and SNES excels in scalability and computational efficiency. Collectively, these methods form the cornerstone of evolutionary optimization, enabling effective exploration and exploitation in complex real-world scenarios.

1) *Cross-Entropy Method (CEM)*: Initially proposed for rare-event simulation and subsequently adopted for optimization, CEM is an iterative procedure optimizing a parameterized probability distribution of solutions. At each iteration, CEM samples solutions from a parameterized distribution (typically Gaussian), selects elite samples based on their performance, and updates the distribution parameters by minimizing the cross-entropy distance between the sampling distribution and the distribution of elite samples. Formally, the update rule for mean μ and covariance Σ can be expressed as:

$$\mu_{t+1} = \frac{1}{N_e} \sum_{i=1}^{N_e} x_i, \quad \Sigma_{t+1} = \frac{1}{N_e} \sum_{i=1}^{N_e} (x_i - \mu_{t+1})(x_i - \mu_{t+1})^T, \quad (17)$$

where x_i are elite samples, and N_e denotes the number of elite samples. The simplicity and adaptability of CEM make it appealing for problems where gradient computation is infeasible or costly, although it can be prone to premature convergence in multimodal landscapes.

2) *Covariance Matrix Adaptation Evolution Strategy (CMA-ES)*: CMA-ES represents a significant advancement in evolutionary methods, particularly adept in handling continuous optimization problems with noisy fitness landscapes. CMA-ES explicitly models dependencies among variables by adaptively updating a covariance matrix of a multivariate Gaussian distribution. Formally, CMA-ES updates the mean and covariance matrix using information from elite solutions:

$$\begin{aligned} \mu_{t+1} &= \mu_t + \sigma_t \sum_{i=1}^{N_e} w_i (x_i - \mu_t), \\ \Sigma_{t+1} &= \sum_{i=1}^{N_e} w_i (x_i - \mu_t)(x_i - \mu_t)^T, \end{aligned} \quad (18)$$

where w_i are weights assigned according to the rank-based fitness of the samples x_i . This allows CMA-ES to adaptively adjust the step-size and the search direction, significantly enhancing its efficiency and robustness against ill-conditioned objective functions and improving convergence speed and robustness.

3) *Separable Natural Evolution Strategies (SNES)*: Separable Natural Evolution Strategies (SNES) simplify the computational complexity of covariance-based methods like CMA-ES by assuming a diagonal covariance structure. This method optimizes a Gaussian distribution characterized by its mean and a diagonal covariance, significantly reducing the number of parameters to be updated. SNES maximizes expected fitness through a gradient estimate derived from natural gradients:

$$\theta_{t+1} = \theta_t + \eta F^{-1} \nabla_{\theta} \mathbb{E}_{\theta}[f(x)], \quad (19)$$

where $\theta = (\mu, \Sigma)$ are the parameters of the Gaussian distribution, and the diagonal covariance ensures scalability to high-dimensional problems. SNES is particularly effective in large-scale optimization tasks where computational efficiency is critical.

E. Introduction to Differentiable Physics

Gradient-based learning with differentiable simulations, or differentiable physics (DP), is a growing field of machine learning that takes advantage of direct gradient propagation and offers significant improvements in sample efficiency, training stability, and computational performance in complex control tasks, compared to reinforcement learning [14], [15], [16], [17].

1) *Foundational Concepts*: Differentiable simulators process physical dynamics as end-to-end differentiable computational graphs [14]. Unlike conventional simulators, which treat state transitions such as

$$s_{t+1} = f(s_t, a_t)$$

as black-box functions, differentiable implementations enable gradient calculation through the state transition process. This allows for a direct computation of expressions such as

$$\frac{\partial s_{t+1}}{\partial a_t} \quad \text{and} \quad \frac{\partial s_{t+1}}{\partial s_t},$$

creating a continuous optimisation pathway from actions to outcomes. For example, in robotic manipulation tasks, this property enables backpropagation of positioning errors through the entire sequence of manipulation tasks.

In contrast, reinforcement learning (RL) generally relies on stochastic policy exploration and reward signal propagation, requiring extensive environmental interactions to estimate value functions. The policy gradient approach of the

REINFORCE algorithm [3] exemplifies this trial-and-error paradigm, where agents must discover successful action sequences through temporal difference learning and Monte Carlo sampling. Although effective in diverse domains, this methodology becomes prohibitively expensive in scenarios with high-dimensional state spaces or delayed rewards. Gradient-based learning circumvents RL's exploration bottleneck by using analytic gradients from differentiable models to directly optimise policy parameters.

F. State of the Art in Differentiable Physics (EDI)

1) *Overview of Differentiable Simulator Toolkits:* Early work on DP for control by Belbute-Peres et al. [18] shows that end-to-end differentiability with analytical gradients comes with a number of benefits, including parameter identification and sample-efficient learning by embedding differentiable physics within deep networks for more accurate predictions of physical behaviours from minimal observational data across diverse applications, including fluid-flow dynamics [19], [20], [21], [22], [23], and robotics tasks [24], [25]. Since then, there has been a rise of diverse differentiable simulation engines over the past years, including tools such as Brax [26], Dojo [27], NimblePhysics [28], Taichi [29] and Warp [30]. An overview of their features can be seen in Table I.

Compared to general-purpose ML libraries such as TensorFlow and PyTorch, differentiable simulation toolkits feature diverse physics solvers and may come with optimisation and learning tools or easy integration with existing machine learning libraries. For example, BRAX has Built-in RL algorithms in JAX, while Warp and Nimble can be easily integrated with PyTorch or Tensorflow.

2) *Applications in Robotic Manipulation Tasks:* Bridging the reality gap is one of the most important steps in robotics in order to operate in the real world. One method for accomplishing this is by parameter identification. Lutter et al. [17] demonstrate that knowledge of physics can significantly reduce the required number of measurement samples for achieving accurate control. This is accomplished by comparing the offline learning performance of an analytically differentiable white-box model of a robotic arm with black-box models of the same robot in a ball-in-cup task. The authors find that the white-box model consistently perform better than black-box models, despite being trained on significantly less data, while also complying with nonholonomic constraints (e.g. ball-on-string). Collins et al. [31] demonstrate similar results in an online learning setting by iterating through real-to-sim optimisations and sim-to-real transfers for continuous improvements of the robot model.

Accurate contact modelling is another important requirement for contact-rich robotic manipulation tasks. This is a challenge, since contact simulations are inherently difficult to differentiate due to potentially numerous discontinuities. Still, even these challenges can be addressed with differentiable physics by either mollifying (softening) contact forces [32], [24], [25], integrating collision detection and contact simulation into one unified differentiable framework [33] and introducing adaptive-horizon rollouts to limit gradient errors

from stiff contact dynamics within differentiable simulations [34].

III. DRL AND OPTIMIZATION: APPLICATION AREAS IN INDUSTRY 5.0

DRL and Optimization are broadly applied in Industry 4.0 and Industry 5.0, e.g. in the areas:

- Supply chain and logistics
- Autonomous robotic systems and adaptive control
- Energy and resource management
- Quality control, predictive maintenance
- Collaborative robots, adaptive workflows
- Resilient production systems

Next, we present a detailed discussion of applications in specific use cases:

- Wafer Lot Scheduling in Semiconductor Frontend Fabs
- Master Planning for Semiconductor Supply Chain Simulation
- Reel Utilization in Production by 3D printing
- Long-Term Rapid Scenario Planning in the Semiconductor Industry using DRL
- Differentiable physics training method for robot motion planning

A. Wafer Lot Scheduling in Semiconductor Frontend Fabs

1) *Problem Description:* Semiconductor Frontend Manufacturing (SCFM) is one of the most complex industrial processes in the world. Silicon wafers are going through an intricate process flow which can entail hundreds of steps and can take multiple months. Those steps can include diffusion, etching processes, lithography, ion implementation, measuring steps and more. During those processes, the intended electrical structure is build layer by layer on the wafer. After the wafer processing in the frontend of line, the wafers are tested and move on to the backend of line where they are cut into piece, packaged and bonded to create the interconnections between the integrated circuit (IC) and its packaging. As the tools are very expensive and can cost many millions of USD each [35], we want to utilize them as much as possible. In order to achieve a high utilization, we accept queues of waiting lots in front of the machines. To achieve good WIP (Work-In-Progress)-flow and balancing of load across the routes of the different products as well as across the machines, the queues of waiting lots have to be optimized. [36]

This optimization is described by the Semiconductor Factory Scheduling Problem (SFSP), which is regarding the efficient assignment of wafer lots to specific machines in a designated order. The problem is often formalized as a Complex-Job-Shop-Scheduling Problem (CJSSP) in academia. The CJSSP is based on the Job-Shop-Scheduling Problem (JSSP), which describes the efficient assignment of n jobs $\mathcal{J}$ to m machines $\mathcal{M}$. In the basic version, each job of a unique type has to be processed once on each machine in a specific order. The CJSSP extends this problem by introducing machine groups with parallel machines. Furthermore, batching processes are considered, where multiple jobs can be processed

TABLE I
FEATURES OF DIFFERENTIABLE SIMULATION TOOLKITS.

Feature/Engine	BRAX	Dojo	Nimble	Taichi	Warp
Physics domain	Particle/Rigid/CFD	Robotics	Robotics	Multi-domain	Rigid/Soft
Differentiation	JAX AD	Implicit diff.	Chain rule AD	Kernel-grad AD	CUDA AD
Contact handling	Convex approx.	NCP/SOC	LCP	Hybrid	Compliant
Contact differentiation	Smooth approx.	Analytic	Limited	TOI-Velocity	Semi-analytic
Acceleration	GPU/TPU	GPU/CUDA	CPU/GPU	GPU	GPU

on the same machine in parallel if their setup requirements are matching. The CJSSP also incorporates the concept of reentrant flow, meaning that the jobs can revisit machines multiple times in their process flow. [36]

An important intrinsic property of SCFM is the strong stochasticity. Frequent machine failures are normal in day to day operations and directly affect the SFSP. To prevent unscheduled downtime, some manufacturers do preventive maintenance. The waiting times of wafer lots at individual steps, transport times as well as processing times are stochastic. Quality checks can trigger optional rework steps. [36]

2) *Heuristics*: Traditionally, simple heuristics, often in an hierarchical order, have been used in the SCFM industry to control the dispatching order of queues of waiting lots. Those can be as simple as First-In-First-Out (FIFO). This approach is good for comprehensibility as well as keeping a good overviews. The system is however limited to manual human intervention when it comes to dynamically reacting to arising bottlenecks or other material-flow related problems.

Researchers have presented solutions to optimize metaheuristics to better fit specific problem instances. This can be done by use of WIP-flow simulations. Linear combinations of weighted lot properties can be used to score and sort queues of weighting lots. The coefficients of that linear combination can be optimized though iterative perturbation and simulation runs.

An alternative approach in academia is the use of population-based metaheuristics called swarm algorithms. Some popular example are the Artificial Bee Colony (ABC) swarm algorithm as well as Ant Colony Optimization (ACO) which aim at a decentralized optimization. The idea is to observe properties in the neighboring environment, such as the utilization of the machines in the following steps. [37] and [38] show the effectiveness of these algorithms in smaller scheduling instances regarding semiconductor manufacturing. Another population-based metaheuristic approach are Evolutionary Algorithms (EAs) such as Genetic Algorithms (GAs), which have been shown to be effective for complex scheduling problems [39].

3) *Classical Mathematical Optimization*: Scheduling problems are often solved with exact mathematical methods such as Constraint Programming (CP), Mixed-Integer-Programming (MIP), Linear Programming (LP) or Answer-Set-Programming (ASP) [40], [41], [42], [43], [44], [45]. Those methods aim at finding the exact best solution or to at least estimate the solution gap. One of their advantages is that they can dynamically adapt to the local situation, i.e., to rebalance the

load across parallel machines.

As already the simple JSSP is NP-hard [46], the SFSP is often decomposed into subproblems, such as the Parallel-Machine-Scheduling-Problem (PMSP), which considers only the queue of waiting lots in front of a set of parallel machines of the same type. An acceptable solution has to be found by the solver within a specified interval time. This interval times depends on the dynamics of the manufacturing environment which can require rescheduling from one minute to the other if, i.e., tools are going down or rework has been triggered. An alternative is to accept strong abstractions of the problem, not considering all constraints in detail and disregarding important aspects such as stochasticity.

4) *Deep Reinforcement Learning*: Similar to metaheuristics, DRL can iteratively optimize a scheduling policy using a simulation based approach. This enables the direct optimization towards global fab level KPIs. Furthermore, DRL enables the training of a dynamic policy which is changing based on the state of the environment. A metaheuristic can learn a policy defined by a static linear combination of input features which works well on average. A stochastic policy trained by RL can be defined by a Deep Neural Network and dynamically change a nonlinear combination of input features to adapt to the dynamically changing situation in the manufacturing environment which is subject to an every changing load mix and stochastic machine failures.

One of the main challenges in DRL is the problem formulation through the characterization of the main components of the DRL paradigm, namely reward signal, state and action space. The choice of a suitable DRL algorithm is heavily influenced by this preliminary settings. In the context of lot scheduling in semiconductor manufacturing, the main objectives are usually makespan minimization [47], [48], utilization [49] or on time delivery [50]. Thus, the reward signal is usually related to these metrics. In this context, the state provided to the agent is dependent on the type of event that triggers the interaction between agent and simulation. A discrete event simulation may either halt after a predefined time interval or when a machine under the agent's control becomes idle. Regarding the former approach, some studies have incorporated an embedded graph structure into the observation space [47], [51], [52], which represents the relationships between jobs and machines or among the machines.

Alternatively, other approaches represent the state using multiple matrices [53], [54], [55], which define job-to-machine assignments, job processing times per machine, or the remaining processing times for each machine. In case the simulation

dynamically presents jobs waiting in front of an idle machine to the agent, the extracted features are typically more closely related to the characteristics of the lots [56], [49].

In this setting, the action space can be defined as assigning the position of the lot in the queue. Alternatively, actions can be defined as selecting or weighting predetermined dispatching rules commonly used in manufacturing contexts, such as shortest processing time (SPT) or first-in first-out (FIFO) [50], [54]. Another approach is to define actions space as the set of possible job-to-machine allocations [53], [52].

Once the main components of the MDP are defined, the choice of RL algorithm depends heavily on the complexity of the underlying simulation. If the simulation models a real semiconductor manufacturing process, the computational time required for the agent to interact with the simulator can be significantly higher compared to commonly used open-source benchmark models. In such cases, on-policy methods like PPO may be more suitable, as they compute updates on the fly without storing samples in a replay buffer. Asynchronous policy updates, such as in the A3C algorithm, can be beneficial when using multiple agents in parallel, as it allows for gradient calculation to begin without waiting for all agents to finish collecting their samples by interacting with the environment. This is particularly useful, as parallel agents can potentially complete episodes at very different times, i.e., due to stochastic events, such as machine breakdowns. Table II summarizes the major contributions regarding RL methods in the field of scheduling and dispatching which are cited in this section.

One limitation of RL approaches is the comprehensibility of the result, as the results have to be communicated to experts in the manufacturing domain who might not have in-depth knowledge of Deep Learning. Even for data scientists, the resulting policies are not fully explainable and can only be understood approximately. There is a tradeoff between the comprehensibility as well as the performance of the methods. Explainable Reinforcement Learning (XRL) is a growing field that aims to make RL agents more understandable and interpretable to humans by providing insights into the inner workings of an agent's policy. Several methods have been developed to address this challenge. One approach involves visualizing which input feature were most influential for the agent decision, i.e., with saliency maps for image based observations [57]. Another strategy is imitation-based explanations, where interpretable models such as decision trees or symbolic rules are trained to mimic the behavior of a more complex policy. Reward decomposition is another powerful technique [58]. Instead of treating the reward as a single scalar signal, it can be broken down into human understandable components that reveal what aspects of the environment the agent is prioritizing [59].

5) *Evolution Strategies / Neuroevolution*: One of the main problems in RL is the credit-assignment problem. It is difficult to find a reward which offers direct feedback and correlates with the actual target KPIs. An individual scheduling or dispatching decision might have long lasting effects which cannot be seen in the immediate returns of the reward. Furthermore, the many stochastic effects and dynamic nature of the system overlay the reward signal with noise. Lastly, the environment

is practically not fully observable if we consider all relevant constraints and details. In case we are using only a partial observation of the environment's state, we violate the Markov-Property.

We can circumvent all of the above problems, especially the credit-assignment problem, by using a black-box-optimization-algorithm which only requires one reward at the end of an episode. In this case we do not have to focus on reward shaping but we can directly return the KPIs of interest at the end of each iteration. [7] have shown Evolution Strategies (ES) to be a scalable alternative to policy-gradient-based RL. It is able to optimize smaller Neural Networks without the intrinsic problems of policy-gradient methods described above. It can be highly parallelized and offers good scalability with the number of utilized CPU cores. by combining ES with Neural Networks, it becomes a method within the area of Neuroevolution. CMA-ES is an even more effective enhancement of ES which takes into account the covariance among the parameters under optimization. [56] show its effectiveness for the SFSP in a real world industrial setting.

B. Master Planning for Semiconductor Supply Chain Simulation

1) *Problem Description*: Semiconductors are essential components in modern electronics, such as computers or smartphones. Their manufacturing is intricate, involving multiple stages that require precise coordination and efficient planning across various production sites. To manage this complexity, supply chain master planning is employed to coordinate resources and optimize supply, production, and distribution to meet demand effectively. Traditionally, this planning relies on linear programming (LP) models which are unable to deal with uncertainty. To analyze the effects of uncertainty on the supply chain, rapid event simulation is performed, where a host of different scenarios are simulated and their effects can be analyzed. A supply chain simulation model needs a policy model in the simulation, to determine the quantities of each product that should be produced in each period. Thus a policy model needs to be able to solve the production inventory control problem. The specific production inventory control problem addressed in this section deals with capacity sharing, dual sourcing systems. This means that each product can be produced using multiple different combinations of machine groups, and these combinations may share capacities. Furthermore, since rapid event simulation is the goal, LP cannot be used as a policy model, as a single LP run can take several hours, and in a simulation dozens of sequential LP runs would need to be performed. A good policy model needs to have short computation times, such that a wide range of scenarios can be explored, whilst approximating optimal solutions at the same time, to reflect reality.

2) *Heuristics*: A wide range of algorithms has been developed to achieve good solutions for the inventory control problem. Base-stock policies offer rapid execution and can yield near-optimal solutions in relatively simple settings with limited uncertainty [60]. More sophisticated heuristics that

TABLE II
EXAMPLES OF RL APPROACHES APPLIED TO SCHEDULING PROBLEMS WITH THEIR RESPECTIVE MDP CHARACTERISTICS.

Publication	RL Algorithm	Action Space	Observation Space	Reward
[47]	P3O	Lot-machine assignment	Graph based schedule	Makespan
[48]	DQN	Lot-machine assignment	Graph based schedule	Makespan
[49]	DQN	Lot sequence	Lot features	Utilization
[56]	CMA-ES	Lot sequence	Lot features	Tardiness/Throughput
[53]	PPO	Lot-machine assignment	Lot-machine features	Makespan
[54]	DDPG	Dispatching rules	Lot-machine features	Makespan
[50]	DQN	Dispatching rules	Production status	Tardiness

have been applied to supply chain simulation optimization include genetic algorithms and meta-heuristics [61]. Recently deep reinforcement learning (DRL) has shown considerable promise in inventory control applications [62]. Whilst a wide array of DRL algorithms is available, proximal policy optimization (PPO) is widely considered to be easy to tune for such problems [63], though it may not be able to outperform heuristics in all circumstances [64]. Moreover, to prevent the state-action space from becoming prohibitively large in large-scale problems, PPO often requires customization of the method. An example is to reduce the action space by using knowledge of optimal solutions for small problem instances, and applying this knowledge to larger problem instances [65], [66]. In response to limitations of PPO, DRL algorithms tailored specifically to the challenges of inventory control have been proposed the difficulties of inventory control have recently been proposed [8], [67], which have been shown to support scalability when paired with algorithmic enhancements based on problem-specific insights [68]. DRL is able to learn adaptive, situation dependent strategies, in contrast to traditional models that often rely on static policies. The ability to learn adaptive strategies is especially useful in scenarios with non-stationary demand and/or complex supply chain dynamics [69], [70], [71].

3) *Implementation*: The implemented model addresses the capacitated production inventory control problem, striving to deliver solutions rapidly and with near-optimal quality. Consequently, a DRL approach utilizing the PPO algorithm is developed as the policy generator within the supply chain simulation framework. This DRL model receives input in the form of state information for each item, including available production capacities, current inventory levels, and forecasted demand for multiple upcoming periods. Based on this input, the model determines production decisions—specifically, the quantity of each product to produce and the resources or supply chains to employ. To enhance computational efficiency and manage complexity, both the action and state spaces are discretized. This discretization reduces the complexity and size of the problem, facilitating faster convergence of the DRL model toward an effective policy. Additionally, action masking is applied to exclude infeasible production decisions due to capacity constraints. Action masking identifies infeasible actions, setting their selection probability to zero, thereby preventing the DRL agent from exploring irrelevant or impossible state-

action combinations. This targeted exploration significantly accelerates the training and convergence process [66]. To assess the effectiveness of the implemented DRL model, it is bench-marked against a (S,Q) base-stock policy in a realistic multi-objective setting. This policy orders items in each period, until the inventory position is above the safety stock level S, it can only order in multiples of Q items. In situations with ample capacity, this policy will mimic the behavior of LP models closely.

4) *Results*: To compare the two approaches, fifty agents were trained using the same PPO algorithm and subsequently evaluated against a baseline base-stock policy. The DRL models consistently achieved higher cumulative rewards relative to the base-stock counterparts. In addition, the variance in overall performance was lower for the DRL models, indicating more stable aggregate outcomes. However, variance across individual objectives was notably higher, suggesting that DRL agents converged to multiple distinct strategies, each capable of attaining similar near-optimal rewards. This phenomenon can be attributed to the inherent stochasticity of DRL training, which arises from factors such as distributional action sampling, variability in the training environment, and the complexities of non-convex optimization landscapes [72]. While the ability to discover diverse high-performing strategies enhances the model's exploratory capacity, it comes at the cost of reduced transparency. In particular, the absence of a clear, explainable policy may undermine user trust in the model's recommendations [73].

Another finding is that the DRL agents effectively leverage the future demand forecast embedded within the state-space to anticipate and proactively plan for substantial fluctuations in demand. In contrast, the benchmark model lacks this capability; as a result, when demand shifts abruptly, its response lags, leading to suboptimal planning and stock-outs.

5) *Scalability*: The current implementation has been applied to a small subset of real-world data; however, for the model to provide tangible business value, it must be scaled to life-sized datasets. To achieve this, several strategies can be employed. Transfer learning offers a promising solution by allowing a model, trained on a generalized smaller dataset, to be adapted for larger datasets, thus reducing training time without compromising performance [74]. Initial experiments applying transfer learning to this problem have produced encouraging results. Furthermore, graph-based approaches, such as the

integration of graph databases and graph neural networks, can enhance both generalization and scalability, outperforming traditional neural networks that rely on relational databases [75]. Another viable strategy involves the use of mixture of expert (MoE) models. By activating only subsets of neurons relevant to specific product groups, MoE models enable a single DRL framework to efficiently manage extensive and diverse product portfolios. This approach avoids the added complexity of multi-agent systems while maintaining high levels of performance [76].

C. Reel Utilization in Production by 3D Printing

1) *Problem Description:* In industrial additive manufacturing (AM), production relies on filament reels as the primary consumable. Each job depletes a portion of a reel, with varying consumption amounts depending on component complexity and size. Efficient utilization of these reels is essential to minimize material waste, avoid mid-print interruptions, and reduce downtime caused by frequent reel replacement.

The underlying optimization challenge mirrors a dynamic variant of the Cutting Stock Problem (CSP), where each incoming component request must be assigned to one of the available reels, which may have partial remaining capacity. However, unlike classical CSPs, this is an online decision-making problem as component requests arrive sequentially in real time, and each assignment must be made in real time without knowledge of future demands. Furthermore, the objective function explicitly incorporates the cost associated with residual filament that remains unusable for subsequent print jobs due to insufficient remaining capacity.

2) *Heuristics:* Conventional industrial practice employs simple greedy heuristics, such as First Fit or Best Fit Decreasing, to assign jobs to reels. A commonly used rule prioritizes assigning a job to the smallest available reel that can accommodate it, thereby conserving larger reels for future tasks. While computationally efficient and interpretable, these heuristics disregard long-term consequences, such as the accumulation of unusable leftovers and the stochastic nature of job sequences. This often results in suboptimal resource utilization and increased material waste.

Offline strategies consider a list of component requests in advance and solve the problem with integer programming techniques. However, these methods are inapplicable in real-time production environments characterized by high product variability, dynamic demand, and uncertainty, where immediate, adaptive decision-making is required.

3) *Reinforcement Learning Approach:* To address these limitations, we model the reel assignment problem as a Markov Decision Process (MDP). The state space comprises the current filament levels of all active reels, together with the filament requirement of the incoming component request. The action space consists of assigning the component request to one of the available reels. The goal is to minimize average filament waste while ensuring that all jobs are assigned and completed.

We introduce an index-based heuristic policy, assigning to each possible reel capacity a computed priority index that

reflects its future cost given the current job and reel state. We employ Deep Controlled Learning [8], an approximate policy iteration framework, where the index policy is refined through repeated evaluation and improvement steps using Deep Reinforcement Learning techniques. This allows us to maintain a low-dimensional, interpretable control policy while incorporating long-term reward optimization.

4) *Results:* We assess the performance of our proposed approach through a comparative analysis involving three policies: a baseline random policy, the index policy derived from structural problem analysis, and a DRL-enhanced index policy obtained via approximate policy iteration. The random policy, which assigns jobs to reels uniformly at random, serves as a naive benchmark. The index policy introduces a rule-based reel selection that leverages the filament requirement distribution of the component requests to guide decisions. Finally, the DRL policy learns to adaptively refine this structure by estimating long-term value under uncertainty.

Experimental and theoretical results demonstrate that the index policy significantly outperforms the random baseline in terms of waste reduction. Furthermore, the DRL-enhanced policy consistently outperforms the static index policy.

5) *Research Implications:* This case study highlights the potential of integrating structural modeling with reinforcement learning to address dynamic, real-time scheduling challenges in resource-constrained manufacturing systems. The proposed index-based formulation, combined with DRL-driven policy refinement, offers a promising balance between interpretability, scalability, and performance. It demonstrates how reinforcement learning can be effectively tailored to discrete, resource-constrained environments by embedding problem structure into the policy design.

From a broader perspective, this work aligns with the objectives of Industry 5.0 by enabling intelligent and sustainable decision-making practices. In particular, it illustrates how adaptive control strategies can reduce material waste, enhance resource utilization, and support more flexible, demand-driven production systems in additive manufacturing.

D. Long-Term Rapid Scenario Planning in the Semiconductor Industry using DRL

1) *Problem Description:* In the semiconductor industry, effective supply chain planning is crucial given the complexity and global nature of manufacturing processes. Traditional planning methods often rely on Mathematical Programming (MP) models that provide optimal solutions but suffer from high computational complexity. At NXP Semiconductors N.V., the challenge is compounded when rapid adjustments are required due to unforeseen events, where planners manually update production plans generated by MP models. This manual approach frequently leads to suboptimal resource utilization and inefficiencies. Thus, there is a critical need for automated, near-optimal solutions that can swiftly respond to dynamic changes in supply chain conditions.

2) *Current Approaches and Limitations:* Presently, NXP utilizes MP models primarily for weekly planning. However, daily updates required by unexpected occurrences cannot be

efficiently managed by these models due to computational delays. Dispatching Prioritization Rules (DPRs), both generic and tailored, offer faster solutions but lack the capability for long-term, strategic decision-making, often leading to inefficiencies and underutilized resources. Consequently, rapid scenario planning at NXP requires an agile, scalable solution capable of handling dynamic conditions effectively.

3) **Deep Reinforcement Learning Approach:** To overcome these limitations, a Deep Reinforcement Learning (DRL) framework is proposed [55], leveraging its strength in handling large-scale, sequential decision-making problems. The approach is formalized as a Markov Decision Process (MDP), with clearly defined states, actions, and rewards:

- **State Space:** Incorporates detailed information regarding remaining capacities for each week and supply path, as well as unplanned orders categorized by priority and due date.
- **Action Space:** Defines actions as allocations of orders to specific supply paths and production weeks, considering capacities and Just-In-Time (JIT) scheduling constraints.
- **Reward Function:** Immediate feedback is provided based on lateness of scheduled orders and final cumulative shortages, weighted by customer priority.

Two key DRL algorithms were explored:

- 1) **Proximal Policy Optimization (PPO):** Balances exploration and exploitation with clipped policy updates.
- 2) **Deep Controlled Learning (DCL):** An approximate policy iteration algorithm [8] transforming RL into a supervised learning classification problem.

4) **Results:** The DRL framework, particularly the DCL algorithm, demonstrated significant advantages. In scenarios with varying degrees of stochasticity, DCL consistently matched or slightly outperformed tailored DPRs while drastically reducing runtime compared to MP models. For instance, with low randomness, DCL showed objective values comparable to tailored DPRs, while PPO's performance lagged behind. Under higher stochastic conditions, DCL maintained superior performance, reinforcing its robustness and adaptability to dynamic order distributions.

5) **Implications and Scalability:** The DRL-based solution proves especially effective for rapid, scenario-driven adjustments in supply chain planning. Its ability to adapt to stochastic environments ensures sustainable and efficient resource allocation. Although DRL requires substantial initial training, its subsequent operational deployment significantly reduces decision-making latency.

Future enhancements to this DRL approach could involve:

- Shortening planning horizons for improved learning efficiency.
- Integrating real-time order arrivals for dynamic responsiveness.
- Expanding supply path options to address the dimensionality challenge while retaining flexibility.

Overall, leveraging DRL in semiconductor supply chain planning presents a promising direction, effectively balancing long-term strategic objectives with the agility required for rapid decision-making.

E. Differentiable physics for robot motion planning

1) **Description:** This chapter introduces the application of differentiable physics (DP) for trajectory optimization in robotic (UR5) manipulation and provides a systematic comparison with reinforcement learning (RL) method Proximal Policy Optimization (PPO), a widely recognized baseline in robotic learning tasks. Using DP, the robotic arm adaptively learns to reach the goal state and optimizes control parameters through gradient-based learning. The optimization process is governed by a loss function measures the deviation between the current state and the target state, and gradients are propagated through the simulator to update the control variables.

The proposed algorithm is integrated into a broader robotic system that includes: a code synthesis tool for robot control, a robot control system that integrates machine vision, robot motion planning and execution, and 6D object pose estimation [25]. The algorithm is evaluated via comparison with Proximal Policy Optimization (PPO), a method widely recognized as one of the most suitable approaches for robotic learning tasks. DP-based training is expected to reduce the sim-to-real gap, produce smoother trajectories, and improve placement accuracy. Incorporating DP into the training process further facilitates rapid adaptation to new tasks, thereby enhancing the robustness and flexibility of motion planning in robotic manipulators.

2) **Task Setup and Simulation:** To assess the effectiveness of DP in improving task performance, two bottle-insertion scenarios are considered. The first scenario involves insertion into a stationary socket, while the second introduces a dynamic setting in which the socket moves along a conveyor belt and stops at unpredictable positions. This dual setup enables evaluation under both controlled and partially uncertain environmental conditions [24].

A UR5 robotic manipulator, pre-equipped with a bottle, initiates each task from a randomly sampled position above the conveyor system. This randomized initialization increases variability across trials and prevents overfitting to a fixed starting configuration. The interaction between the robotic arm and the environment is modeled using a PD simulator Nimble Physics (NP). At each control step, the current joint states of the robot, together with the position of the target socket, are provided as inputs to a feedforward neural network. The network outputs joint torques that drive the manipulator toward successful insertion. To promote smooth and adaptive control behavior, the network architecture incorporates sinusoidal input encoding, GELU activation functions, and instance normalization. These components contribute to stable gradient propagation and enable the generation of continuous and responsive control signals throughout task execution.

3) **Training:** The model is trained in the Nimble Physics simulator, allowing gradients to propagate over time using backpropagation through time. This allows the optimization process to account for long-horizon dependencies between control actions and the resulting system states. The objective function consists of multiple components designed to enforce task specific constraints. A primary term minimizes positional error between the bottle and the target socket, ensuring accurate alignment. An additional term maintains

upright orientation during insertion. A velocity-regularization component penalizes excessive end-effector speed, promoting controlled motion, while a smoothness term reduces abrupt variations in joint torques to generate continuous trajectories. The objective is defined by the composite loss function shown in the Fig. 2 :

$$\mathcal{L} = \mathcal{L}_{pos} + \lambda_o \mathcal{L}_{orien} + \lambda_c \mathcal{L}_{contact} + \lambda_v \mathcal{L}_{vel}.$$

Each term of the loss function enforces a task-specific constraint. The position term is defined as

$$\mathcal{L}_{pos} = \|P_{bottle} - P_{target}\|^2,$$

where P_{bottle} denotes the position of the bottle and P_{target} represents the position of the socket target. This term minimizes positioning error. The orientation term maintains the correct upright orientation:

$$\mathcal{L}_{orien} = \|R_{bottle} - R_{upright}\|^2,$$

where R_{bottle} is the current orientation and $R_{upright}$ denotes the desired upright configuration. To address contact interactions, rigid collision responses are replaced with differentiable penalty functions:

$$\mathcal{L}_{contact} = \phi(d),$$

where d denotes the signed distance between interacting bodies and ϕ is a smooth penalty function. This "soft-contact" formulation preserves gradient flow while guiding the bottle toward insertion from above without introducing non-differentiable discontinuities typical of rigid-body contacts. To promote stable motion, velocity regularization is introduced:

$$\mathcal{L}_{vel} = \|\mathbf{v}_e\|^2,$$

where $\mathbf{v}_e$ represents the end-effector velocity. To address contact interactions, rigid collision responses are replaced with differentiable penalty functions. This soft-contact formulation preserves gradient flow while guiding the bottle toward insertion from above without introducing non-differentiable discontinuities typical of rigid-body contacts.

Optimization is performed over 500,000 training iterations using the Adam optimizer, enabling stable convergence under gradient-based updates.

4) *Results and Analysis:* The robot is trained using the DP and RL approaches in two variant tasks: with a fixed socket and a moving socket. In both cases, the objective is to insert a bottle with high precision, speed, and stability (see in the Fig. 3)[24].

In the fixed socket scenario, the controller trained in DP achieved a success rate of 94%, compared to 92% for RL. The DP method also showed better accuracy, with an average positional error of 1.21 mm (vs 4.32 mm for RL) and an orientation deviation of 0.21° (vs 0.58° for RL). Additionally, the DP approach produced slightly shorter and smoother trajectories (0.35 m vs. 0.38 m). In the more challenging task of moving the socket, the DP controller maintained a success rate of 84%, which is close to the 87% achieved by RL. This is explained by the fact that RL training used a much larger number of steps (12 million) compared to the DP method

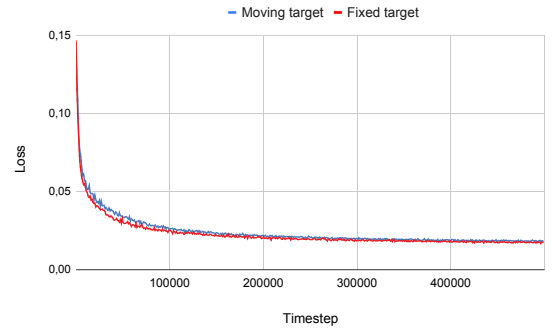


Fig. 2. DP training loss: red fixed socket scenario; blue moving socket scenario.

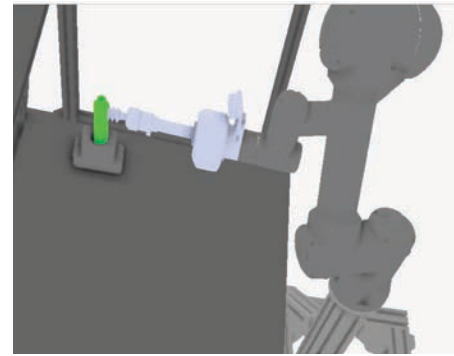


Fig. 3. Simulation of a UR5 robot arm that needs to insert a bottle into a designated socket.

(500K). The DP model was trained from scratch for both scenarios (500k iterations each). In contrast, the RL model was first trained on the fixed-socket task for 4 million steps and then fine-tuned for an additional 8 million steps on the moving-socket task.

However, the DP model again outperformed in accuracy, showing an average positional error of 1.51 mm (compared to 5.36 mm) and an orientation error of 0.23° (compared to 0.68°). DP trajectories were also more efficient, averaging 0.36 m compared to 0.43 m with RL.

These results highlight that while RL can achieve comparable success rates, DP consistently yields more precise, stable, and physically realistic trajectories, with lower deviation and faster per-action computation (0.3 ms), making it especially suitable for real-time deployment.

5) *Conclusions:* The DP training approach enables efficient learning of precise and physically consistent motion plans. Unlike RL, it requires less training time and provides more stable trajectories. This positions DP as a practical tool for industrial tasks involving contact-rich dynamic manipulation, with ongoing work focusing on integrating camera-based perception and extending the method to real-world hardware.

IV. IDENTIFICATION OF THE RESEARCH GAPS – FOR
INDUSTRY 4.0/5.0

Despite the remarkable progress in applying Deep Reinforcement Learning (DRL), Evolutionary Strategies (ES), and Differentiable Physics (DP) to industrial scheduling, control, and supply chain optimization, several key research gaps remain, particularly in the context of Industry 4.0 and 5.0:

1. Lack of Generalization Across Use Cases and Environments

Many DRL models and heuristic optimizations are highly tuned to specific problem instances or environments. This limits their transferability across different manufacturing setups, supply chain configurations, or robotic systems. There is a clear lack of reusable software, as well as techniques for making algorithms broadly applicable, in this domain.

2. Integration of Human-AI Collaboration

Industry 5.0 emphasizes human-centric design and human-AI collaboration. However, most current DRL or optimization-based methods operate as black-box solvers without mechanisms for interpretability, controllability, or integration with human decision-making. Research is needed into hybrid frameworks that allow operators to guide, intervene in, or understand the reasoning of the algorithm.

3. Scalability and Real-Time Constraints

As demonstrated in supply chain and scheduling use cases, scalability remains a limitation. While such limitations have been overcome in related domains [77], there remains a need for hardware-efficient learning frameworks and scalable DRL techniques.

V. CONCLUSION

This chapter has reviewed the recent advances in applying Deep Reinforcement Learning, Evolutionary Strategies, and Differentiable Physics to industrial scheduling, supply chain planning, and robotic control. We explored the foundational principles of these methods and showcased their application in high-impact use cases, from wafer lot scheduling in semiconductor manufacturing to supply chain master planning, 3D printing reel optimization, and robot motion planning.

These approaches promise more adaptive, scalable, and robust decision-making systems that align with the evolving goals of Industry 4.0 and 5.0—namely, increased automation, flexibility, and human-machine collaboration. However, transitioning from academic success to industrial impact demands further progress in several key areas: improving generalization, enhancing interpretability, reducing simulation dependence, and integrating uncertainty-aware learning.

The combination of DRL, ES, and DP offers a powerful toolkit. When aligned with domain expertise and industry requirements, these technologies can shift traditional optimization paradigms toward autonomous, self-adaptive, and intelligent industrial systems. As industrial AI evolves, continued cross-disciplinary collaboration between AI researchers, engineers, and domain experts will be crucial to closing remaining gaps and ensuring sustainable, human-centered innovation in manufacturing and logistics.

REFERENCES

- [1] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski et al., “Human-level control through deep reinforcement learning,” *nature*, vol. 518, no. 7540, pp. 529–533, 2015.
- [2] H. Van Hasselt, A. Guez, and D. Silver, “Deep reinforcement learning with double q-learning,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 30, no. 1, 2016.
- [3] R. J. Williams, “Simple statistical gradient-following algorithms for connectionist reinforcement learning,” *Machine learning*, vol. 8, pp. 229–256, 1992.
- [4] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, “Policy gradient methods for reinforcement learning with function approximation,” *Advances in neural information processing systems*, vol. 12, 1999.
- [5] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, “Asynchronous methods for deep reinforcement learning,” in *International conference on machine learning*. Pmlr, 2016, pp. 1928–1937.
- [6] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, “Trust region policy optimization,” in *International conference on machine learning*. PMLR, 2015, pp. 1889–1897.
- [7] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” *CoRR*, vol. abs/1707.06347, 2017.
- [8] T. Temizöz, C. Imdahl, R. Dijkman, D. Lamghari-Idrissi, and W. van Jaarsveld, “Deep controlled learning for inventory control,” *European Journal of Operational Research*, 2025. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0377221725000463>
- [9] X. Chen, D. Diao, H. Chen, H. Yao, H. Piao, Z. Sun, Z. Yang, R. Goebel, B. Jiang, and Y. Chang, “The sufficiency of off-policy and soft clipping: Ppo is still insufficient according to an off-policy measure,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 6, 2023, pp. 7078–7086.
- [10] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, “Continuous control with deep reinforcement learning,” *arXiv preprint arXiv:1509.02971*, 2015.
- [11] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, “Deterministic policy gradient algorithms,” in *International conference on machine learning*. Pmlr, 2014, pp. 387–395.
- [12] S. Fujimoto, H. Hoof, and D. Meger, “Addressing function approximation error in actor-critic methods,” in *International conference on machine learning*. PMLR, 2018, pp. 1587–1596.
- [13] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, “Soft actor-critic: Offpolicy maximum entropy deep reinforcement learning with a stochastic actor,” in *International conference on machine learning*. Pmlr, 2018, pp. 1861–1870.
- [14] R. Newbury, J. Collins, K. He, J. Pan, I. Posner, D. Howard, and A. Cosgun, “A review of differentiable simulators,” *IEEE Access*, 2024.
- [15] Y. Hu, L. Anderson, T.-M. Li, Q. Sun, N. Carr, J. Ragan-Kelley, and F. Durand, “DiffTaichi: Differentiable programming for physical simulation,” *arXiv preprint arXiv:1910.00935*, 2019.
- [16] J. Heeg, Y. Song, and D. Scaramuzza, “Learning quadrotor control from visual features using differentiable simulation,” *arXiv preprint arXiv:2410.15979*, 2024.
- [17] M. Lutter, J. Silberbauer, J. Watson, and J. Peters, “Differentiable physics models for real-world offline model-based reinforcement learning,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 4163–4170.
- [18] F. de Avila Belbute-Peres, K. Smith, K. Allen, J. Tenenbaum, and J. Z. Kolter, “End-to-end differentiable physics for learning and control,” *Advances in neural information processing systems*, vol. 31, 2018.
- [19] K. Freivalds, L. Leja, and O. Teikmanis, “Learning to move objects with fluid streams in a differentiable simulation,” in *2024 7th Iberian Robotics Conference (ROBOT)*. IEEE, 2024, pp. 1–7.
- [20] O. Teikmanis, L. Leja, and K. Freivalds, “Applying a differentiable physics simulation to move objects with fluid streams,” in *International Workshop on Embedded Digital Intelligence (IWEDI)*, 2023, p. 2.
- [21] L. Leja, O. Teikmanis, and K. Freivalds, “Shaping flames with differentiable physics simulations,” *Machine Learning and the Physical Sciences* at NeurIPS, vol. 202, pp. 1–6, 2024.
- [22] L. Leja and K. Freivalds, “Controlling shape of flame with differentiable physics,” *Engineering for Rural Development*, vol. 24, pp. 666–672.

- [23] L. Leja and U. Strautins, "Network model for thermal conversion of heterogeneous biomass granules," *Engineering for Rural Development*, pp. 737–743, 2021.
- [24] L. Leja, G. V. Strazds, C. Chronis, I. Varlamis, and K. Freivalds, "Differentiable physics training method for robot motion planning," *IFAC-PapersOnLine*, vol. 59, no. 27, pp. 220–225, 2025.
- [25] G. V. Strazds, K. Vitrake, G. Hollósi, L. Leja, A. Prieditis, D. Dupl, evska, M. Meiers, C. Chronis, I. Varlamis, G. Dimitrakopoulos et al., "Applied ai for adaptive robotic conveyor-feeding," *IFAC PapersOnLine*, vol. 59, no. 27, pp. 214–219, 2025.
- [26] C. D. Freeman, E. Frey, A. Raichuk, S. Girgin, I. Mordatch, and O. Bachem, "Brax-a differentiable physics engine for large scale rigid body simulation," in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021.
- [27] T. A. Howell, S. Le Cleac'h, J. Z. Kolter, M. Schwager, and Z. Manchester, "Dojo: A differentiable simulator for robotics," *arXiv preprint arXiv:2203.00806*, vol. 9, no. 2, p. 4, 2022.
- [28] K. Werling, D. Omens, J. Lee, I. Exarchos, and C. K. Liu, "Fast and feature-complete differentiable physics engine for articulated rigid bodies with contact constraints," in *Robotics: Science and Systems*, 2021.
- [29] Y. Hu, T.-M. Li, L. Anderson, J. Ragan-Kelley, and F. Durand, "Taichi: a language for high-performance computation on spatially sparse data structures," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 6, pp. 1–16, 2019.
- [30] M. Macklin, "Warp: A high-performance python framework for gpu simulation and graphics," <https://github.com/nvidia/warp>, 2022, nVIDIA GPU Technology Conference (GTC).
- [31] J. Collins, R. Brown, J. Leitner, and D. Howard, "Follow the gradient: Crossing the reality gap using differentiable physics (realitygrad)," *arXiv preprint arXiv:2109.04674*, 2021.
- [32] M. Geilinger, D. Hahn, J. Zehnder, M. Bächer, B. Thomaszewski, and S. Coros, "Add: Analytically differentiable dynamics for multi-body systems with frictional contact," *ACM Transactions on Graphics (TOG)*, vol. 39, no. 6, pp. 1–15, 2020.
- [33] S. Le Cleac'h, M. Schwager, Z. Manchester, V. Sindhwani, P. Florence, and S. Singh, "Single-level differentiable contact simulation," *IEEE Robotics and Automation Letters*, vol. 8, no. 7, pp. 4012–4019, 2023.
- [34] I. Georgiev, K. Srinivasan, J. Xu, E. Heiden, and A. Garg, "Adaptive horizon actor-critic for policy learning in contact-rich differentiable simulation," *arXiv preprint arXiv:2405.17784*, 2024.
- [35] G. Tembey, A. Dahik, C. Richard, and V. Rastogi, "Navigating the costly economics of chip making," September 2023, [Online; accessed on 21-October-2024]. [Online]. Available: <https://www.bcg.com/publications/2023/navigating-the-semiconductor-manufacturing-costs>
- [36] L. Mönch, J. W. Fowler, and S. J. Mason, *Production Planning and Control for Semiconductor Wafer Fabrication Facilities: Modeling, Analysis, and Systems*, ser. *Operations Research/Computer Science Interfaces Series*. New York: Springer, 2013, vol. 52.
- [37] M. Umlauf, M. Gojkovic, K. Harshina, K. Majbour, and M. Schranz, "Bees, bats and glowworms: Swarm algorithms for optimizing industrial plants from the bottom-up," in *Agents and Artificial Intelligence - 15th International Conference, ICAART 2023, Lisbon, Portugal, February 22-24, 2023, Revised Selected Papers*, ser. *Lecture Notes in Computer Science*, A. P. Rocha, L. Steels, and H. J. van den Herik, Eds., vol. 14546. Springer, 2023, pp. 3–25.
- [38] R. Ali, S. Qaiser, M. El-Kholany, P. Eftekhari, M. Gebser, S. Leitner, and G. Friedrich, "A greedy search based ant colony optimization algorithm for large-scale semiconductor production," in *Proceedings of the 14th International Conference on Simulation and Modeling Methodologies, Technologies and Applications*. SCITEPRESS - Science and Technology Publications, 2024, p. 138–149. [Online]. Available: <http://dx.doi.org/10.5220/0012813100003758>
- [39] B. Kovács, P. Tassel, and M. Gebser, "Optimizing dispatching strategies for semiconductor manufacturing facilities with genetic programming," in *Proceedings of the Genetic and Evolutionary Computation Conference*, ser. *GECCO '23*. New York, NY, USA: Association for Computing Machinery, 2023, p. 1374–1382.
- [40] G. Da Col and E. Teppan, "Google vs ibm: A constraint solving challenge on the job-shop scheduling problem," *Electronic Proceedings in Theoretical Computer Science*, vol. 306, p. 259–265, Sep. 2019.
- [41] G. Da Col and E. C. Teppan, "Industrial-size job shop scheduling with constraint programming," *Operations Research Perspectives*, vol. 9, p. 100 249, 2022. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S2214716022000215>
- [42] A. M. Ham and E. Cakici, "Flexible job shop scheduling problem with parallel batch processing machines: MIP and CP approaches," *Comput. Ind. Eng.*, vol. 102, pp. 160–165, 2016.
- [43] A. Ham and J. Fowler, "Constraint programming approach for scheduling batch operations with incompatible job families in fab and compatible job families in backend: Fa: Factory automation," in *2017 28th Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC)*, 2017, pp. 294–297.
- [44] A. Klemmt, J. Kutschke, and C. Schubert, "From dispatching to scheduling: Challenges in integrating a generic optimization platform into semiconductor shop floor execution," in *2017 Winter Simulation Conference, WSC 2017, Las Vegas, NV, USA, December 3-6, 2017*. IEEE, 2017, pp. 3691–3702.
- [45] R. Ali, M. M. El-Kholany, and M. Gebser, "Flexible job-shop scheduling for semiconductor manufacturing with hybrid answer set programming (application paper)," in *International Symposium on Practical Aspects of Declarative Languages*. Springer, 2023, pp. 85–95.
- [46] D. L. Applegate and W. J. Cook, "A computational study of the job-shop scheduling problem," *INFORMS Journal on Computing*, vol. 3, no. 2, pp. 149–156, 1991.
- [47] L. Hua, H. Liu, and Y. Pan, "Solving industrial chain job scheduling problems through a deep reinforcement learning method with decay strategy," *Information Sciences*, vol. 702, p. 121906, 2025. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0020025525000386>
- [48] Y. Wang, H. Liu, W. Zheng, Y. Xia, Y. Li, P. Chen, K. Guo, and H. Xie, "Multi-objective workflow scheduling with deep-q-network-based multiagent reinforcement learning," *IEEE Access*, vol. 7, pp. 39 974–39 982, 2019.
- [49] B. Waschneck, A. Reichstaller, L. Belzner, T. Altenmüller, T. Bauernhansl, A. Knapp, and A. Kyek, "Optimization of global production scheduling with deep reinforcement learning," *Procedia CIRP*, vol. 72, pp. 1264–1269, 2018, 51st CIRP Conference on Manufacturing Systems. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S221282711830372X>
- [50] S. Luo, "Dynamic scheduling for flexible job shop with new job insertions by deep reinforcement learning," *Applied Soft Computing*, vol. 91, p. 106208, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1568494620301484>
- [51] C. Zhang, W. Song, Z. Cao, J. Zhang, P. S. Tan, and C. Xu, "Learning to dispatch for job shop scheduling via deep reinforcement learning," 2020. [Online]. Available: <https://arxiv.org/abs/2010.12367>
- [52] W. Song, X. Chen, Q. Li, and Z. Cao, "Flexible job-shop scheduling via graph neural network and deep reinforcement learning," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 2, p. 1600–1610, 2023, cited by: 143; All Open Access, Green Open Access. [Online]. Available: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85134304558&doi=10.1109%2FTHI.2022.3189725&partnerID=40&m d5=ea3b88863f1c3e57663a04ad2bab1c33>
- [53] L. Wang, X. Hu, Y. Wang, S. Xu, S. Ma, K. Yang, Z. Liu, and W. Wang, "Dynamic job-shop scheduling in smart manufacturing using deep reinforcement learning," *Computer Networks*, vol. 190, p. 107969, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1389128621001031>
- [54] C.-L. Liu, C.-C. Chang, and C.-J. Tseng, "Actor-critic deep reinforcement learning for solving job shop scheduling problems," *IEEE Access*, vol. 8, pp. 71 752–71 762, 2020.
- [55] B. de Jong, K. Schelthoff, R. L. Bianco, and W. van Jaarsveld, "Longterm rapid scenario planning in the semiconductor industry using deep reinforcement learning," in *2024 Winter Simulation Conference (WSC)*. IEEE, 2024, pp. 1818–1829.
- [56] P. Stöckermann, A. Immordino, T. Altenmüller, G. Seidel, M. Gebser, P. Tassel, C. W. Chan, and F. Zhang, "Dispatching in real frontend fabs with industrial grade discrete-event simulations by deep reinforcement learning with evolution strategies," 2023, p. 3047–3058.
- [57] S. Greydanus, A. Koul, J. Dodge, and A. Fern, "Visualizing and understanding atari agents," 2018. [Online]. Available: <https://arxiv.org/abs/1711.00138>
- [58] A. Kuhnle, J. Kaiser, F. Theiß, N. Stricker, and G. Lanza, "Designing an adaptive production control system using reinforcement learning," *Journal of Intelligent Manufacturing*, vol. 32, no. 3, pp. 855–876, 2021.
- [59] Z. Juozapaitis, A. Koul, A. Fern, M. Erwig, and F. Doshi-Velez, "Explainable reinforcement learning via reward decomposition," 2019.
- [60] F. Chen, Y. Lu, and M. Xu, "Optimal inventory control policy for periodic-review inventory systems with inventory-level-dependent demand," *Naval Research Logistics*, vol. 59, pp. 430–440, 09 2012.

- [61] H. Jalali and I. V. Nieuwenhuyse, "Simulation optimization in inventory replenishment: a classification," *IIE Transactions*, vol. 47, no. 11, pp. 1217–1235, 2015.
- [62] R. N. Boute, J. Gijsbrechts, W. van Jaarsveld, and N. Vanvuchelen, "Deep reinforcement learning for inventory control: A roadmap," *European Journal of Operational Research*, vol. 298, no. 2, pp. 401–412, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0377221721006111>
- [63] F. S. Francesco Stranieri and C. Kouki, "Performance of deep reinforcement learning algorithms in two-echelon inventory control systems," *International Journal of Production Research*, vol. 62, no. 17, pp. 6211–6226, 2024. [Online]. Available: <https://doi.org/10.1080/00207543.2024.2311180>
- [64] I. Kaynov, M. van Knippenberg, V. Menkovski, A. van Breemen, and W. van Jaarsveld, "Deep reinforcement learning for one-warehouse multi-retailer inventory management," *International Journal of Production Economics*, vol. 267, p. 109 088, 2024.
- [65] L. v. Hezewijk, N. Dellaert, T. v. Woensel, and N. Gademann, "Using the proximal policy optimisation algorithm for solving the stochastic capacitated lot sizing problem," *International Journal of Production Research*, vol. 61, no. 6, pp. 1955–1978, 2023. [Online]. Available: <https://doi.org/10.1080/00207543.2022.2056540>
- [66] S. Huang and S. Ontañón, "A closer look at invalid action masking in policy gradient algorithms," *The International FLAIRS Conference Proceedings*, vol. 35, May 2022. [Online]. Available: <http://dx.doi.org/10.32473/flairs.v35i.130584>
- [67] P. Harsha, A. Jagmohan, J. Kalagnanam, B. Quanz, and D. Singhvi, "Deep policy iteration with integer programming for inventory management," *Manufacturing & Service Operations Management*, 2025.
- [68] T. van Dijk, T. Fleuren, T. Temizoz, Y. Merzifonluoglu, M. Hendriks, and W. van Jaarsveld, "Inventory planning in capacitated high-tech assembly systems under non-stationary demand," *Available at SSRN 4843271*, 2024.
- [69] B. Rolf, I. Jackson, M. Müller, S. Lang, T. Reggelin, and D. Ivanov, "A review on reinforcement learning algorithms and applications in supply chain management," *International Journal of Production Research*, vol. 61, no. 20, pp. 7151–7179, 2023. [Online]. Available: <https://doi.org/10.1080/00207543.2022.2140221>
- [70] L. van Hezewijk, N. P. Dellaert, and W. L. van Jaarsveld, "Scalable deep reinforcement learning in the non-stationary capacitated lot sizing problem," *International Journal of Production Economics*, vol. 284, p. 109601, 2025.
- [71] T. Temizöz, C. Imdahl, R. Dijkman, D. Lamghari-Idrissi, and W. van Jaarsveld, "Zero-shot generalization in inventory management: Train, then estimate and decide," *arXiv preprint arXiv:2411.00515*, 2024.
- [72] A. Choromanska, M. Henaff, M. Mathieu, G. B. Arous, and Y. LeCun, "The loss surfaces of multilayer networks," 2015. [Online]. Available: <https://arxiv.org/abs/1412.0233>
- [73] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," 2019. [Online]. Available: <https://arxiv.org/abs/1811.10154>
- [74] F. Ezzahra Achamrah, "Leveraging transfer learning in deep reinforcement learning for solving combinatorial optimization problems under uncertainty," *IEEE Access*, vol. 12, pp. 181 477–181 497, 2024.
- [75] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, C. Gulcehre, F. Song, A. Ballard, J. Gilmer, G. Dahl, A. Vaswani, K. Allen, C. Nash, V. Langston, C. Dyer, N. Heess, D. Wierstra, P. Kohli, M. Botvinick, O. Vinyals, Y. Li, and R. Pascanu, "Relational inductive biases, deep learning, and graph networks," 2018. [Online]. Available: <https://arxiv.org/abs/1806.01261>
- [76] O. Sanseviero, L. Tunstall, P. Schmid, S. Mangrulkar, Y. Belkada, and P. Cuenca, "Mixture of Experts Explained," <https://huggingface.co/blog/moe>, December 2023, accessed 2025-04-22.
- [77] J. F. van der Haar, W. van Jaarsveld, R. J. Basten, and R. N. Boute, "Industrializing deep reinforcement learning for operational spare parts inventory management," *Available at SSRN 4999374*, 2024.

Robotics, Automation, and Process Management for Industrial Operations

Review and Results from Real-World Use Cases Applying Machine Learning and Alternative Approaches

Fabian Lindner^{1,2}, Iraklis Varlamis³, Katrīna Viltrake⁴, Karl-Benedikt Reith⁵, Sebastian Rank⁶,
Michał Affek⁷, Germar Schneider^{1,8}, Rosa Leschly⁹, Venugopal Ramadasu⁹, Gerald Reiner⁹,
Pascal Herfort¹, Guntis V. Strazds⁴, and Marek Tatara⁷

Abstract—As industrial systems become increasingly complex, dynamic, and data-rich, organizations face growing pressure to enhance efficiency, flexibility, and decision-making through intelligent automation. This work examines how robotics, automation, and process management enable these transformations in modern industrial operations through data-driven and machine learning-based approaches. It first reviews the current state of research and industrial practice to establish a foundation for understanding technological and methodological developments in highly automated manufacturing environments. Building on this foundation, the article presents real-world industry use cases demonstrating how advanced analytics and machine learning improve material handling, robotics, intra-logistics automation, fleet management, and production operations. Particular attention is given to the operational deployment of machine learning, including machine learning operations (MLOps), and to the integration of engineering and management perspectives required to achieve robust and efficient manufacturing systems. The presented applications highlight improvements in system performance, operational efficiency, and decision support in complex industrial settings. By synthesizing practical insights and methodological advances, the article shows how these approaches can be transferred and extended across industries, contributing to more adaptive, intelligent, and resilient production systems.

Index Terms—Robotized picking and placing, travel time prediction, transportation system monitoring, rapid modeling, work-in-process level prediction, machine learning operations, semiconductor manufacturing, cosmetics industry

The project AIMS5.0 is supported by the Chips Joint Undertaking and its members, including the top-up funding by National Funding Authorities from involved countries under grant agreement no. 101112089.

¹ Zittau/Görlitz University of Applied Sciences, Germany

² HTW Dresden - University of Applied Sciences, Germany

³ Harokopio University of Athens, Greece

⁴ EDI - Institute of Electronics and Computer Science, Latvia

⁵ Fabmatics GmbH, Germany

⁶ TU Dresden, Germany

⁷ DAC.digital, Poland

⁸ Infineon Technologies Dresden AG & Co. KG, Germany

⁹ Vienna University of Economics and Business, Austria

I. INTRODUCTION

The successful coordination of value-adding transformation processes in manufacturing has always been a joint effort between operations management and engineering. This includes product design and development, as well as procurement, manufacturing, distribution, and sales processes [1], [2]. Therefore, both business and technical operations must be integrated through shared information systems [3] to enable efficient and effective production systems that reduce time and costs, increase quality, satisfy customer demands, and enable competitive advantages [4].

A fundamental approach to optimization is to focus on and depict the material, information, and cash flows of value and supply chains from a process perspective. This perspective highlights the partners, steps, interfaces, and transformations involved — from the supply of raw materials to the delivery of finished products to the end customer.

The increasing complexity, variability, and speed of modern industrial systems require tighter integration of management and engineering perspectives. Digitalization and machine learning (ML) have emerged as powerful enablers of this integration. They provide tools to extract actionable insights from complex and dynamic environments, bridging the gap between operational and information technology (IT). ML enables predictive maintenance, dynamic scheduling, adaptive control, and real-time decision support. When combined with advanced robotics, automation, and intralogistics, ML-driven approaches open new opportunities to increase throughput, reduce variability, and improve the robustness of industrial operations.

Equally important, process management provides the organizational and methodological framework to ensure that technological advancements align with business objectives and operational constraints. The interplay of machine learning, robotics, and management practices is particularly relevant for highly automated industries such as semiconductor manufacturing, intralogistics, and process industries, where variability, high product complexity, and strict quality requirements present unique challenges.

This work presents several engineering and management approaches that use digitalization, machine learning, and deep learning tools in the European research and development project AIMS5.0 (*Artificial Intelligence for Manufacturing Leading to Sustainability and Industry 5.0*).¹ It shows the improvements of these approaches along their respective transformation processes, as depicted in Fig. 1.

The paper begins with a section on state-of-the-art approaches in the literature and industry regarding individual aspects, such as robotics, automation, and process management — especially material handling, intralogistics, and production management. The subsequent section presents approaches beyond the state of the art as developed in the project AIMS5.0 and applied in industries ranging from cosmetics to semiconductor manufacturing. Finally, the approaches, benefits, and challenges are compared and generalized to other industries. Outlooks on their further development and application are also provided.

II. LITERATURE REVIEW

This section reviews the state of the art in key technological and methodological domains that enable intelligent industrial operations. The objective is to provide a structured overview of foundational concepts, established approaches, and recent developments in operations management, robotics and material handling, automation and intralogistics, and machine learning operations (MLOps). These domains jointly form the technological and organizational basis for modern data-driven manufacturing systems.

To ensure a consistent perspective, each subsection introduces the scope and role of the respective domain, outlines core methods and technologies, discusses industrial applications and challenges, and highlights current research trends. This structure enables a systematic comparison of existing approaches and provides the conceptual foundation for the industrial applications presented in the subsequent sections.

A. Operations Management

a) Scope and Role: Production and process management addresses the systematic planning, organization, and optimization of value-adding activities in manufacturing. Its primary objective is to align resources — including machines, materials, labor, and information — with production goals such as cost reduction, lead time minimization, quality assurance, and operational flexibility. As such, operations management provides the conceptual and methodological backbone of modern manufacturing systems, bridging engineering practices on the shop floor with managerial decision-making at the organizational level.

¹<https://aims50.eu/>

b) Core Methods and Technologies: A wide range of approaches has been developed to model, analyze, and improve production systems. Classical methods include deterministic and stochastic modeling, queueing theory, linear programming, and discrete-event simulation, which provide analytical insights into system performance and constraints. Complementary methodological frameworks such as *Lean Production* [5], *Quick Response Manufacturing* [6], and *Factory Physics* [2] emphasize waste reduction, flow efficiency, and quantitative understanding of production dynamics.

More recent approaches focus on rapid modeling techniques [7], [8], which enable fast and flexible analysis of production systems under uncertainty by representing processes as networks of queues. These approaches address limitations of detailed simulation models, particularly high data requirements and long computation times, while maintaining sufficient accuracy for decision support.

The increasing availability of industrial data and advances in digitalization have further extended operations management methods through machine learning techniques. These enable data-driven forecasting, anomaly detection, adaptive scheduling, and real-time decision support, complementing traditional analytical and simulation-based approaches and supporting more agile and intelligent production systems.

c) Industrial Applications and Challenges: Effective production and process management is critical for product quality, reducing costs, and improving customer satisfaction across industrial domains [9]. Industrial applications include integrated product and process control for sustainable manufacturing [10], quality management systems in semiconductor production [11], and quality function deployment for aligning customer requirements with production processes [12].

However, industrial production systems are characterized by inherent variability arising from process disruptions, equipment failures, and supply chain fluctuations [2], [13]. Managing this variability requires appropriate buffering strategies involving inventory, capacity, or lead time adjustments, often captured by trade-off models such as the operations management triangle [14], [13], [15]. Modeling the dynamic interaction of variability remains challenging, particularly in complex and highly automated environments where detailed simulation models may be costly and time-consuming to implement.

d) Research Trends and Outlook: Recent research increasingly focuses on digital transformation and intelligent manufacturing, integrating artificial intelligence, Internet of Things technologies, and data-driven decision-making into production management systems [16], [17], [18]. These developments aim to enhance responsiveness, improve system efficiency, and enable adaptive production control. Despite significant progress, challenges remain in modeling complex system dynamics, managing heterogeneous data sources, and integrating advanced analytics into operational workflows, motivating the development of the approaches presented in this work.

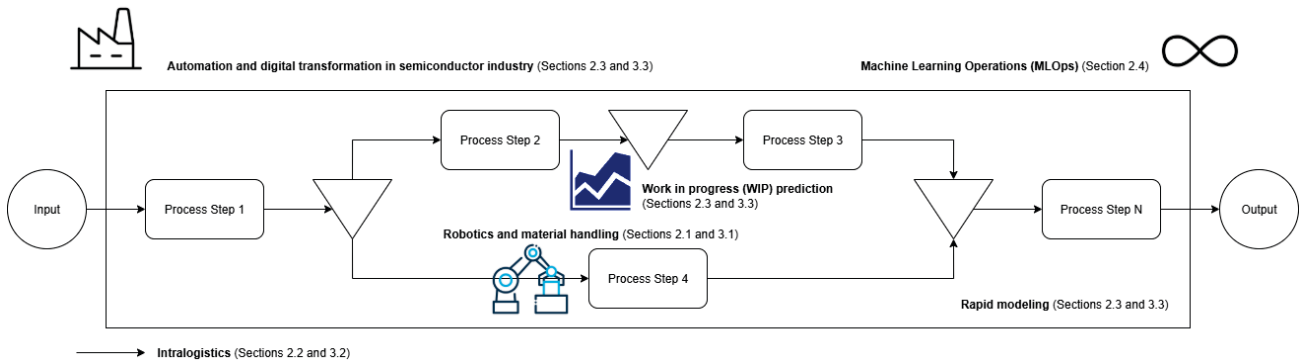


Fig. 1: Generic industrial transformation process and respective topics addressed in this article.

B. Robotics and Material Handling

a) Scope and Role: Material handling encompasses tasks such as intralogistics, machine tending, assembly line feeding, and warehouse operations. In the context of robotics, material handling refers to the use of autonomous or semi-autonomous robotic systems to move, manipulate, and organize materials within production and logistics environments [19]. Robotics is defined as the science and practice of designing, manufacturing, and applying robots [20].

Robotic material handling represents a subset of broader material handling equipment, which also includes conventional systems such as conveyors or palletizers [21], [22]. However, robotic solutions enable higher flexibility, adaptability, and automation, directly affecting throughput, quality, and operational safety. Consequently, robotic material handling has become a central component of modern manufacturing and logistics systems [21].

b) Core Methods and Technologies: Robotic material handling systems can be broadly categorized into mobile robots, robotic manipulators, and hybrid systems combining mobility and manipulation capabilities. Mobile robots transport materials across operational environments without being confined to fixed workspaces. These include automated guided vehicles (AGVs), autonomous mobile robots (AMRs), and unmanned aerial vehicles (UAVs). AGVs typically operate on predefined paths and execute repetitive tasks using programmed instructions with minimal human intervention [23], [24]. In contrast, AMRs navigate dynamically and make independent decisions based on environmental perception, enabling flexible operation in changing environments [23], [24]. UAVs extend material handling capabilities to aerial transport tasks, such as automated delivery of lightweight materials in construction environments [25], [26].

Robotic manipulators enable precise handling of objects and are widely used in machine tending and internal logistics applications [27]. Hybrid mobile manipulators combine transportation and manipulation capabilities by integrating robotic arms with mobile platforms, enabling flexible execution of complex tasks such as machine tending under uncertain conditions [28].

Recent developments further include collaborative robots designed for safe human-robot interaction in Industry 5.0 environments [19], soft robotic end-effectors for handling

delicate or deformable objects [29], and advanced perception and control technologies integrating vision systems, force sensing, motion planning, and multi-robot coordination.

c) Industrial Applications and Challenges: Robotic material handling systems are widely deployed in manufacturing plants, assembly lines, warehouses, and logistics facilities, where they support efficient transport of raw materials and finished goods between production stages. These systems improve operational efficiency, reduce manual labor requirements, and enhance workplace safety.

However, several challenges remain for industrial adoption. Handling complex or deformable objects requires advanced perception and control capabilities, while robust human-robot collaboration demands high safety standards and reliable interaction mechanisms. In addition, system integration, scalability, and performance limitations in dynamic environments continue to present technical and operational challenges.

d) Research Trends and Outlook: Current research focuses on increasing autonomy, adaptability, and intelligence in robotic material handling systems. Emerging directions include AI-driven perception and decision-making, improved manipulation of complex objects, enhanced human-robot collaboration, and more flexible mobile manipulation systems. Progress in robotics, materials science, and artificial intelligence is expected to further expand the capabilities of material handling systems and enable more adaptive and intelligent industrial operations.

C. Automation and Intralogistics

a) Scope and Role: Automation aims to improve efficiency, quality, safety, and cycle times in industrial production and logistics processes [30]. In the context of intralogistics, automation focuses on managing internal material flows, including handling and transportation tasks required to move loads between production resources. While handling involves storing, unpacking, or transferring goods over short distances, transportation covers longer-distance movements between locations such as storage systems and production tools. These functions are closely interconnected, particularly in highly automated environments such as semiconductor manufacturing, where Automated Material Handling Systems (AMHS) coordinate material flows across production facilities.

b) *Core Methods and Technologies:* Automation in intralogistics is commonly characterized by hierarchical control structures and levels of autonomy. A widely used classification distinguishes automation stages ranging from manual operation to fully autonomous systems [31], [32], alongside task levels covering execution, monitoring, control, and planning functions [33]. These classifications reflect a transition from centralized to decentralized decision-making in automated production environments.

In semiconductor manufacturing, automation technologies include automated process tools, standardized equipment interfaces, and integrated control architectures enabling automated transfer and transport operations [34], [35]. Modern high-volume fabrication facilities employ overhead transport (OHT) systems that enable fully automated wafer handling using standardized carriers such as Front Opening Unified Pods (FOUPs). Advanced system designs include unified rail layouts and buffer mechanisms that improve flexibility, throughput, and tool utilization [36], [37].

c) *Industrial Applications and Challenges:* Automated intralogistics systems are essential for maintaining high production efficiency, particularly in semiconductor fabrication facilities where equipment utilization and transport reliability directly affect operational performance. These systems support large-scale material flows, reduce manual intervention, and enable precise coordination between production scheduling and transport operations.

However, increasing system complexity introduces challenges related to planning, coordination, and control across multiple hierarchical levels. The integration of production management and material handling control requires standardized communication architectures and interoperable control systems [38], [39]. In addition, infrastructure costs, system scalability, and dynamic transport requirements may limit the economic efficiency of certain automation solutions, motivating the exploration of alternative technologies such as flexible floor-based vehicles.

d) *Research Trends and Outlook:* Current research focuses on increasing autonomy and intelligence in intralogistics systems through artificial intelligence and machine learning techniques [40]. These technologies enable decentralized decision-making, adaptive scheduling, and improved system optimization. At the same time, ongoing work addresses challenges related to system complexity, data integration, and robust control in highly automated production environments. Future developments are expected to further enhance flexibility, scalability, and economic efficiency in automated material flow systems.

D. Machine Learning Operations

a) *Scope and Role:* Machine Learning Operations (MLOps) integrates practices from software engineering, machine learning, and data engineering to manage the lifecycle of machine learning systems in production environments [41]. In industrial manufacturing, MLOps supports the deployment, monitoring, and maintenance of data-driven models, enabling reliable and scalable use of machine learning for tasks such as

machine vision, predictive analytics, and process optimization. Its primary objectives include rapid model iteration, automated deployment, and continuous performance monitoring.

b) *Core Methods and Technologies:* MLOps encompasses a set of processes covering data collection and pre-processing, model training and evaluation, deployment, and monitoring throughout the model lifecycle [42]. Similar to DevOps practices, MLOps employs Continuous Integration and Continuous Delivery (CI/CD) pipelines to enable automated updates and seamless deployment of machine learning models.

Key technical aspects include model versioning, performance evaluation, and retraining strategies to ensure model adaptability to changing operational conditions [43]. In industrial settings, MLOps systems must address constraints related to computational resources, real-time data processing, and integration of heterogeneous data sources, which pose additional challenges compared to traditional software environments.

c) *Industrial Applications and Challenges:* Industrial adoption of MLOps enables standardized and scalable management of machine learning systems across diverse application domains, including manufacturing, robotics, and digital technology systems [42]. However, implementation requires significant investment in infrastructure and specialized expertise, which may limit adoption in risk-sensitive production environments.

Major challenges include integrating heterogeneous data sources, ensuring model reliability in safety-critical systems, and maintaining performance under changing process conditions. Furthermore, the setup of fully operational MLOps pipelines requires substantial organizational and technical effort, particularly in established industrial environments with legacy systems.

d) *Research Trends and Outlook:* Current research focuses on standardizing MLOps practices, improving data pipeline integration, and developing scalable frameworks for industrial machine learning deployment. As machine learning applications continue to expand in manufacturing, MLOps is expected to play an increasingly important role in enabling reliable and maintainable AI-driven systems. Although adoption is expected to be gradual, the growing maturity of MLOps technologies supports their integration into industrial operations and motivates the development of practical deployment strategies.

III. INDUSTRIAL APPLICATIONS

In this section, several real-world applications are demonstrated, especially in the fields of robotics and material handling in the process industry, and concerning automation, intralogistics, and production management in semiconductor manufacturing.

A. Operations Management

1) *Use Case: Production Optimization in Semiconductor Manufacturing at Infineon Dresden:*

TABLE I
CLASSIFICATION MATRIX CONNECTING AUTOMATION STAGES AND TASK LEVELS IN INTRALOGISTICS (ADAPTED FROM [33]).

Task level	Automation Stages					
	Stage 0	Stage 1	Stage 2	Stage 3	Stage 4	Stage 5
Overall system planning (Level 4)	Manual decision making in overall system planning	Decision support system in overall system planning	Partly centralized decision making in overall system planning	Centralized decision making in overall system planning	Partly decentralized decision making in overall system planning	Decentralized decision making in overall system planning
Overall system control (Level 3)	Manual decision making in overall system control	Decision support system in overall system control	Partly centralized decision making in overall system control	Centralized decision making in overall system control	Partly decentralized decision making in overall system control	Decentralized decision making in overall system control
Process control and monitoring (Level 2)	Manual decision making in process control and monitoring	Decision support system in process control and monitoring	Partly centralized decision making in process control and monitoring	Centralized decision making in process control and monitoring	Partly decentralized decision making in process control and monitoring	Decentralized decision making in process control and monitoring
Device control and information processing (Level 1)	Manual search and transfer of information on demand	Assisted search and transfer of information on demand	Assisted and partly self-contained information acquisition and processing	Fully self-contained information acquisition, generation of information and interaction with central system	Fully self-contained information acquisition, generation of information and partly interaction with other system elements	Fully self-contained information acquisition, generation of information and interaction with other system elements
System execution and real time control (Level 0 and 1)	Manual execution in static environment	Manual assisted execution in static environment	Assisted and partly self-contained execution in static environment	Partly self-contained execution in infrequently changing environment	Fully self-contained execution in dynamic environment	Fully self-contained execution in complex and dynamic environment



Fig. 2: Inside the fully automated 200mm line at Infineon Dresden.



Fig. 3: IoT Technologies Used to Move Production to Industry 5.0.

a) Industrial Context and Problem Definition: Semiconductor manufacturing is characterized by high capital investment, strict quality requirements, and complex production processes. Efficient production and process management are therefore critical to ensure product quality, reduce costs, and maintain competitiveness. The Infineon Dresden wafer fabrication facility operates a highly automated 200 mm production line, established in 1995 and later transformed from a high-volume memory facility into a high-mix, high-volume logic factory. The line achieves an automation level of approximately 93%, enabling flexible production of hundreds of products (Fig. 2).

The facility relies on advanced automation, digitalization, and lean production principles to manage variability, optimize resource utilization, and ensure stable production performance. Increasing system complexity and data availability motivate the use of advanced analytics and machine learning methods to improve production planning, process control, and decision support.

b) Approach / Methodology: Production optimization at Infineon Dresden combines several complementary approaches. Factory automation integrates robotics, machine learning, and data analytics for real-time monitoring and

process optimization. Digitalization initiatives based on Industry 4.0 concepts enable the integration of cyber-physical systems, predictive analytics, and Internet of Things technologies to support data-driven decision-making (Fig. 3). Future developments aim to further integrate AI technologies and human-machine collaboration in Industry 5.0 environments [44].

Operational performance is further supported by lean manufacturing principles [45], Statistical Process Control (SPC), Advanced Process Control (APC), and Fault Detection Control (FDC). These methods enable continuous process monitoring, reduction of variability, and early detection of anomalies. Machine learning techniques complement these approaches by enabling predictive analytics and adaptive process optimization.

c) Experimental Setup / Implementation: The production environment is supported by a Manufacturing Execution System (MES) that serves as a central data infrastructure, collecting more than one billion sensor readings per day. The MES integrates data from process monitoring systems, equipment sensors, and production workflows, providing a foundation for advanced analytics and machine learning applications.

Process monitoring, quality control, and operational decision-making are implemented through integrated SPC, APC, and FDC systems combined with machine learning methods. The production environment also incorporates health and safety measures, including enclosed fabrication systems

and automated processes to minimize workplace risks [46], [47].

d) Results and Insights: The integration of automation, digitalization, and process control enables stable production performance, improved quality, and increased operational efficiency. Data-driven monitoring allows early detection of process deviations and supports continuous process improvement. The large-scale data infrastructure provides opportunities for advanced machine learning applications, enabling predictive maintenance, adaptive production planning, and improved decision support.

e) Implications and Transferability: This case demonstrates how the integration of automation, digitalization, and data-driven process management enables efficient operation of highly complex production systems. The combination of lean production, advanced process control, and machine learning provides a transferable framework for managing variability and improving operational performance in other highly automated manufacturing environments.

2) *Use Case: Rapid Modeling to Identify Process Variability:*

a) Industrial Context and Problem Definition: Efficient utilization of production capacity is essential in semiconductor manufacturing due to high investment costs and long scale-up timelines. In 2023, Infineon Dresden observed increasing work-in-process (WIP) levels and long waiting times in the production of wafers based on a specific power technology (PTX). The issue was particularly pronounced at a workstation referred to as Y, motivating a study to determine whether performance limitations were caused by insufficient capacity or high process variability.

b) Approach / Methodology: A rapid modeling approach based on queueing theory was implemented using the IMPAQT software framework [48]. Rapid modeling enables efficient simulation of production dynamics and evaluation of key performance indicators such as flow time, WIP, and utilization.

The analysis focused on workstation Y and an upstream workstation X to capture the impact of upstream variability. Since process and interarrival times were not exponentially distributed, the system was modeled as a G/G/m queue. Waiting times were estimated using Whitt's approximation [49], and equipment failure effects were incorporated following Segal and Whitt [50].

c) Experimental Setup / Implementation: Production data from 2023 were used to parameterize the model, including process times, arrival rates, and equipment availability. Only productive lots were considered. The simulation was conducted on a monthly basis to account for the expansion of workstation Y from one to two machines during the observation period.

d) Results and Insights: The analysis revealed high variability in process times and unscheduled downtime, particularly at workstation Y. This variability caused extended idle periods and long flow times of nearly two days per wafer.

Compared with theoretical capacity specifications, simulation results showed unexpected utilization patterns (Fig. 4). Utilization was higher during low throughput periods and lower during high throughput periods, indicating that perfor-

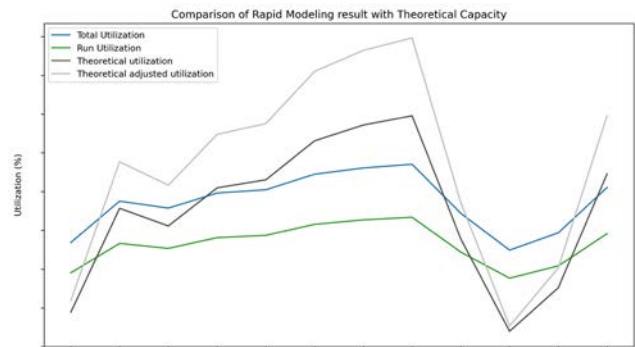


Fig. 4: Comparison of theoretical workstation Y utilization (with PTX technology lots) with rapid model results

mance limitations were primarily driven by variability rather than insufficient capacity.

Workstation Y was also more sensitive to demand fluctuations than workstation X, highlighting its role as a major source of production instability.

e) Implications and Transferability: Rapid modeling provides an efficient analytical tool for identifying variability-driven bottlenecks in complex production systems. The approach enables targeted improvement measures without requiring detailed simulation models and is transferable to other manufacturing environments facing similar capacity and variability challenges.

3) *Use Case: Prediction of Work-in-Process Levels Using Machine Learning:*

a) Industrial Context and Problem Definition: Factory simulations are widely used in semiconductor manufacturing for forecasting material flows and supporting capacity planning. However, existing simulations typically operate on daily or weekly horizons and do not provide sufficiently granular predictions for shop-floor decision-making. Accurate short-term forecasts of work-in-process (WIP) levels would enable operators to anticipate bottlenecks, balance workloads, and respond to production variability.

b) Approach / Methodology: The proposed solution applies machine learning methods for ultra-short-term forecasting of WIP levels based on real-time production data. The prediction task is formulated as a time-series regression problem. After data preprocessing following established guidelines [51], [52], multiple models including SARIMAX, XGBoost, LSTM, and CNN were evaluated [53], [54], [55].

An automated workflow was implemented within the Python ecosystem, covering data preprocessing, model training, hyperparameter optimization, and evaluation. Model performance was assessed using RMSE, MAE, and R^2 . Explainable AI methods such as SHAP values were applied to enhance interpretability and support human-AI interaction [56], [57].

c) Experimental Setup / Implementation: The models were trained using historical production data and evaluated in collaboration with factory experts. Visualization tools based on Plotly and Jupyter were developed to present predictions

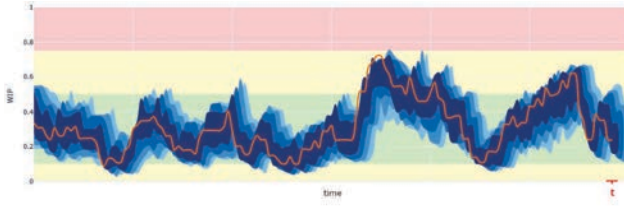


Fig. 5: Visualisation of WIP-development (orange) up to time t (red), alongside the corresponding forecast from $WIP_{t+0:30h}$ to $WIP_{t+2:00h}$ (blue with increasing brightness) at a confidence interval of 90% per forecast

and explanations to operators (Fig. 5, Fig. 6). Behavioral experiments measured user acceptance using the Weight on Advice (WOA) metric.

$$WOA = \frac{(\text{Initial Human Prediction} - \text{Adjusted Human Prediction})}{(\text{Initial Human Prediction} - \text{AI Prediction})}$$

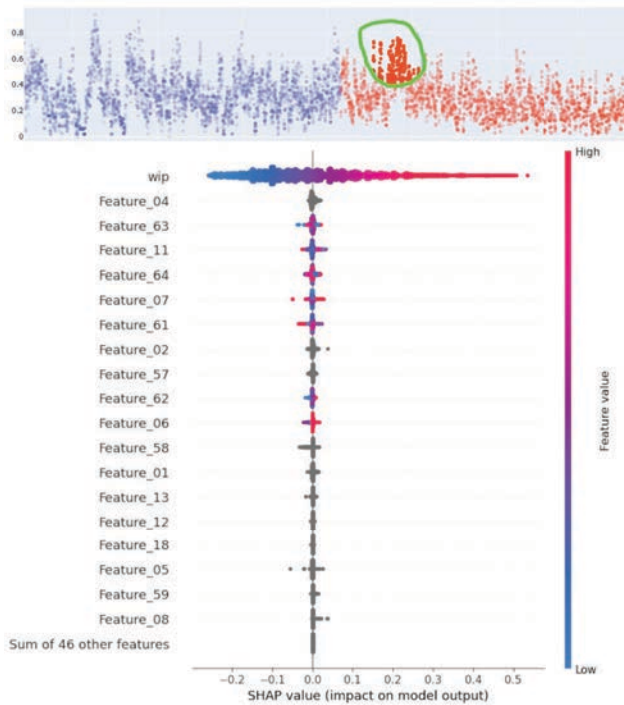


Fig. 6: Interactive dashboard-visualisation of the WIPprediction (red) and the SHAP-values (below) for selected data-points (green)

d) Results and Insights: The best-performing model (XGBoost) achieved an RMSE of 0.1035, MAE of 0.0813, and $R^2 = 0.5442$ for a prediction horizon of two hours. Although predictive accuracy remains limited, the results demonstrate the feasibility of short-term WIP forecasting and highlight the importance of data quality and feature selection.

Behavioral experiments ($N = 71$) showed that users relied more strongly on AI predictions when no explanations were provided ($WOA = 0.63$) than when explanations were included

($WOA = 0.58$), indicating that transparency influences trust and decision behavior.

e) Implications and Transferability: Machine learning-based WIP forecasting provides a promising approach for supporting operational decision-making in highly automated production environments. The integration of predictive models with explainable AI and human-AI interaction concepts supports practical adoption and offers transferable insights for decision-support systems in complex industrial settings.

B. Intralogistics

1) Use Case: Predicting Travel Times of AMR fleets:

a) Industrial Context and Problem Definition: This use case considers a fleet of free-ranging Automated Mobile Robots (AMRs) operating in a semiconductor fabrication environment. The robots transport carriers between input/output stations, storage areas, and production tools, and are coordinated by a fleet management system responsible for task assignment, routing, and congestion handling.

Fleet management in such environments involves highly interdependent decisions under dynamic conditions. Task allocation, route planning, and prioritization must be continuously updated to ensure efficient and collision-free operation. Semiconductor fabrication facilities operate under strict production schedules, where transport delays can lead to costly tool idle times and reduced throughput. Consequently, accurate prediction of AMR travel times is essential for maintaining operational efficiency.

Conventional travel time estimation methods based on average velocities or historical data provide limited accuracy, as they do not sufficiently account for real-time system conditions such as congestion or routing conflicts. These limitations motivate the development of data-driven prediction approaches that incorporate dynamic fleet interactions.

b) Approach / Methodology: A supervised learning approach is proposed to predict AMR travel times based on historical transportation data. A log file parser extracts structured input-output pairs from system logs, where input features describe the system state at a given time, including robot locations and destinations, while the target variable represents the observed travel time.

The prediction task is formulated as a regression problem that approximates the mapping from system state to expected travel time. The methodology incorporates both intermediate waypoints and final destinations to capture routing conflicts occurring during transport operations. The resulting model is integrated into the fleet management system to provide real-time predictions and can be periodically retrained using newly collected data.

c) Experimental Setup / Implementation: Training data were generated from several months of historical transportation logs in environments with medium-complexity layouts. The dataset included information on robot locations, destinations, and system states at corresponding timestamps. Various model configurations were evaluated, with a multilayer perceptron (MLP) achieving the best performance.

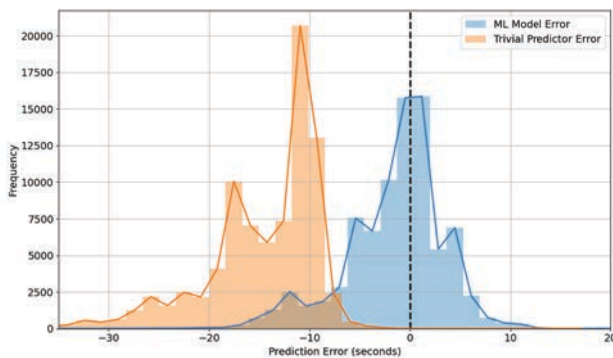


Fig. 7: Error distribution for travel time prediction with two different estimators

The model was deployed within the fleet management framework to estimate travel times dynamically and support task assignment and routing decisions.

d) Results and Insights: The trained model achieved a mean absolute error of approximately 3.7 seconds and an R^2 value of 0.82 on test data. Compared to a baseline estimator based on average speed, the prediction error was reduced from approximately 14.7 seconds.

The error distribution shown in Fig. 7 demonstrates that heuristic approaches systematically underestimate travel times, whereas the learned model captures congestion effects and fleet interactions more accurately.

Model performance was found to depend on data quality, layout characteristics, and traffic stability. High predictive accuracy was achieved in controlled environments with stable operating conditions.

e) Implications and Transferability: The results demonstrate that data-driven travel time prediction can significantly improve fleet management performance in automated intralogistics systems. The approach supports more efficient task allocation and routing decisions and can be transferred to other automated transport systems operating in dynamic industrial environments. Future work will investigate scalability to larger fleets, more complex layouts, and enhanced feature representations.

2) Use Case: Monitoring of Automated Material Handling Systems (AMHS):

a) Industrial Context and Problem Definition: Efficient operation of Automated Material Handling Systems (AMHS) is critical for semiconductor fabrication facilities, where reliable material transport directly affects production efficiency and tool utilization. Preventing AMHS downtime requires continuous monitoring of system components to detect malfunctions and initiate maintenance actions.

Condition monitoring typically relies on industrial sensor systems, which provide high-quality measurements but are costly to acquire and operate. This motivates the investigation of alternative low-cost sensing solutions. In particular, the suitability of consumer-grade sensors, such as those integrated into mobile devices, for monitoring AMHS components is evaluated.

Previous work has demonstrated the effectiveness of sensor-based condition monitoring for industrial systems [58], while highlighting the associated implementation costs [59]. Consumer devices equipped with embedded sensors have been successfully applied in domains such as healthcare and traffic monitoring [60], although their performance may be limited compared to industrial equipment [61]. Their applicability in semiconductor intralogistics environments remains largely unexplored.

b) Approach / Methodology: A modular sensor platform was developed to compare different sensing technologies for AMHS condition monitoring. The platform integrates multiple sensor systems, including an Android mobile phone, two single-board computers (Arduino and Raspberry Pi), and a professional industrial sensor system (Fig. 8). The collected acceleration data are analyzed in both time and frequency domains to assess measurement accuracy and reliability.

c) Experimental Setup / Implementation: The platform was tested in representative scenarios, including overhead hoist transportation (OHT) movements, automated guided vehicle (AGV) transport, and controlled experiments using a calibrated shaker. Sensor characteristics are summarized in Table II.

Experiments evaluated sensor performance under controlled oscillations (Fig. 9), shock events (Fig. 10), and real AMHS movements (Fig. 11). A lightweight Android application was developed to record and preprocess sensor data (Fig. 12), enabling deployment in cleanroom environments.

The platform was subsequently applied in operational tests involving FOUP transport via OHT systems and AGVs. Additional experiments combined acceleration and audio data to improve event detection through data fusion.

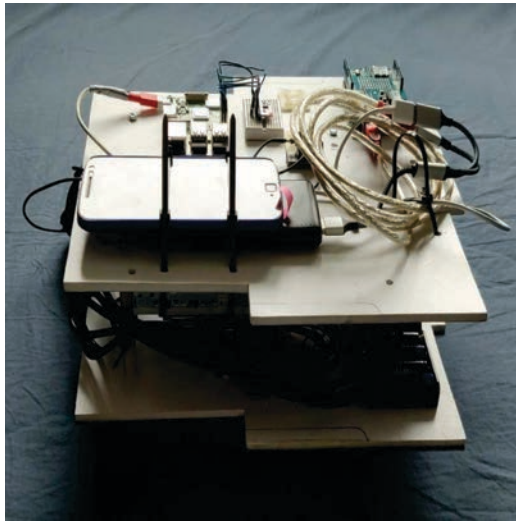
d) Results and Insights: The experiments showed that consumer-grade sensors can capture key dynamic events such as oscillations and shock loads with reasonable accuracy. The Android mobile phone provided measurement values comparable to the professional system in several scenarios, while the Raspberry Pi and Arduino sensors showed limitations in reliability and accuracy.

All sensors detected shock events with reproducible response times (Fig. 10), although frequency detection was limited by the sampling rate of mobile devices. During AGV operation, only the mobile phone and professional system produced plausible measurements (Fig. 11).

Operational tests in the cleanroom environment revealed no unexpected mechanical loads during transport operations (Fig. 13). Additional experiments demonstrated that combining acceleration and audio data enables detection of specific AMHS events such as FOUP door locking and hoisting processes (Fig. 14, Fig. 15).

e) Implications and Transferability: The results indicate that low-cost consumer sensors can provide useful monitoring capabilities for AMHS condition monitoring, particularly when combined with data fusion techniques. Although limitations remain in measurement precision and sampling rates, the approach offers a cost-effective alternative to traditional industrial sensing solutions.

The developed sensor platform and analysis workflow provide a transferable framework for condition monitoring in



(a) Sensor platform

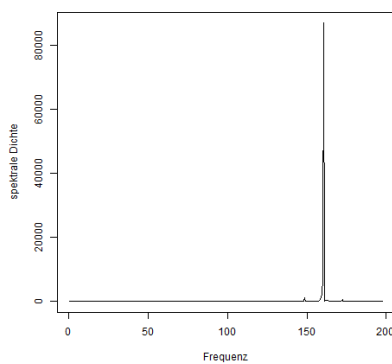


(b) Sensor platform in a FOUP

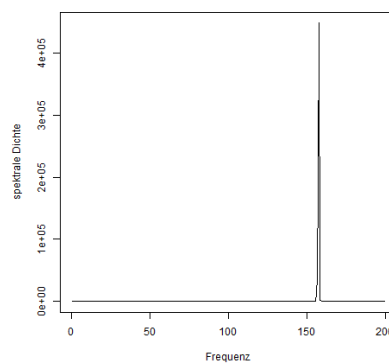
Fig. 8: Sensor platform with different sensor systems

TABLE II
APPLIED SENSOR SYSTEMS AND THEIR SPECIFICATIONS; $g \approx 9,81 \text{ m/s}^2$; LSB: LEAST SIGNIFICANT BIT

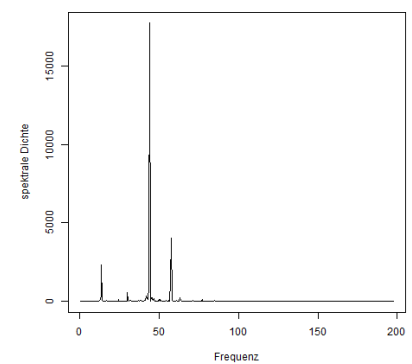
Specification	Professional System	Raspberry Pi	Mobile Phone
Range	+4 g, -1 g	$\pm 4 \text{ g}$	$\pm 4 \text{ g}$
Resolution	12 bits	10 bits	10 bits
Accuracy	1 %	128 LSB/g	128 LSB/g
Measuring Rate	continuous (analog)	3200 Hz	1600 Hz
Sampling Rate	1000 Hz	4040 Hz	$\approx 100 \text{ Hz}$



(a) Industrial system



(b) Raspberry Pi



(c) Mobile phone

Fig. 9: Periodograms of the sensor systems' data when triggered with 159.2 Hz

automated transport systems and support further research on scalable and cost-efficient monitoring solutions for industrial environments.

C. Robotized Picking and Placing in the Process Industry

a) *Industrial Context and Problem Definition:* In industrial process automation, AI solutions increasingly support

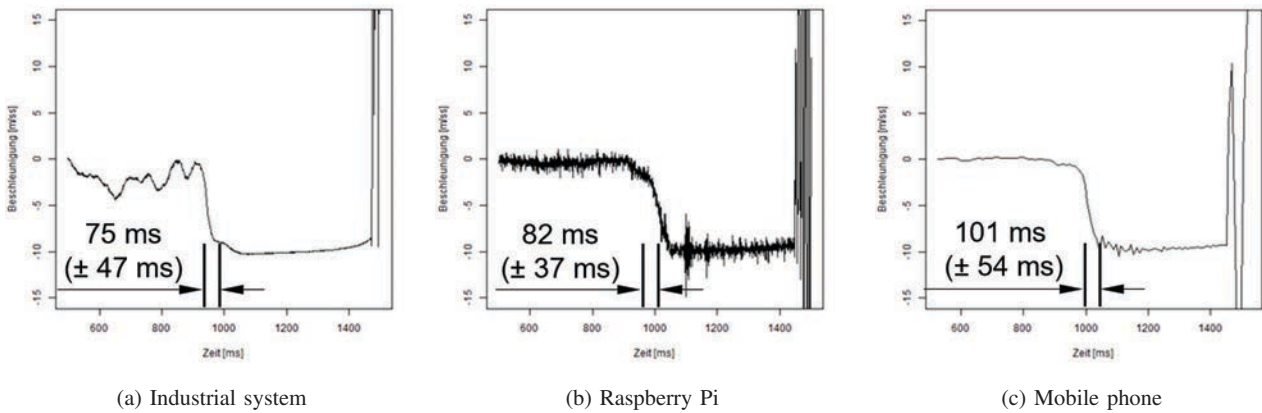


Fig. 10: Time series of the sensor systems’ data recording a shock event;

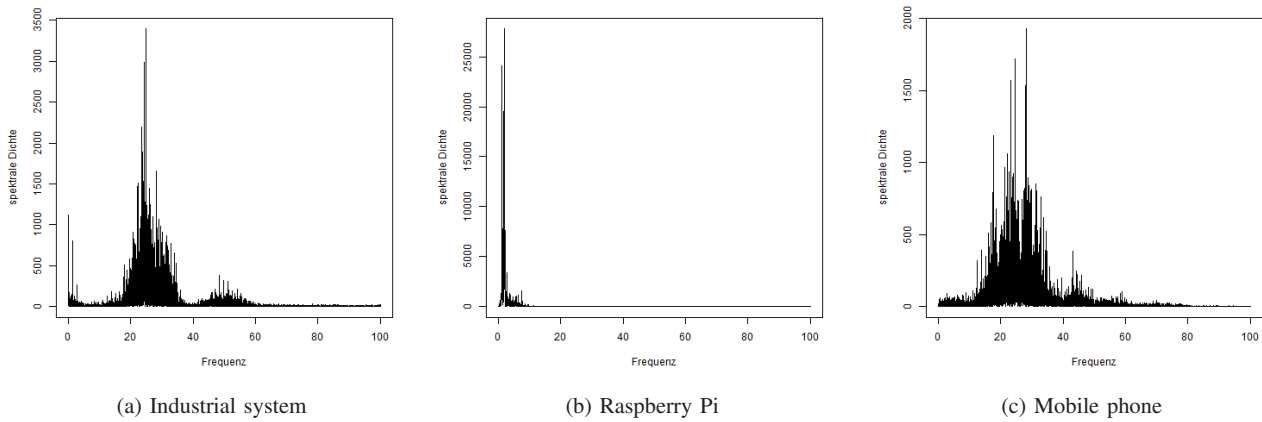


Fig. 11: Periodograms of the sensor systems’ data on a random AGV movement

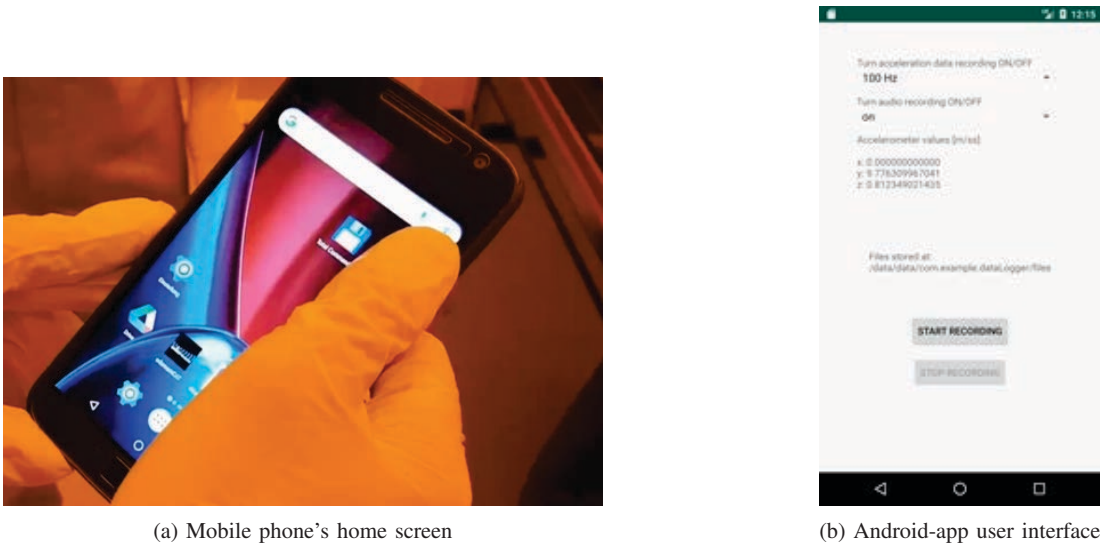
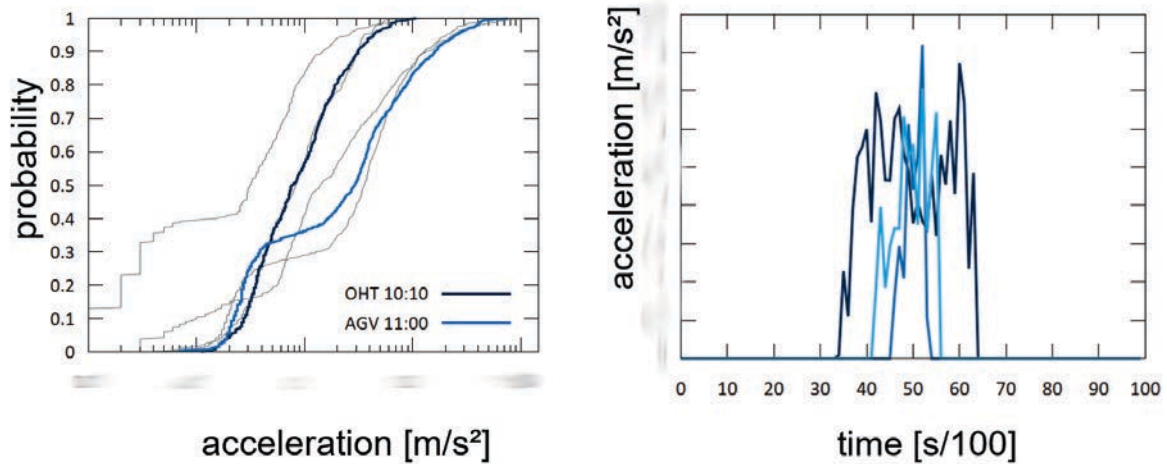


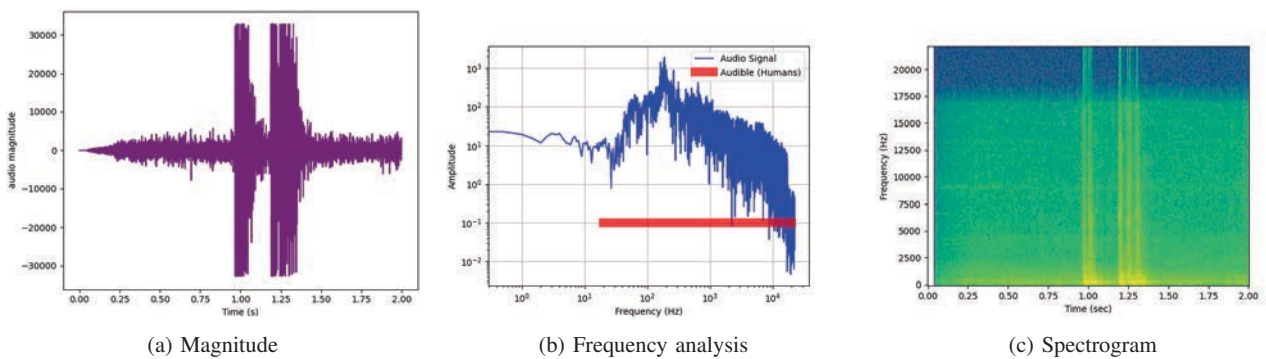
Fig. 12: Android mobile phone with app for sensor data recording



(a) Cumulative distribution function of recorded acceleration values

(b) Time series of recorded acceleration values

Fig. 13: Acceleration values recorded by the sensor platform applied in a FOUN applied in a FOUN carried by an OHT-vehicle (darker blue) and an AGV (brighter blue); precise values blurred for reasons of confidentiality

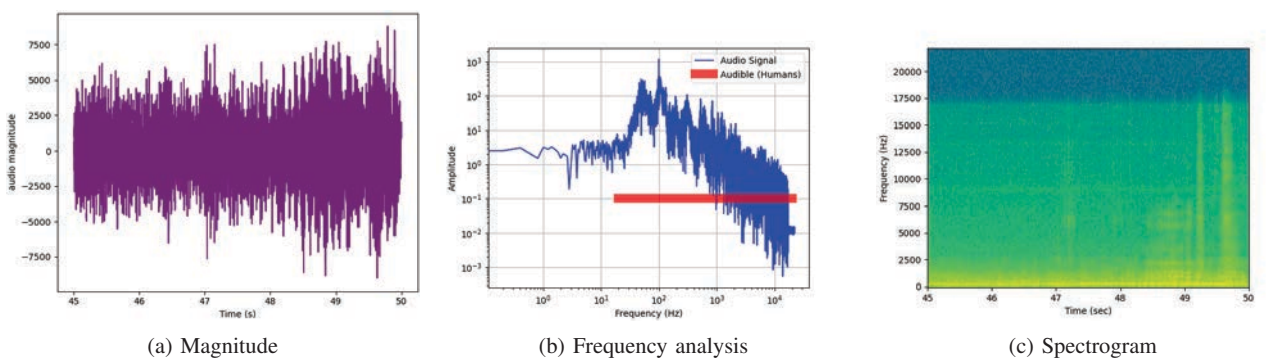


(a) Magnitude

(b) Frequency analysis

(c) Spectrogram

Fig. 14: Sensor data values while opening a FOUN's door



(a) Magnitude

(b) Frequency analysis

(c) Spectrogram

Fig. 15: Sensor data values while hoisting a FOUN by an OHT-vehicle

flexible and adaptable operations, often in confined and dynamic environments where robots must operate safely and efficiently alongside humans. A core task in this context is object picking and placement [62], which underpins assembly lines as well as packaging and material handling use cases [63]. Modern robotic solutions combine manipulators with

machine vision and advanced control to perform these tasks with high precision and speed [64].

Industrial deployments face recurring challenges: (i) robust operation under uncertainty, including varying object shapes, sizes, and poses [65], (ii) frequent task and product changes that require rapid reconfiguration with minimal human inter-

vention, and (iii) safe operation near humans through effective human intention detection and responsive motion planning [66]. Reinforcement Learning (RL) and related machine learning methods offer mechanisms for adaptive behavior, enabling robots to learn strategies in simulation and transfer them to real-world environments, improving safety and efficiency under changing conditions.

b) Approach / Methodology: The proposed approach combines modular robotic manipulation, machine vision, and learning-based control to automate conveyor feeding in a cosmetics production line. The system integrates (i) perception modules for object recognition, bottle pose estimation, grasp affordance estimation, and socket tracking, (ii) classical motion planning for robust pick trajectories, and (iii) a machine-learned controller for dynamically adapting the “place” motion to a potentially moving target socket.

In addition, the system targets rapid task reconfiguration through a generate-and-refine workflow that uses LLM-based code generation [67], [68], [69], [70] combined with simulation-based feedback in Isaac Sim [71]. This supports fast adaptation to changing product runs and operating constraints without requiring extensive manual reprogramming.

c) Experimental Setup / Implementation: A practical deployment scenario is drawn from a cosmetics production facility, where robots already perform filling, capping, brush insertion, and packaging tasks. The targeted automation step is currently performed manually: picking unfilled bottles from piles or delivery boxes and inserting them into sockets in carrier blocks moving along a conveyor belt. This task is constrained by limited space (no room for extensive pre-sorting hardware) and by the need for precise synchronization between real-time bottle pose estimation and socket motion to achieve reliable insertion at production throughput. Fig. 16 illustrates the station layout where a robot could be deployed.

The conveyor feeding system is implemented using a UR5e robot arm from Universal Robotics. A separate computer hosts machine vision and robot motion planning components, communicating with the robot via ROS2 APIs [72]. Motion planning and pick trajectory execution use services from MoveIt2 [73]. Grasp affordance estimation computes a 6DoF gripper pose and 3D approach direction using a multi-stage pipeline that combines off-the-shelf neural models with point cloud segmentation and shape fitting, selecting an accessible bottle for picking.

After grasping, a wrist-mounted camera provides refined relative pose estimation of the bottle to set the start state for motion planning toward the insertion area. For the placement phase, joint action planning is performed by a custom machine-learned model trained to adapt the robot motion based on real-time robot state and target socket state. Socket tracking uses a fine-tuned machine vision model based on YOLOv9 [74], [75]. Inputs to the placement controller consist of joint angles and velocities (from ROS2) and the current 2D position and velocity of the target socket; outputs are joint actuation commands (desired joint velocities and angles for the next time step) sent to the robot controller via a customized ROS2 action controller.

d) Results and Insights: The integrated system architecture demonstrates how combining classical robotics with learning-based components enables robust automation under realistic industrial constraints. MoveIt2-based motion planning provides reliable pick behavior, while the learned placement controller supports dynamic adaptation to moving sockets using real-time state inputs. The use of fine-tuned perception models (YOLOv9-based socket tracking) improves robustness in the presence of variable object and target configurations. At the workflow level, LLM-assisted task reconfiguration paired with simulation feedback supports faster iteration and reduces engineering overhead for frequent product changes.

This case study further illustrates a practical spectrum of AI integration: from off-the-shelf components (MoveIt2 motion planning, externally hosted LLM assistants), to fine-tuned pretrained models (YOLO), to custom neural models trained specifically for dynamic placement control. Together, these choices address key constraints of small-to-medium manufacturers, including limited space, limited engineering resources, and frequent task switching.

e) Implications and Transferability: This use case highlights a transferable integration pattern for industrial pick-and-place automation: combine modular perception, reliable classical planning for nominal motion generation, and learning-based controllers for high-variability sub-tasks where adaptation is essential. The generate-and-refine workflow using LLM code generation and simulation feedback offers an extensible strategy for rapid task reconfiguration beyond cosmetics manufacturing, particularly in settings with frequent product changes and limited commissioning time.

More broadly, the approach demonstrates how layered AI methods can improve automation performance while managing deployment risk: by constraining learned components to the most uncertainty-sensitive parts of the task (e.g., dynamic placement) and retaining proven planning and control frameworks for the remainder of the system.

D. Machine Learning Operations for Semi-Supervised Data Annotation

a) Problem Description: While machine learning models offer significant potential for industrial automation, their practical deployment and management extend far beyond initial training. In dynamic manufacturing environments, models face challenges such as data drift, where real-world data changes over time, causing performance to degrade. Manually managing the entire ML lifecycle, from data collection and annotation to retraining and redeployment, is inefficient, prone to error, and not scalable. One of the challenges addressed within the project is the operationalization of ML models in a structured manner, with one application being a semi-supervised data annotation pipeline. Without a dedicated MLOps framework, it would be difficult to consequently analyze the model’s incremental improvements in the semi-supervised loop, systematically compare parallel runs to select the most effective configuration, and continuously monitor model-related metrics to make informed decisions within the pipeline.



Fig. 16: Overhead view of a production line station where bottles are placed onto a conveyor

b) Approach: To address the challenges of manual annotation and model degradation, we propose an iterative, MLOps-driven pipeline for semi-supervised data annotation. The process begins by presenting a small, representative sample of manually annotated data to the system. Experimental results demonstrate the high data efficiency of this method, showing it functions effectively with an initial set of as few as 10 samples. Utilizing this seed dataset, an initial training phase involving multiple set of hyperparameters is conducted, after which the best-performing configuration is selected to serve as the baseline model. Potential extension of the method could involve training multiple architectures at once and selecting the baseline model among them.

Once established, this baseline model is deployed to generate pseudo-labels for the larger pool of unlabeled data. This initiates the core semi-supervised learning loop, where the newly generated pseudo-labels are evaluated, which relies on a composite confidence score, which is assigned based on both the model's inherent prediction confidence and the embedding similarity of the detected objects.

The operationalization of this iterative loop relies heavily on the MLOps practices. Pseudo-labels that exceed a high-confidence threshold are automatically converted into true ground-truth annotations and appended to the training set. A dedicated data versioning system manages this integration, capturing a snapshots of the dataset as it evolves to guarantee full reproducibility across iterations. This newly versioned dataset subsequently triggers an automated training pipeline, which retrains the baseline model without the need to manually annotate or review the labels.

The automatically retrained model is then deployed to infer a new set of pseudo-labels for the remaining unlabeled data, it-

eratively refining the dataset's quality. This cyclical process of generation, confidence evaluation, automated data versioning, and model retraining continues until the termination criterion is met. For the conducted experiments, when the change in the number of high-confidence labels produced between successive iterations falls below a specified threshold, the process stops, indicating pipeline convergence.

IV. CONCLUSIONS AND IDENTIFICATION OF FUTURE STREAMS FOR RESEARCH AND DEVELOPMENT

The described applications in practice—from material handling to intralogistics and production management (Table III)—show the interconnectedness of engineering-oriented solutions (e.g., robotic handling, health monitoring of automated transport systems) and management-oriented methods (e.g., fleet management, variability reduction, bottleneck identification). These examples, spanning industries such as cosmetics and semiconductors, highlight how the joint application of engineering and management perspectives contributes to the operational efficiency and resilience of modern production systems.

The applied approaches rely on and make use of the increasing availability of industrial data. Both heuristic methods, such as rapid modeling, and data-intensive techniques, such as computer vision, supervised learning, and reinforcement learning, are complemented by human reasoning and decision-making support (e.g., SHAP-based explainability). Together, these approaches demonstrate that hybrid solutions—combining data-driven methods with domain knowledge and interpretability—yield promising results in practice.

As summarized in Table III, several insights can be derived. First, machine learning does not replace established production

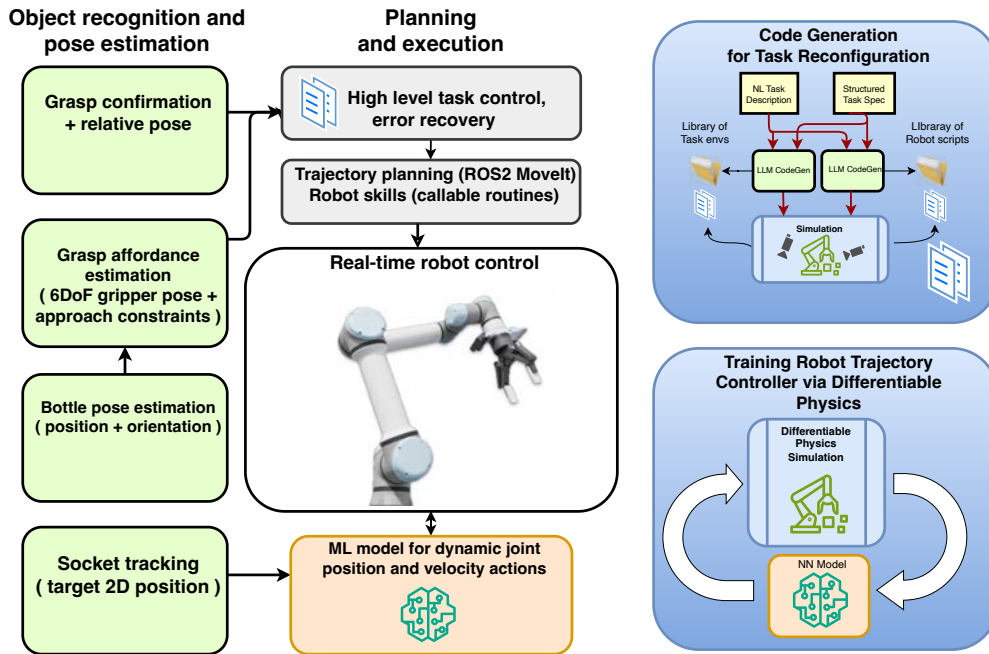


Fig. 17: System Overview. Green boxes (on the left) are components that supply real-time data to the robot control system. Blue boxes on the right represent training-time (prior to deployment) systems that produce scripts and machine-learned models for deployment into the runtime system (central column).

TABLE III
OVERVIEW OF PRESENTED APPLICATION AREAS, OPERATIONS, CHALLENGES, AND APPROACHES IN MANUFACTURING.

Application	Operation	Industry	Challenge	Approach
Material handling Picking and placing	Assembly and commissioning of components on conveyor belt	Cosmetics	Accuracy, safety in human-robot collaboration, and adaptation to frequent product reconfigurations	Computer vision and reinforcement learning for adaptive control systems
Intralogistics Fleet management	Transportation task assignment and route planning of AMRs	Semiconductors	Coordinating AMR fleets with >1,000 possible locations in real-time	Predicting travel times using supervised learning on log-files
Health monitoring	Automated transportation carrier	Semiconductors	Recognizing imminent malfunctions of the automated transportation system	Using low-cost sensor platform
Production Variability reduction	Process variability at work-station	Semiconductors	Identifying impact of process variability on flow time and utilization	Rapid modeling technology based on networks of queuing models
Bottleneck identification	WIP levels in front of work-station	Semiconductors	Short-term prediction of WIP levels and identifying influencing process variables to mitigate bottlenecks	Prediction of WIP levels using XGBoost and visualization of local feature importances with SHAP values

and process management methodologies but amplifies their impact when properly integrated. Further, the chosen methods must be tailored to the specific challenges of the application domain: reinforcement learning for adaptive robotic control in cosmetics, supervised learning for intralogistics fleet optimization in semiconductors manufacturing, or rapid modeling for analyzing variability in production flows. Additionally, across domains, the combination of digital methods and managerial practices shows the potential to create more adaptive, efficient, and sustainable production systems capable of coping with rising complexity and variability.

Overall, the experiences gathered in the project AIMS5.0 illustrate how interdisciplinary collaboration can turn advanced methods into operational improvements, offering a pathway toward more intelligent, connected, and robust manufacturing systems.

REFERENCES

- [1] M. Baudin and T. H. Netland, *Introduction to manufacturing: An industrial engineering and management perspective*, 1st ed. New York, NY: Routledge, 2023.
- [2] W. J. Hopp and M. L. Spearman, *Factory Physics*, 3rd ed. Long Grove, IL, USA: Waveland Press, 2011.

- [3] A.-W. Scheer, *Wirtschaftsinformatik: Referenzmodelle für industrielle Geschäftsprozesse*, 6th ed. Berlin, Heidelberg: Springer, 1995.
- [4] M. E. Porter and M. E. Porter, *Competitive advantage : creating and sustaining superior performance*. New York [u.a.] : Free Pr. [u.a.], 1985.
- [5] J. P. Womack and D. T. Jones, *Lean Thinking: Banish Waste and Create Wealth in Your Corporation, Revised and Updated*. New York: Free Press, Jun. 2003.
- [6] R. Suri, *Quick response manufacturing : a companywide approach to reducing lead times*. Portland, Or.: Productivity Press, 1998.
- [7] G. Reiner, *Rapid Modelling for Increasing Competitiveness – Tools and Mindset*. Germany: Springer, 2009.
- [8] —, *Rapid Modelling and Quick Response - Intersection of Theory and Practice*, 1st ed. Germany: Springer, 2010.
- [9] A. Rolstadås, "Production management systems," in *Computer-Aided Production Management*, A. Rolstadås, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1988, pp. 3–19.
- [10] L. Chen and Y. Huang, "Integrated product and process control for sustainable semiconductor manufacturing," *Chinese Journal of Chemical Engineering*, vol. 19, no. 2, pp. 192–198, 2011.
- [11] M. Xu, "The application of quality management in chip production," *Journal of Innovation and Development*, vol. 6, no. 3, pp. 4–7, 2024.
- [12] C.-C. Chen, "Integration of quality function deployment and process management in the semiconductor industry," *International Journal of Production Research*, vol. 47, no. 6, pp. 1469–1484, 2009.
- [13] R. D. Klassen and L. J. Menor, "The process management triangle: An empirical investigation of process trade-offs," *Journal of Operations Management*, vol. 25, no. 5, pp. 1015–1034, 2007.
- [14] W. S. Lovejoy, "Integrated operations: A proposal for operations management teaching and research," *Production and Operations Management*, vol. 7, no. 2, pp. 106–124, 1998.
- [15] G. M. Schmidt, "The om triangle," *Operations Management Education Review*, vol. 1, no. 1, pp. 87–104, 2005.
- [16] T. Yang, X. Yi, S. Lu, K. H. Johansson, and T. Chai, "Intelligent manufacturing for the process industry driven by industrial artificial intelligence," *Engineering*, vol. 7, no. 9, pp. 1224–1230, 2021.
- [17] S. A. S. Moreira and S. Dallavalle, "Unraveling the trends in business process management: a comprehensive bibliometric analysis of management and business literature," *Business Process Management Journal*, vol. 30, no. 7, pp. 2541–2563, 2024.
- [18] J. Butt, "A conceptual framework to support digital transformation in manufacturing using an integrated business process management approach," *Designs*, vol. 4, no. 3, p. 17, 2020.
- [19] Y. Dmytriiev, M. Carnevale, and H. Giberti, "Enhancing flexibility and safety: collaborative robotics for material handling in end-of-line industrial operations," *Procedia Computer Science*, vol. 232, pp. 2588–2597, 2024.
- [20] *ISO 8373:2021 – Robotics – Vocabulary*, International Organization for Standardization, 2021, accessed: 30 Mar. 2025. [Online]. Available: <https://www.iso.org/standard/72700.html>
- [21] S. S. Heragu and B. Ekren, *Materials Handling System Design*. John Wiley & Sons, Ltd, 2009, ch. 1, pp. 1–29. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470432730.ch1>
- [22] K. Kiran, M. Komagan, M. Jayashree, S. Meda, D. Gupta, and K. K. Thoti, "Impact of material handling equipment on warehouse performance-a case of indian warehouses located in metro cities," *Metallurgical and Materials Engineering*, vol. 31, p. 272, 05 2025.
- [23] P. Alumbugu, W. Shakantu, A. Tsado, and W. Ola-awo, "Efficient utilisation of automation in construction materials-handling processes," *Acta Structilia*, vol. 29, no. 2, p. 33–57, Nov. 2022. [Online]. Available: <https://journals.ufs.ac.za/index.php/as/article/view/6907>
- [24] M. I. Robots. Amr vs agv - key differences explained. [Online]. Available: <https://mobile-industrial-robots.com/blog/agv-vs-amr-whats-the-difference>
- [25] I. Ali Ibrahim and M. Baballe, "The different types of unmanned aerial vehicle (uav): Characteristics, capabilities, and challenges," vol. 4, pp. 89–94, 08 2024.
- [26] Z. Imani, P. Forsythe, A. A. Fard Fini, M. Maghrebi, and T. S. Waller, "Autopilot drone in construction: A proof of concept for handling lightweight instruments and materials," *Results in Engineering*, vol. 23, p. 102 498, 2024. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2590123024007539>
- [27] A. K. Pinagapani, A. G. A. S. N. K. R. V. and P. G. B. "A hybrid analysis method for multi-class queueing networks with multi-server nodes," *The International Measurement Confederation*, vol. 13, no. 4, pp. 1–8, 2024. [Online]. Available: <https://acta.imeko.org/index.php/acta-imeko/article/view/1728>
- [28] S. Al-Hussaini, S. Thakar, H. Kim, P. Rajendran, B. Shah, J. Marvel, and S. Gupta, "Human-supervised semi-autonomous mobile manipulators for safely and efficiently executing machine tending tasks," 10 2020.
- [29] J. F. Elfferich, D. Dodou, and C. D. Santana, "Soft robotic grippers for crop handling or harvesting: A review," *IEEE access*, vol. 10, pp. 75 428–75 443, 2022.
- [30] J. A. Ceroni, "Economic Rationalization of Automation Projects and Quality of Service," in *Springer Handbook of Automation*, S. Y. Nof, Ed. Cham: Springer International Publishing, 2023, pp. 683–698.
- [31] S. J3016, *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*. Warrendale, Troy: sae.org, 2021.
- [32] T. Gamer, M. Hoernicke, B. Kloepper, R. Bauer, and A. J. Isaksson, "The Autonomous Industrial Plant -Future of Process Engineering, Operations and Maintenance," *IFAC-PapersOnLine*, vol. 52, no. 1, pp. 454–460, Jan. 2019.
- [33] J. Fottner, D. Clauer, F. Hormes, M. Freitag, T. Beinke, L. Overmeyer, S. N. Gottwald, R. Elbert, T. Sarnow, T. Schmidt, K.-B. Reith, H. Zadek, and F. Thomas, "Autonomous systems in intralogistics: State of the art and future research challenges," *Logistics Research*, vol. 14, no. 1, pp. 1–41, 2021.
- [34] Y. Nishi and R. Doering, Eds., *Handbook of Semiconductor Manufacturing Technology*, 2nd ed. Boca Raton: CRC Press, Jul. 2007.
- [35] G. Agrawal and S. Heragu, "A survey of automated material handling systems in 300-mm SemiconductorFabs," *IEEE Transactions on Semiconductor Manufacturing*, vol. 19, no. 1, pp. 112–120, Feb. 2006.
- [36] J. Tung, T. Sheen, M. Kao, and C. H. Chen, "Optimization of AMHS Design for a Semiconductor Foundry Fab by Using Simulation Modeling," in *Proceedings of the 2013 Winter Simulation Conference*, ser. WSC '13, R. Pasupathy, S.-H. Kim, A. Tolc, R. Hill, and M. E. Kuhl, Eds. Piscataway: Institute of Electrical and Electronics Engineers, Inc., 2013, pp. 3829–3839.
- [37] C. Fischmann, F. Böttinger, and R. Wertz, "Buffer management for automated material handling systems in semiconductor industries," in *Proceedings of the 22nd European Conference on Modelling and Simulation, ECMS*, L. S. Louca, Y. Chrysanthou, Z. Oplatková, and K. Al-Begain, Eds., Nicosia, 2008, pp. 423–427.
- [38] SEMI, "Introduction to SEMI's Communication Standards: SECS/ GEM I SEMI," <https://www.semi.org/en/standards-watch-2022-Sept/intro-to-semi-communication-standards>, 2022.
- [39] T.-E. Lee, H.-J. Kim, and T.-S. Yu, "Semiconductor Manufacturing Automation," in *Springer Handbook of Automation*, S. Y. Nof, Ed. Cham: Springer International Publishing, 2023, pp. 841–863.
- [40] B. Ferreira and J. Reis, "A Systematic Literature Review on the Application of Automation in Logistics," *Logistics*, vol. 7, no. 4, p. 80, Dec. 2023.
- [41] D. Kreuzberger, N. Kühl, and S. Hirschl, "Machine learning operations (mlops): Overview, definition, and architecture," *IEEE access*, vol. 11, pp. 31 866–31 879, 2023.
- [42] L. Faubel, K. Schmid, and H. Eichelberger, "Mlops challenges in industry 4.0," *SN Comput. Sci.*, vol. 4, no. 6, Oct. 2023. [Online]. Available: <https://doi.org/10.1007/s42979-023-02282-2>
- [43] O. E. Oluyisola, S. Bhalla, F. Sgarbossa, and J. O. Strandhagen, "Designing and developing smart production planning and control systems in the industry 4.0 era: a methodology and case study," *Journal of Intelligent Manufacturing*, vol. 33, no. 1, pp. 311–332, 2022.
- [44] G. Dimitrakopoulos, T. Gutt, H. Ehm, A. Hoess, K. Karathanasopoulou, A. Lackner, and G. Schneider, "On the way to realize the 5th industrial revolution: Achievements, challenges and research areas," in *NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2023, pp. 1–6.
- [45] D. Eberts, R. Rottnick, G. Schneider, S. Keil, R. Lasch, and O. Buhmann, "Managing variability within waferet production by combining lean and six sigma," in *2012 SEMI Advanced Semiconductor Manufacturing Conference*. IEEE, 2012, pp. 33–38.
- [46] F. Lindner, K. A. Przybysz, G. Schneider, and S. Keil, "Workers' attitudes towards digital transformation and perceived well-being at work: A case study," *ACC JOURNAL*, vol. 30, no. 2, pp. 47–62, 2024.

- [47] F. Zoghalmi, O. K. Sen, H. Heinrich, G. Schneider, E. Ercelik, A. Knoll, and T. Villmann, "Tof/radar early feature-based fusion system for human detection and tracking," in *2021 22nd IEEE International Conference on Industrial Technology (ICIT)*. IEEE, 2021, pp. 942–949.
- [48] B. Rabta, R. Schodl, G. Reiner, and J. Fichtinger, "A hybrid analysis method for multi-class queueing networks with multi-server nodes," *Decision Support Systems*, vol. 54, no. 4, pp. 1541–1547, 2013, rapid Modeling for Sustainability. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0167923612001777>
- [49] W. Whitt, "The queueing network analyzer," *Bell System Technical Journal*, vol. 62, no. 9, pp. 2779–2815, 1983.
- [50] M. Segal and W. Whitt, "A queueing network analyzer for manufacturing," in *Teletraffic Science for New Cost-Effective Systems, Networks and Services*, M. Bonatti, Ed. North-Holland, 1989, pp. 1146–1152.
- [51] J. Brownlee, *Data Preparation for Machine Learning: Data Cleaning, Feature Selection, and Data Transforms in Python*. n.p.: Machine Learning Mastery, 2020. [Online]. Available: <https://books.google.de/books?id=uAPuDwAAQBAJ>
- [52] *Artificial intelligence — Data quality for analytics and machine learning (ML) — Part 4: Data quality process framework*, ISO/IEC JTC 1/SC 42 Std. ISO/IEC 5259-4, 2024, first edition. Published July 2024. ©ISO/IEC 2024.
- [53] J. Kim, H. Kim, H. Kim, D. Lee, and S. Yoon, "A comprehensive survey of deep learning for time series forecasting: Architectural diversity and open challenges," *Artificial Intelligence Review*, vol. 58, p. 216, 2025.
- [54] X. Kong, Z. Chen, W. Liu, K. Ning, L. Zhang, S. M. Marier, Y. Liu, Y. Chen, and F. Xia, "Deep learning for time series forecasting: A survey," *International Journal of Machine Learning and Cybernetics*, vol. 16, pp. 5079–5112, 2025.
- [55] T. Hall and K. A. Rasheed, "A survey of machine learning methods for time series prediction," *Applied Sciences*, vol. 15, p. 5957, 2025.
- [56] T. Jiang, Z. Sun, S. Fu, and Y. Lv, "Human-ai interaction research agenda: A user-centered perspective," *Data and Information Management*, vol. 8, 2024.
- [57] S. Mehrotra, C. Degachi, O. Vereschak, C. M. Jonker, and M. L. Tielman, "A systematic review on fostering appropriate trust in humanai interaction: Trends, opportunities and challenges," *ACM Journal on Responsible Computing*, vol. 1, no. 4, 2024.
- [58] T. Wagner, J. Seitz, and G. Schneider, "Vibration Measurement and Visualization in Semiconductor AMHS on the basis of IoT," in *Proceedings of the 22nd IEEE International Conference on Industrial Technology (ICIT)*, vol. 1. Valencia, Spain: Institute of Electrical and Electronics Engineers, Inc., Mar. 2021, pp. 1211–1216. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/9453604>
- [59] P. Strauß, M. Schmitz, R. Wöstmann, and J. Deuse, "Enabling of Predictive Maintenance in the Brownfield through Low-Cost Sensors, an IIoT-Architecture and Machine Learning," in *Proceedings of the 2018 IEEE International Conference on Big Data (Big Data)*. Seattle, WA, USA: IEEE, Dec. 2018, pp. 1474–1483. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/8622076>
- [60] J. Engelbrecht, M. J. Booyens, G.-J. van Rooyen, and F. J. Bruwer, "Survey of smartphone-based sensing in vehicles for intelligent transportation system applications," *IET Intelligent Transport Systems*, vol. 9, no. 10, pp. 924–935, 2015.
- [61] A. Ait Ben Ahmed, A. Touache, A. El Hakimi, and A. Chamat, "Experimental Evaluation of MEMS Accelerometers Integrated into Smartphones: A Case Study of Bearing Condition Monitoring," in *Digital Technologies and Applications*, ser. Lecture Notes in Networks and Systems, S. Motahhir and B. Bossoufi, Eds. Cham: Springer International Publishing, 2021, pp. 1777–1786.
- [62] G. Casella, A. Volpi, R. Montanari, L. Tebaldi, and E. Bottani, "Trends in order picking: A 2007–2022 review of the literature," *Production & Manufacturing Research*, vol. 11, no. 1, p. 2 191 115, 2023.
- [63] O. Vismanis, J. Arents, K. Freivalds, V. Ahluwalia, and K. Ozols, "Robotic system for post office package handling," *Applied Sciences*, vol. 13, no. 13, 2023. [Online]. Available: <https://www.mdpi.com/2076-3417/13/13/7643>
- [64] S. Surati, S. Hedao, T. Rotti, V. Ahuja, and N. Patel, "Pick and place robotic arm: a review paper," *Int. Res. J. Eng. Technol*, vol. 8, no. 2, pp. 2121–2129, 2021.
- [65] J. Zhu, A. Cherubini, C. Dune, D. Navarro-Alarcon, F. Alambeigi, D. Berenson, F. Ficuciello, K. Harada, J. Kober, X. Li et al., "Challenges and outlook in robotic manipulation of deformable objects," *IEEE Robotics & Automation Magazine*, vol. 29, no. 3, pp. 67–77, 2022.
- [66] L. Gualtieri, E. Rauch, and R. Vidoni, "Development and validation of guidelines for safety in human-robot collaborative assembly systems," *Computers & Industrial Engineering*, vol. 163, p. 107801, 2022.
- [67] Gemini Team, Google, "Gemini 3: State-of-the-art reasoning and agentic workflows," Google DeepMind Technical Report, 2025. [Online]. Available: <https://ai.google.dev/gemini-api/docs/gemini-3>
- [68] Google Cloud, "Gemini command line interface (cli)," 2025, part of the Google Cloud SDK. [Online]. Available: <https://cloud.google.com/sdk/gcloud/reference/ai-platform>
- [69] OpenAI, "Gpt-5.2 technical report: Enhancing professional knowledge work and reasoning," 2025, accessed: 2025-12-18. [Online]. Available: <https://openai.com/index/introducing-gpt-5-2/>
- [70] —, "Codex cli: Lightweight coding agent for the terminal," 2025. [Online]. Available: <https://developers.openai.com/codex/cli/>
- [71] NVIDIA Corporation, "Nvidia isaac sim 5.0: Robotics simulation and synthetic data generation platform," 2025, version 5.0.0, Released August 2025. [Online]. Available: <https://github.com/isaac-sim/IsaacSim>
- [72] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, "Robot operating system 2: Design, architecture, and uses in the wild," *Science Robotics*, vol. 7, no. 66, p. eabm6074, 2022. [Online]. Available: <https://www.science.org/doi/abs/10.1126/scirobotics.abm6074>
- [73] D. Coleman, I. A. S. ucan, S. Chitta, and N. Correll, "Reducing the barrier to entry of complex robotic software: a moveit! case study," *Journal of Software Engineering for Robotics*, vol. 5, no. 1, pp. 3–16, 2014.
- [74] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," 2016. [Online]. Available: <https://arxiv.org/abs/1506.02640>
- [75] C.-Y. Wang, I.-H. Yeh, and M. L. Hong-Yuan, "Yolov9: Learning what you want to learn using programmable gradient information," 2024, <https://arxiv.org/abs/2402.13616>.

Large Language Models in Industry 5.0

– Multi-Case Experiences from the AIMS5.0 Project

Csaba Hegedűs¹, Péter Olaszi¹, Máté Bancsics¹, Dániel Ficzer^{1,2}, Alex Latipov³, Mike Holenderski³, Guntis V. Strazds⁴, Belal Abu Naim⁵, Yasin Ghafourian⁵, Markus Tauber⁵, Marta Bonik⁶, Chris Damkat⁷, David Dandolo⁸, Chiara Masiero⁸, Manuel Barusco⁹, Gian Antonio Susto⁹, Ruba Almahasneh¹, and Pál Varga¹

Abstract—The transition from Industry 4.0 toward the humancentered vision of Industry 5.0 creates new opportunities for AI systems that augment rather than replace human expertise. This paper presents the integration of Large Language Models (LLMs) and AI copilots into industrial engineering workflows through multiple cases within the EU AIMS5.0 project. Eight use cases span system-of-systems engineering, support and aftersales services, regulatory self-assessment, product engineering, factual correctness checking, robot programming, ontology and knowledge-graph applications, and AI-assisted testing. The cases range from conceptual designs and proofs of concept to pilot implementations and therefore provide implementation experience and design insights rather than directly comparable performance results. Across the cases, recurring architectural patterns include Retrieval-Augmented Generation (RAG), semantic search, multiagent planning, natural-language interfaces, code generation, and verification mechanisms. The experiences also reveal common deployment challenges related to heterogeneous and evolving data sources, multimodal content, access control, factual reliability, legacy-system integration, and evaluation. Viewed through the Industry 5.0 dimensions of human-centricity, sustainability, and resilience, the cases illustrate how LLM-based systems can support knowledge access, engineering assistance, system adaptation, and safer interaction with industrial AI. The findings emphasize that successful industrial adoption depends not only on model capabilities, but also on high-quality domain content, appropriate safeguards, transparent system behavior, integration with existing infrastructure, and continuous human oversight.

Index Terms—Large Language Models, Industry 5.0, Human-Centered AI, AI Copilots, Retrieval-Augmented Generation, Industrial AI, Intelligent Manufacturing, Human–AI Collaboration

¹ Department of Telecommunications and Artificial Intelligence, Budapest University of Technology and Economics, Műgyetem rkp. 3., H-1111 Budapest, Hungary.

² AI & IIoT Division, AITIA International Inc., 7 Záhony Street, 1031 Budapest, Hungary.

³ Department of Mathematics and Computer Science, Eindhoven University of Technology, The Netherlands.

⁴ Institute of Electronics and Computer Science, Dzērbenes iela 14, LV-1006, Rīga, Latvia.

⁵ Research Studios Austria Forschungsgesellschaft mbH, Smart Digital Industries and Services, Heinrich Schmidladl-Straße 15, A-3100 St. Pölten, Austria.

⁶ Infineon Technologies AG, Am Campeon 1–15, 85579 Neubiberg, Germany.

⁷ Signify Netherlands B.V., High Tech Campus, 5656AE Eindhoven, The Netherlands.

⁸ Statwolf Data Science S.r.l., Padova, Italy.

⁹ Department of Information Engineering, University of Padova, Italy.

I. INTRODUCTION AND MOTIVATION FOR LLMs IN INDUSTRIAL PRODUCTION

Industry 5.0 extends the digitalization and automation of Industry 4.0 with a stronger focus on human-centricity, sustainability, and resilience. In this setting, Artificial Intelligence (AI) is not only a means of automating industrial processes. It can also support engineers, operators, and service personnel in tasks that require access to complex information and interaction with technical systems. Large Language Models (LLMs), chatbots, and AI copilots are especially relevant because they provide a natural-language interface to knowledge, software, and engineering tools.

This role is important because industrial information is often fragmented. Engineers may need to consult technical manuals, specifications, maintenance records, project documentation, and internal databases before they can resolve a problem. The relevant material may also be distributed across systems, stored in different formats, or maintained in different versions. Finding and combining this information can therefore be slow and error-prone. LLM-based assistants can help by retrieving and summarizing relevant material, particularly when they are combined with Retrieval-Augmented Generation (RAG) and domain-specific knowledge sources. Their usefulness, however, depends on more than the language model itself. Data quality, access rights, factual reliability, and integration with existing infrastructure are equally important in industrial settings.

Recently many research works have explored LLMs for individual engineering and manufacturing tasks, including knowledge retrieval, code generation, robotic control, and production-system interaction. These studies show the breadth of possible applications, but industrial deployments differ greatly in maturity, autonomy, and available evidence. A broader view is therefore useful for understanding which implementation patterns recur across use cases and which challenges appear when LLMs are integrated into real engineering environments.

This paper reports experience from the EU AIMS5.0 project through a portfolio of industrial LLM applications. The cases span several stages of the industrial lifecycle and include engineering assistance, support services, compliance guidance, software and robot programming, knowledge management,

and AI evaluation. Their maturity ranges from conceptual architectures and proofs of concept to pilot implementations. For this reason, the aim of the paper is not to compare the cases statistically. Instead, we examine how different forms of LLM integration can support industrial workflows and how they relate to the goals of Industry 5.0.

The paper makes three main contributions. It first brings together implementation experience from several industrial LLM applications within one project. It then analyzes the cases through a common framework that considers the role of the LLM, the human–system relationship, maturity, available evidence, and relevance to Industry 5.0. Finally, it identifies recurring implementation patterns and practical challenges across the cases.

The remainder of the paper is organized as follows. Section II reviews related work on LLM technologies and industrial applications. Section III introduces the qualitative multi-case methodology and the analytical framework. Section IV presents the individual use cases and their implementation experience. Section V concludes the paper with the main observations, limitations, and directions for future work.

II. RELATED WORKS

A. Large language models and Agents

The introduction of the transformer architecture [1] started the epoch of Large Language Models (LLMs). LLMs can be categorized by their architecture as encoder-, decoder, or encoder-decoder-based. From an application standpoint, they can be completion, chat, or reasoning models. We can differentiate between the supported modalities as well.

Most state-of-the-art (SotA) LLMs utilize In-Context Learning (ICL) with prompt engineering, where models learn to perform new tasks by using examples provided in the input prompt without parameter updates [2]. Studies suggest [3], [4] that the emergent reasoning abilities of LLMs are enabled by ICL, though the exact mechanisms remain to be explained.

LLMs and their reasoning abilities can be utilized with creative prompt engineering. A single-task prompt is an instruction or query that requires a single, specific task for the Artificial Intelligence (AI) to complete. They are commonly used in various Natural Language Processing (NLP) applications, such as text classification, summarization, translation or Named Entity Recognition. These tasks are fundamental to many AI applications, including chatbots, search engines, automated content generation, and data analysis [5], [6].

Prompt engineering assumes that effective prompts are built on specific core elements. The effect of these elements is measurable even for smaller, fine-tuned models: adding instructional text to the input raised the accuracy of a Hungarian BERT model on a language-understanding benchmark from 65% to 85% [7]. These include instructions which outline the task the model needs to perform; context, which offers external information to guide the model's understanding; input data, representing the questions or problems requiring solutions; and output indicators, which define the expected format or type of response [8], [9].

More complicated tasks require a chain of thought and/or a chain of interactions with external services. In general LLMs

struggle with complex reasoning. To address these challenges the LLM can be guided to reason over multi-step solutions. Chain-of-Thought (CoT) [10] prompting, for example, helps LLMs break down complex tasks into smaller steps. The limitations of CoT can be addressed by various decomposition strategies, increasing context memory, and adaptive step-by-step processing, such as Tree of Thoughts [11]. CoT can even be extended with Self-Consistency [12]. ReAct [13] interleaves Chain-of-Thought reasoning steps with calls to external APIs, whose results are processed by subsequent inference steps for generating the final response. Since its introduction, the ReAct "think, act, think, ... respond" pattern has become fairly common for tool-using agents. PAL [14] integrates LLMs with a Python interpreter to generate reasoning steps with Python code. Note that communication is one-way, with no option for feedback.

LLM agents can learn from their history of interactions with users and the external world. They are often implemented as a set of collaborating LLM modules. LLM agent components include an action/output generator (the "proposer") and a critic and/or learner (Reflexion [15]). LLM agents often employ some form of external memory system.

B. Retrieval Augmented Generation

LLMs often generate incorrect information, rely on outdated knowledge, or reach false conclusions through incorrect reasoning. These challenges are particularly evident in domain-specific or knowledge-intensive tasks which require up-to-date information or information not present in the training data. LLMs often confabulate facts ("hallucinate") in such situations. Retrieval Augmented Generation (RAG) [16] has emerged as mitigation to these issues by amending the user's input prompt with reliable, up-to-date data from reliable knowledge sources.

The RAG pipeline utilizes a knowledge base that can be searched by the LLM agents. First, raw data is extracted from PDF, HTML, Word, Markdown or even multimodal documentation. Then the material is cleaned and broken down into smaller segments called *chunks*. These chunks are subsequently encoded into a vector representations using an embedding model, and stored in a vector database. The choice of the embedding model matters, especially for languages other than English and for company-specific terminology: on a Hungarian corporate document set, retrieval quality varied substantially across eight multilingual embedding models [17]. The user's query is also converted into the same embedding vector representation which was used for indexing the reference documentation.

The naive RAG approach can be refined to serve specific use cases. Recursive Abstractive Processing for Tree-Organized Retrieval (RAPTOR) [18] implements a hierarchical, tree-based retrieval system. This improves Contextual Depth and enables efficient Multi-Step Reasoning by employing recursive summaries. Some other methods implement knowledge graphs [19] or combine RAG with agentic approaches that execute multiple retrievals [20].

C. Program Code Generation with LLMs

Besides processing natural language texts, LLMs can also deal with formal languages such as specifications, logics, and programming languages. LLMs can be used to generate, summarize, or modify programming languages and documentation. They can also translate from a natural language to source code (like Python or JavaScript), or modify existing code (e.g. add features, generate unit tests, or fix bugs) [21]–[26].

D. LLM Application Architectures

The typical LLM-based architecture is composed of a pretrained Large Language Model, external “tool” APIs, databases or vector stores for RAG, multiple specialized prompts or models, and external solvers or verifiers. Implementing the system requires a fair amount of coding for handling control flow, business logic, connection to cloud-based APIs, handle errors, etc. Various frameworks have been developed for building such systems. LlamaIndex (<https://llamaindex.ai>) and LangChain (<https://langchain.com>) are two of the most popular such frameworks. Originally developed for supporting RAG, both frameworks today help building complex agentic systems and work with third-party libraries and services.

LLM-based chatbots rely on Natural Language Understanding (NLU) and Natural Language Generation (NLG) support. LLMs in customer service provide personalized support, automated content generation, and sentiment analysis. Integrating LLMs into Customer Relationship Management (CRM) systems improves response relevance. The benefits of LLM-powered chatbots include scalability, consistent response quality, customization for brand alignment, and expenditure savings. Challenges still persist, such as addressing the risk of factually incorrect responses, non-compliance with data privacy laws. Handling out-of-domain queries requires hybrid models and fallback mechanisms.

The formal evaluation of LLMs improves their accuracy, safety, and reliability. **Factual correctness** is evaluated in question answering benchmarks and by measuring factuality in LLM-generated summaries. Core benchmarks are FEVER [27], FRANK [28], WikiFact [29], SQuAD [30]. Evaluation methods include FactScore [31], QAGS [32], BERTScore [33], and FactCC [34]. **Logical reasoning** is assessed through commonsense, mathematical (GSM8K [35]), logical (FOLIO [36], LogiQA [37]), and coding benchmarks. **Language Proficiency** evaluation assesses the model’s ability to generate fluent, coherent, and contextually appropriate language. This includes assessing grammar and style, context preservation, and adaptability across diverse prompts. Classic metrics like BLEU [38] and ROUGE [39] compare LLMs’ outputs with reference data. The Arena chatbot [40] is the primary platform for evaluating conversational abilities and interaction quality. **Alignment** ensures that content produced by LLMs is ethical, non-biased, and in line with human values. TrustGPT [41] among others measures bias, toxicity, and truthfulness of the model’s outputs. **LLMs as agents**. AgentBench [42] is a benchmark for evaluating the agency of

LLMs, assessing their decision-making, planning, and interactive capabilities. Reinforcement learning from human feedback (RLHF) is also used to address biases.

LLMs depend on large and high-quality datasets, which involves efficient data collection, preprocessing, and versioning. Ensuring privacy and compliance with regulations like GDPR or CCPA requires secure handling of sensitive information throughout the model lifecycle. LLMOps frameworks streamline model lifecycle management, ensuring efficient deployment and continuous improvement. LLMOps builds on MLOps practice that is already established in smart manufacturing, where automated pipelines manage data generation, training, and deployment of models on the factory floor [43]. Model versioning tracks changes to code, model architectures, training data, and parameters across different iterations. This helps organizations collaborate effectively, speeds up experimentation, and ensures compliance with regulatory or auditing requirements. Version control also supports reproducibility for debugging, analysis, or further development of the models.

Security practices like regular vulnerability assessments, coupled with redundancy measures, ensure model availability and robustness. By integrating these practices, organizations can overcome the complexities of LLM deployment and optimization, achieving scalable and reliable AI systems [44], [45].

LLMs require significant computational resources for training and inference, often necessitating specialized hardware like GPUs or TPUs. This results in both high upfront costs and high operational expenditure, for organizations running models at scale. For real-time applications like chatbots or recommendation systems managing latency and throughput can be a complex task [46]. In cloud-only systems users may be concerned with transmitting personal data. A shift toward hybrid/local deployment is gaining traction, when some or all processing is done locally or (in and edge scenario) close to the user. On-device processing often requires techniques like model pruning and quantization to adapt to limited local resources. NVIDIA TensorRT or ONNX Runtime for deep learning optimization can improve performance. In the long term, managing LLM costs requires a balance between model size, infrastructure investment, and optimized deployment strategies to ensure sustainability without sacrificing performance [47], [48].

E. LLM Applications in Industry

Several recent surveys map the use of LLMs in manufacturing. Ma et al. [49] review LLM applications across intelligent manufacturing and position LLMs as a key enabling technology of Industry 5.0. Zhang et al. [50] propose an LLM-enabled manufacturing framework spanning design, production, and service. Li et al. [51] evaluate the capabilities of general-purpose multimodal models such as GPT-4V on manufacturing tasks ranging from product design to quality control and supply chain management. Sadeqi Bajestani et al. [52] focus on human-in-the-loop integration and argue for a hybrid intelligence model in which LLMs automate tasks while human experts remain responsible for contextual

accuracy and quality assurance – a view closely aligned with the human-centric pillar of Industry 5.0.

The most mature line of application work is the generation of industrial control code. Koziolok et al. [53] tested GPT-4 on 100 prompts in ten categories and found that it frequently produces syntactically correct IEC 61131-3 Structured Text, but that the results still require validation by experienced control engineers. LLM4PLC [54] addresses this by placing the LLM in a loop with grammar checkers, compilers, and formal verification tools, which raised the generation success rate from 47% to 72% and the expert-rated code quality from 3.0 to 7.2 out of 10. In robotics, Code as Policies [55] showed that code-writing LLMs can compose perception and control APIs into executable robot policies from natural-language commands – the same principle that our robot programming use case builds on.

LLM agents are also being used to orchestrate production systems. Xia et al. [56] connect LLM agents to a modular production system through its digital twin and service interfaces, enabling the agents to interpret system information and plan and execute production tasks. Lim et al. [57] extend multi-agent manufacturing systems with LLMs so that agents can interpret human instructions in natural language and coordinate tasks such as G-code allocation among machines. For industrial asset operations and maintenance, AssetOps-Bench [58] provides a benchmark of 141 scenarios in which domain-specific agents for IoT data, failure modes, time-series forecasting, and work orders must collaborate.

Domain adaptation is a recurring theme in engineering assistants. ChipNeMo [59] adapts open LLMs to semiconductor design through domain-specific tokenization, continued pre-training, instruction alignment, and domain-adapted retrieval; its largest variant outperformed GPT-4 on an engineering assistant chatbot and on EDA script generation. Field evidence on the productivity effect of such assistants is still scarce. A notable exception is the study of Brynjolfsson et al. [60], who measured a 15% average increase in issues resolved per hour among more than 5,000 customer support agents using a generative AI assistant, with the largest gains for less experienced workers. On the industrial vendor side, copilots are moving from question answering towards agents that execute complete engineering workflows, such as those announced for the Siemens Industrial Copilot ecosystem [61].

Most of these works examine a single task or a single lifecycle phase, and only a few report results from industrial deployment. The present paper complements them with a cross-case view of LLM applications across engineering, operations, after-sales support, compliance, and knowledge management within one industrial consortium.

III. METHODOLOGY

In this paper we adopt a use-case-driven qualitative methodology to investigate how LLM-based industrial applications contribute to the goals of Industry 5.0. Rather than evaluating a single technical solution, we consider a heterogeneous portfolio of application scenarios that span system-of-systems engineering, customer support, regulatory self-assessment, product

engineering, factual correctness checking, robot programming, ontology and knowledge-graph generation, and AI-gym-based evaluation. All of them were designed and implemented by partners of the AIMS5.0 project, and each is reported in Section IV by the team that developed it. The paper is therefore a multi-case experience report: it describes the cases along common dimensions, states their maturity explicitly, and interprets them against the objectives of Industry 5.0.

A. Methodological premise

The methodological premise is that the value of LLMs in Industry 5.0 cannot be understood solely through task-level performance metrics such as latency, accuracy, or success rate. LLM integration affects industrial work more deeply than a single task: it changes how engineers, operators, and service staff access knowledge, how they interact with machines and software systems, and how responsibility is shared between humans and automated components. The contribution of LLMs must therefore be interpreted in relation to broader socio-technical objectives.

For this reason, the analysis is framed around the three widely recognized Industry 5.0 pillars: human-centricity, sustainability, and resilience. In this work, *human-centricity* refers to the extent to which a use case augments human capabilities, reduces cognitive burden, supports explainability, and preserves meaningful human oversight. *Sustainability* refers not only to direct environmental effects, but also to resource efficiency, waste reduction, knowledge reuse, and lifecycle optimization. *Resilience* refers to the capacity of a system to remain functional, adaptable, safe, and trustworthy under uncertainty, change, partial failure, or evolving regulatory and operational conditions.

B. Case selection and lifecycle coverage

The use cases were selected because they represent distinct stages of the industrial lifecycle, involve different stakeholder groups, and address different forms of cognitive, operational, and organizational complexity. Taken together, they provide a broad empirical basis for examining the role of LLMs beyond narrow automation objectives.

To organize the cases, we follow the lifecycle view of human-centered AI in smart manufacturing proposed by Martini et al. [62], which distinguishes the design, production, and maintenance phases (Figure 1). The design phase is addressed by the Arrowhead Copilots for system-of-systems engineering, the RAG chatbot in product engineering, the ontology and knowledge-graph tools, and the AI Act self-assessment chatbot, which supports compliance checks early in development. The production phase is addressed by robot programming, where generated control code allows a production line to be reconfigured quickly, and by the Arrowhead Management Copilot, which interacts with running services. The maintenance phase is addressed by the support and after-sales chatbot. Factual correctness checking and the AI-gym are cross-phase enablers: they validate the output of LLMs and the behavior of AI models before these are relied upon in any phase. Table ?? lists the lifecycle phase of each case.

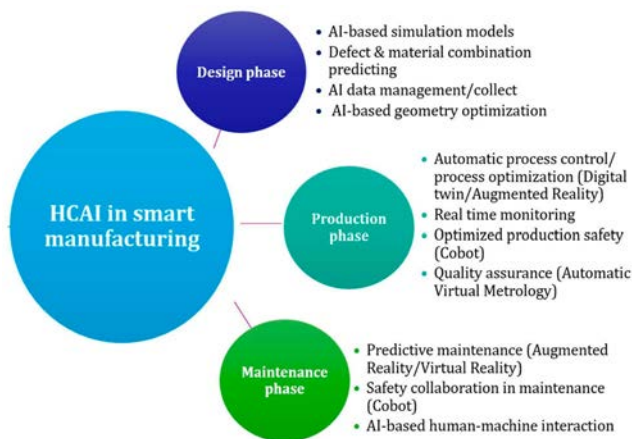


Fig. 1: Human-centered AI (HCAI) in smart manufacturing across the design, production, and maintenance phases. Reproduced from Martini et al. [62] (CC BY). The use cases of this paper are assigned to these phases in Table ??.

C. Analytical framework

The research design is a qualitative cross-case analysis. Each use case is treated as an individual unit of analysis, and all cases are assessed using a common analytical framework. The framework consists of five dimensions.

First, the *industrial context* is identified, including the lifecycle phase addressed by the use case, the primary stakeholder groups involved, and the type of industrial process or ecosystem in which the solution operates.

Second, the *LLM functional role* is characterized, such as retrieval-augmented assistance, code generation, planning and orchestration, compliance guidance, natural-language querying, semantic modeling support, or evaluation in simulated environments.

Third, the *human-system relationship* is analyzed, with particular attention to whether the LLM acts as an assistant, a co-pilot, an autonomous component, or a supervised recommendation engine.

Fourth, the *Industry 5.0 relevance* of the use case is assessed by mapping its expected effects to the three pillars.

Fifth, the *maturity and evidence* of the use case are recorded: how far the solution has been developed, what evidence is available, and which technical and organizational challenges were encountered during design or implementation.

D. Interpretation scheme for the Industry 5.0 pillars

We applied a structured interpretation scheme to make the mapping to the pillars work. A use case is considered to support human-centricity if it reduces the effort required to perform complex engineering or operational tasks, improves accessibility for non-expert users, supports better decision-making, or preserves the possibility of human intervention and accountability. A use case is considered to support sustainability if it contributes to shorter engineering cycles, reduced rework, lower material or energy waste, better resource allocation, more efficient support processes, improved traceability,

or longer useful lifetimes of products and systems through better knowledge access and maintainability. A use case is considered to support resilience if it improves adaptability to changing operating conditions, strengthens robustness against documentation gaps, errors, and uncertainties, enables safer or more reliable system behavior, supports compliance with legal and ethical constraints, or enhances the capability to validate and monitor industrial AI solutions before deployment.

The goal is not to force each use case into all three pillars equally, but to identify its dominant and secondary Industry 5.0 contributions. The methodology therefore allows asymmetric mappings: some use cases are primarily human-centric, others primarily resilience-oriented, while some have stronger sustainability implications through efficiency and optimization effects. This approach reflects the practical reality that Industry 5.0 alignment is often multidimensional but unevenly distributed across applications.

To improve rigor, our methodology distinguishes between direct and indirect links to Industry 5.0. Direct links occur when a use case explicitly addresses a pillar, for example through human-in-the-loop design, sustainability-oriented supply-chain transparency, or safety and compliance mechanisms. Indirect links occur when the primary technical function leads to second-order Industry 5.0 benefits, such as better documentation access leading to less rework, or improved semantic interoperability enabling more robust collaboration across organizations. This distinction prevents overclaiming while still recognizing broader lifecycle effects. It also helps articulate how LLM applications that may initially appear narrow or productivity-focused can still contribute to the strategic goals of Industry 5.0 when embedded into larger industrial workflows.

The mapping was agreed among the contributing authors. It expresses the intended contribution of each solution; the maturity levels described below show how far that contribution has been demonstrated so far.

E. Interpretation of the use cases

From this perspective, the Arrowhead Copilots use case is interpreted as a human-centric and resilience-oriented case, since it reduces the cognitive burden of SoS engineering while improving adaptivity in heterogeneous service environments. The support and after-sales chatbot is primarily human-centric, as it improves accessibility and continuity of service, but it also contributes to sustainability by reducing repetitive manual workload. The AI Act self-assessment chatbot is strongly linked to resilience, because it helps organizations cope with regulatory complexity and compliance risk, while also supporting human-centricity by making governance knowledge accessible to non-specialists. The product engineering RAG chatbot mainly addresses human-centricity and sustainability by improving access to engineering knowledge, reducing duplicated effort, and accelerating document-driven tasks. The factual correctness checking use case is treated as a resilience-enabling mechanism, because it increases trustworthiness in AI-supported query generation and reduces the risk of incorrect machine-mediated decisions. The robot programming use

case spans all three pillars: it is human-centric in reducing programming effort, sustainable in minimizing reconfiguration cost and downtime, and resilient through simulation-based validation and safer deployment of generated code. The ontology generation and knowledge-graph use case contributes to human-centricity by lowering semantic barriers, to sustainability through improved reuse and interoperability of industrial knowledge, and to resilience by enabling robust cross-organizational data integration. Finally, the AI-gym use case is interpreted primarily through the resilience lens, since it provides a controlled environment for safe testing, continuous refinement, and reduced deployment risk, while also indirectly supporting human-centric engineering practices. Table I summarizes this mapping.

F. Limitations of the approach

The case descriptions are written by the teams that developed the solutions, so they are subject to self-reporting bias. The cases use different metrics, and most of them are at the concept or PoC level. The findings should therefore be read as practical experience and design patterns, not as generalizable performance results.

G. Summary of the methodological approach

Hence, the proposed methodology provides a reusable interpretive framework for connecting industrial AI use cases to Industry 5.0. Its central assumption is that the relevance of LLMs in industrial settings should be evaluated not only by technical performance, but also by their capacity to augment people, reduce waste and inefficiency, and strengthen the robustness and adaptability of industrial systems. By applying a common analytical lens to a diverse set of use cases, the study develops a consolidated view of LLMs as enabling components of a more human-centered, sustainable, and resilient industrial future.

IV. APPLICATION AREAS FOR LLMs IN INDUSTRY 5.0

A. Arrowhead Copilots for System-of-Systems Engineering

System-of-systems (SoS) engineering in Industry 5.0 environments involves orchestrating distributed, heterogeneous, and autonomous subsystems, each with its own interfaces, constraints, and lifecycles. Engineers face difficulties in navigating service contracts, maintaining compliance with framework specifications, and designing complex orchestrations across distributed industrial components. Studying documentation, API contracts, and semantic service descriptions is time-consuming and error-prone, particularly when working under time constraints or with legacy systems.

The integration of LLM-based tools into the engineering workflows addresses these systemic problems, speeds up development, and improves productivity. AI agents can quickly search fragmented documentation and assist in interpreting complex service descriptions. Engineers can use natural language instructions, which are automatically translated to programmatic requests – saving the time and effort of learning different subsystem interfaces.

To showcase the power of LLM tools in the system-of-systems engineering process, we developed three agents for the Arrowhead SoS Orchestration Framework, each assisting the workflow in its own way (Figure 2). The framework itself continues to evolve; for example, recent work extends it with service-level synchronization for time-critical applications while preserving late binding and loose coupling [63]. The copilots therefore have to keep up with a growing set of services and documentation.

The *Arrowhead Expert* is a multi-modal chat agent, which uses RAG to quickly find relevant information in a pool of documents about a subsystem or component. The primary advantage of this agent is its ability to handle documents from diverse sources and of various types, such as technical documentation, scientific papers, and flowcharts. The uploaded documents are divided into smaller, semantically coherent segments, which are then converted into embedding vectors, in a way that more fragmented but relevant partitions can also be found effectively. The agent is multi-modal: it can process pictures, tables and figures as well. The responses are accurate, and can even contain relevant document sections or quotes if the user requests more insight.

The *Arrowhead Management Copilot* takes natural language queries in order to programmatically interact with different subsystems. It processes the API documentations of the available components, selects the relevant services, creates and executes the required API calls, and evaluates their results – according to the user's request. The tool was created to handle complex tasks and interact between multiple services, rather than just answer simple queries. To achieve this, we developed a multi-agentic planner engine, which divides the workflow into smaller sub-tasks, each one handled by a specific AI agent. The planner agent devises a plan on how to best achieve the user's goal. The plan is executed step-by-step; each intermediate result is evaluated, and the plan is reviewed if necessary – e.g. on service error or unexpected API call response. The multi-agentic approach allows the planner engine to make changes to its own plan during execution. This makes the tool highly autonomous, adaptive, and robust.

The *Arrowhead Design Copilot* tool helps programmers in software development, SoS orchestration, and the creation of SysML diagrams that integrate into the Arrowhead Framework services.

Using proper guardrail mechanisms, in the Arrowhead Expert we filter out user requests which are not related to the stored documents' domain. In addition to the generic profanity and sensitive content filters, implemented additional guardrails to make sure the queries refer to the use-case context. This has been challenging, because new uploads to the document pool or deleting existing documents may shift the use case domain. The Management Copilot, while being useful and adaptive, struggles interacting with poorly documented components. Some (especially legacy) services don't have proper documentation, do not conform to the documentation, or do not provide helpful error messages. Even if workarounds exist in some cases, such issues need to be considered when selecting the SoS services.

TABLE I
MAPPING OF THE ANALYZED LLM-BASED INDUSTRIAL USE CASES TO THE PILLARS OF INDUSTRY 5.0. PRIMARY CONTRIBUTIONS ARE MARKED WITH X, SECONDARY CONTRIBUTIONS WITH O. THE MARKS EXPRESS THE INTENDED CONTRIBUTION OF EACH SOLUTION.

Use case	Main LLM role	Human-centricity	Sustainability	Resilience	Main contribution to Industry 5.0
Arrowhead Copilots for SoS Engineering	RAG assistance, multi-agent planning, design support	X	o	X	Reduces engineering cognitive load, improves accessibility of complex SoS knowledge, and increases adaptability in heterogeneous service environments.
Chatbots for Support and After-Sales Services	RAG chatbot, ticket handling, human handover	X	o	o	Improves user experience and support accessibility, reduces repetitive manual workload, and preserves continuity through human escalation.
AI Act Self-Assessment Chatbot	Compliance-oriented RAG guidance and reporting	o	o	X	Strengthens governance, legal robustness, and trustworthy deployment of industrial AI by helping developers navigate complex regulations and standards.
RAG Chatbots in Product Engineering	Enterprise knowledge retrieval and grounded answering	X	X	o	Improves access to engineering knowledge, reduces duplicated effort and search time, and supports more efficient engineering workflows.
Factual Correctness Checking	LLM output verification, semantic ranking, NL2SPARQL validation	o	o	X	Increases trustworthiness of AI-assisted querying and reduces the risk of incorrect machine-generated decisions in industrial knowledge access.
Robot Programming	Natural-language-to-code generation with simulation-based validation	X	X	X	Lowers the entry barrier for robot programming, reduces downtime during task changes, and improves safe adaptation through simulation-backed verification.
Ontology Generation and Knowledge Graphs	Ontology support, NL querying, semantic integration	X	X	X	Reduces semantic complexity for users, improves reuse and interoperability of industrial knowledge, and supports robust multi-stakeholder data integration.
Testing LLMs with Arrowhead-compliant AI-gym	Safe evaluation, iterative model refinement, simulation-based testing	o	o	X	Enhances robustness and safe deployment of industrial AI by enabling controlled testing in realistic but low-risk environments.

B. Chatbots for Support and After-Sales Services

Support and after-sales services frequently face the challenge of delivering timely assistance to a diverse customer base, often across multiple languages and for non-critical issues that could be addressed through self-service. These inefficiencies lead to unnecessary workload for human agents and decreased customer satisfaction. A significant portion of customer requests are repetitive and involve information that is already available in company resources, such as manuals or frequently asked questions. However, these documents are often underutilized due to poor accessibility or inadequate formatting for end-user consumption.

To address these issues, the integration of Large Language Models (LLMs) into customer support ecosystems offers a powerful solution. By embedding LLMs within a Retrieval-Augmented Generation (RAG) framework, companies can design chatbots capable of providing intelligent, real-time support based on both structured and unstructured internal data. The cognitive architecture of such a system, depicted in Figure 3, includes three core components: dynamic interaction with existing ticket management systems, retrieval-based access to enterprise knowledge bases, and a live chat interface that facilitates smooth handovers to human agents when necessary. The chatbot interacts with ticketing platforms via APIs to create, update, and categorize tickets based on user input. At the same time, it queries internal documentation – such as technical manuals, past support cases, and FAQs – via retrieval pipelines to generate accurate, context-aware responses. When a query exceeds the chatbot's confidence threshold or requires

expert intervention, it transfers the conversation to a human agent along with the full context to ensure continuity and efficiency.

During implementation, several lessons emerged. A key insight was the importance of user-centered conversational design. Using natural language aligned with the company's tone improves the perceived quality of the interaction. Offering guided prompts and quick replies helps streamline the user journey, especially for frequent tasks. Presenting information dynamically – such as summarizing ticket status or displaying progress indicators – also contributes to better transparency and user trust. Enabling a seamless transition to human agents while preserving the conversational context greatly enhances both customer satisfaction and resolution efficiency.

Despite these advantages, the deployment of LLM-based support chatbots comes with notable limitations. Preparing knowledge assets for retrieval poses a significant challenge. Manuals and technical documents are often multilingual, inconsistently formatted, and include non-textual elements like images and diagrams, which require sophisticated multimodal processing and custom extraction pipelines. Ensuring data privacy and removing sensitive information adds another layer of complexity. Evaluating chatbot performance is also non-trivial. Some metrics, such as response latency and ticket deflection, can be tracked automatically. However, evaluating the accuracy and contextual relevance of the chatbot's answers necessitates the early and ongoing involvement of domain experts. Another critical concern is the risk of confabulated or inappropriate responses. To mitigate this, it is essential to

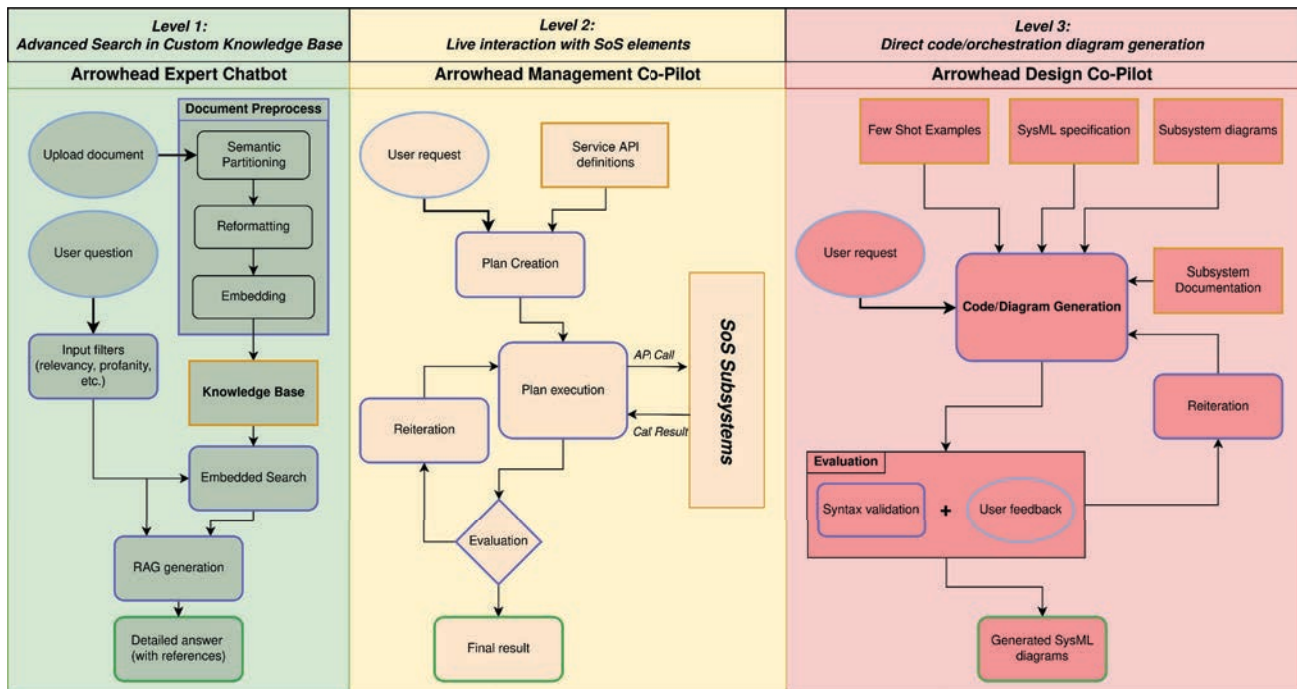


Fig. 2: Structure and workflow of Arrowhead Copilots: Expert Chatbot, Management Copilot, Design Copilot

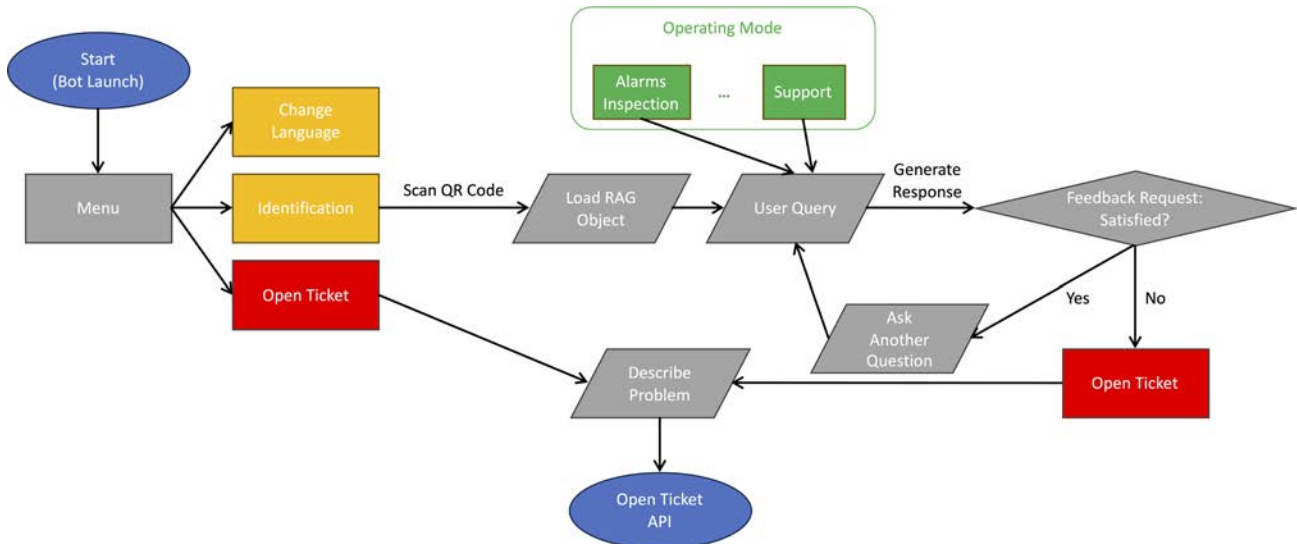


Fig. 3: A schematic description of the chatbot flow for Support and After-Sales Services

implement strict controls based on user identity and role. For instance, users of industrial equipment should never receive instructions to carry out expert-level maintenance procedures which they are not authorized or trained to perform.

In conclusion, the use of LLMs in customer support represents a leap forward in operational efficiency and user experience. By integrating retrieval-augmented reasoning into enterprise systems, LLM-powered chatbots can handle a broad spectrum of support requests autonomously. However, the successful adoption of such systems hinges on careful content preparation, ethical safeguards, and continuous collaboration

with human experts throughout the system lifecycle.

C. AIMS5.0 Chatbot to support self-assessment with the AI Act

The rapid evolution of artificial intelligence technologies enabled transformative applications across industries. These new services also introduced ethics, legal compliance, and governance challenges. As global and regional regulations gain traction – such as the European Union’s AI Act [64] –, developers and publishers of industrial AI systems face increasing complexity in navigating diverse guidelines, standards, and

legal frameworks. This complexity is compounded by the need to ensure ethical soundness, regulatory adherence, and conformance to national and international standards. Although there are tools based on Large Language Models that help users navigate these documents, such as ChatGPT, or tools provided by standardization organizations, these tools are either fundamental inquiry tools or very generic and do not cover copyrighted standards.

To address this challenge, we are building an innovative AI-driven self-assessment tool – a chatbot designed to assist AI developers and users in evaluating the compliance of their applications with ethical guidelines, legal regulations, and industry standards. Leveraging a Retrieval-Augmented Generation (RAG) framework – which combines an LLM with an external knowledge retrieval system – and hosted on cloud infrastructure, the chatbot provides real-time, context-aware guidance, allowing users to assess their AI systems efficiently. This solution not only encourages a consistent approach to ethics and verifies compliance, but also reduces the time and expertise required to navigate complex regulatory landscapes.

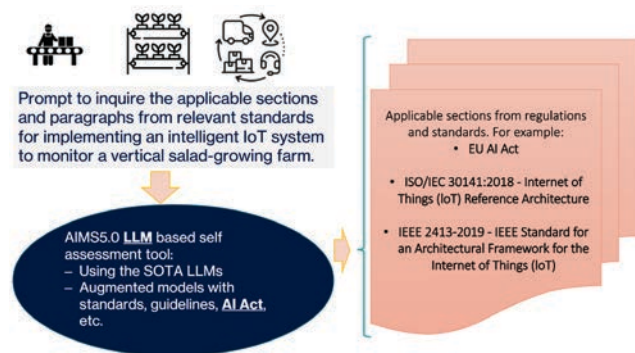


Fig. 4: A schematic description of the self-assessment tool

Integrating AI technology into a Retrieval-Augmented Generation system, we can develop a chatbot that helps application developers check if their work follows legal and ethical rules. This chatbot works as a self-assessment tool that stores important AI regulations, standards, and ethical guidelines in a local database. When developers chat with the system and answer specific questions about their AI project, the chatbot searches through this database to find relevant information and creates a personalized report showing which regulations, standards, and guidelines apply to their situation. This allows developers to check their applications early in the development process, ensuring they meet compliance requirements, safety standards, and ethical guidelines, which builds user trust and helps avoid regulatory violations. The system includes key documents such as the EU AI Act, Digital Services Act (DSA), Digital Markets Act (DMA), Transparency Standard, Bias Standard, HLEG guidelines on trustworthy AI, and various domain-related and technical standards. Figure 4 illustrates the schematic arrangement of the self-assessment tool.

D. RAG Chatbots in Product Engineering

In Signify (formerly Philips Lighting) the integration of a Retrieval-Augmented Generation chatbot in product engineer-

ing aims to improve productivity and quality by providing easy access to critical engineering documents. The SharePoint server hosts product compliance guidelines, manufacturing processes, design rules, component information, and design tool documentation. The current proof-of-concept is a “naive” RAG solution built on Azure components as shown in Figure 5. The initial RAG system combines the Azure OpenAI On Your Data service with Azure AI Search, and gets updated when new frameworks are available under Azure. The Engineering Portal documents have been indexed with vector embeddings in Azure AI Search. The Azure OpenAI On Your Data service takes the user question with the conversation history and uses an LLM to determine the user intent. The intent is used to query Azure AI Search to produce relevant context, which is then used by Azure OpenAI On Your Data service to make another LLM completion call to generate the response. While this Azure-based approach leverages a robust cloud ecosystem, its implementation has revealed several key challenges:

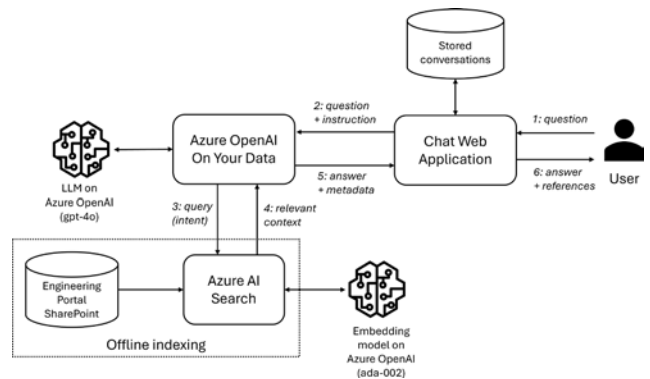


Fig. 5: Azure based architecture of Engineering Portal Chat.

Deployment and Data Security: The solution is deployed entirely within Azure. A SaaS solution may offer ease of deployment but raises concerns about data leakage, especially when sensitive engineering documents are processed by cloud-based language models. Ensuring data confidentiality is critical, as exposure of proprietary information could have severe implications for the company.

Fine-Grained Access Control: The RAG system must enforce document-level permissions, ensuring that users can only access information they are authorized to view. In a “naive” RAG implementation unauthorized documents can be filtered out in the retrieval step. In more sophisticated architectures, like graph-based RAG or hierarchical models such as RAPTOR, security enforcement is more challenging to implement properly.

Integrating Distributed and Multi-Modal Knowledge: The Engineering Portal documents contain links to web pages and documents stored in other systems, comprising different data modalities such as text documents, tables, graphs, and presentations. Aggregating this distributed and multi-modal data into a unified system is an open challenge and likely demands advanced RAG solutions.

Maintaining Data Currency: Engineering documents are

being updated perpetually. Our RAG implementation must regularly check for changes in the SharePoint content and re-index it, so that users always receive the most up-to-date technical references. When improving quality with more advanced RAG solutions, regular and incremental updates should be accounted for.

Ensuring Response Quality and Accuracy: All steps of the RAG process – indexing, retrieval and generation – contribute to response quality. The system prompt's wording also plays a crucial role in achieving meaningful results. In our implementation, the prompt defines the knowledge base scope and establishes answering rules that constrain the LLM to grounded responses while preventing confabulations such as fabricating formulas. Consistently delivering factually accurate and detailed responses remains a significant challenge, particularly when dealing with technical specifications or complex design rules.

Enhancing User Experience: A smooth user experience is essential, because many engineers may be unfamiliar with AI chatbot interactions. The customized system prompt serves two purposes: it instructs the LLM on content boundaries, and informs users about the chatbot's scope and limitations. The users can provide feedback about the chatbot's performance, which is stored alongside the conversations. Analyzing feedback and chatbot usage patterns is essential for improvement, but maintaining privacy remains our top priority.

This exploration of RAG chatbot integration in product engineering has highlighted fundamental challenges in industrial LLM applications. Resolving these challenges is essential for realizing the complete potential of LLMs in industry.

E. Factual Correctness Checking

Artificial intelligence-generated content can substantially improve decision-making across various industries. Integrating large language models with knowledge graphs enables natural language (NL) querying, allowing non-experts to access complex data without needing to master SPARQL. However, because LLMs can generate non-factual or inaccurate outputs, it is critical to ensure the correctness of the translated queries.

The translation of NL to SPARQL (NL2SPARQL) poses multiple challenges. Proprietary models such as GPT-4 [65] often cannot be used in industrial settings, because queries and schema information would have to leave the organization. Open-weight models that can be deployed on premises have narrowed the gap to proprietary models considerably: DeepSeek-V3 [66], DeepSeek-R1 [67], Qwen3 [68], and gpt-oss [69] report performance comparable to leading closed models on reasoning and coding benchmarks, and gpt-oss-120b has been used for SPARQL generation over a corporate knowledge graph [70]. General capability, however, does not guarantee correct queries over a private schema: the model has to use the classes and properties of the target ontology correctly, and the NL-SPARQL pairs needed to fine-tune or evaluate a model for a specific knowledge graph are hard to obtain. For this reason, the first TEXT2SPARQL challenge evaluated systems on a corporate knowledge graph in addition to DBpedia [71]. Methods that verify the semantic accuracy of a generated query before it is executed remain scarce.

To overcome these issues, we propose an approach in which an open-source language model generates multiple candidate SPARQL queries which are then ranked by their semantic similarity to the original NL query. Each SPARQL candidate is encoded into a dense representation and compared with the dense representation of the NL question using a semantic similarity model. For encoding, BERT [72] can be employed, fine-tuned with adapters on both NL questions and SPARQL queries. A multilayer perceptron (MLP) can be trained for similarity scoring. Training this semantic similarity model requires a parallel dataset of NL questions and SPARQL queries [73].

In the processing flow (Figure 6a), from the initial NL question the LLM generates k SPARQL candidates, which are ranked by their semantic similarity to the NL question. The highest-ranking candidate is then used to query the knowledge graph. A candidate is deemed correct if its output matches that of an expert-crafted query, as illustrated in Figure 6b. The collection of expert-crafted queries is incorporated as part of the use-case. This pipeline generates multiple candidates by prompt-tuning of LLMs. As there is no need to train models from scratch, it is sufficient to fine-tune a smaller encoder and train only a small-scale MLP. We assume that generating multiple candidates increases the likelihood of producing a correct symbolic representation. This builds trust in LLM-generated queries, and promotes the adoption of AI solutions in industrial processes.

While the conceptual framework and system architecture have been fully specified, the proposed pipeline has not yet undergone empirical validation. A comprehensive evaluation—including ablation studies on benchmark datasets such as LC-QuAD 2.0 and domain-specific knowledge graphs—is planned for the next phase of this research. These forthcoming experiments will measure (i) the recall of correct SPARQL candidates generated by the LLM, (ii) the precision of the semantic-similarity ranker, and (iii) overall question-answering accuracy compared with current NL2SPARQL baselines. We expect that demonstrating robust performance across these metrics will provide clear guidelines for deployment in production settings. Results from these studies, together with an open-source reference implementation, will be reported in a subsequent publication.

F. Robot Programming

Industrial robots play an important role in modern manufacturing, contributing to precise and efficient automation of production processes. However, it is still difficult to provide detailed task specifications and implement them as robot control programs. Unpredictably positioned objects and variability of motion in the robot's environment are particularly challenging, often requiring integration of machine vision and real-time motion planning into the robot control system. These factors slow down the adoption of robots in industrial automation settings.

Generative AI holds the promise of generating control code from informal, high-level task descriptions. Trained on already available program samples, large language models

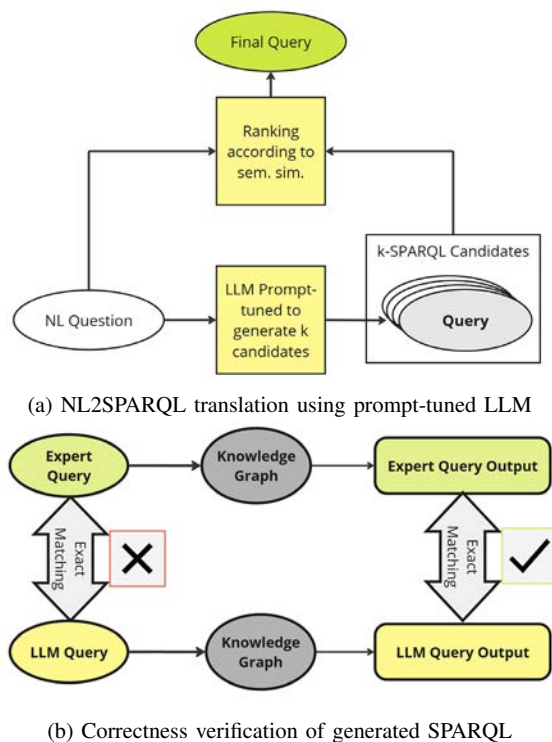


Fig. 6: Overview of NL2SPARQL translation and verification pipeline.

can translate natural language instructions into robot control source code. This reduces the reliance on robotics specialists, democratizes industrial robot programming, and cuts the time required for coding or production line reconfiguration. But applying generative AI introduces its own technical challenges. It is not trivial to ensure that the generated code invokes the machine vision and robot control APIs correctly. Robust verification mechanisms should be in place to rule out unintended or unsafe robot operation and guarantee successful task completion across a range of component positions and motion variations on the production line.

The focus of the AIMS5.0 project's Use Case 6 is to develop a robotic feeding system for a beverage bottling line. Our objective is to quickly switch tasks or introduce new tasks in the processing workflow, with minimal downtime. Tasks are defined by bottle kinds (differing in shape, size, weight, material, color); by bottle carrier-block types; and whether the bottles are piled randomly or neatly layered in the boxes delivered to the production line. The flow of carrier-blocks fluctuates in concordance with the complexity of the processing steps downstream on the production line, and is thus irregular and unpredictable at the bottle insertion station.

For our purposes, the key performance indicators are reliability, processing speed/throughput, safety, and reconfigurability. The entire pick-and-place pipeline (including computer vision, grasp pose estimation, motion planning, and dynamic control) should be fast enough to maintain the current throughput of 40–70 bottles per minute. Safety considerations are

critical – the risk of damage to health or equipment should be minimized to the extent engineering permits. Within these constraints the goal is to minimize the time, effort, and expertise required to switch from one task to another or to introduce a new one.

Figure 7 shows the high-level architecture of the code generation system. The LLM produces two kinds of source code: configuration code for task simulation, and robot control scripts. The task configuration code serves two functions: initializes randomized task simulations, and monitors task progress within the simulation – in order to provide feedback on task success. The robot control scripts call machine vision, motion planning, and robot control components which perform the actual task; such components are increasingly learned models themselves, for example transformer-based path planners [74].

The two kinds of source code are generated independently for multiple reasons: (1) This makes it easier to separate code targeting the simulation environment from code which interacts with the elevated privileges of real-world APIs; (2) the potential bugs in the generated code can be vetted and fixed independently; and (3) the proven versions are saved into distinct repositories of in-context examples for future code generation queries.

The system is intended to be adaptable to applications beyond the specific bottling use case described above. However, two key design assumptions limit the system's scope of applicability: (1) We assume a "closed world" scenario: all encountered object types are defined beforehand (with the option to add new types as needed). The physics models and all objects' spatial characteristics are known in advance. This is a reasonable assumption for the use case's stationary robots in the current manufacturing setting. (2) Rather than relying on large Language+Vision [75]–[77] or Vision+Action models [78], [79] to make inferences about object locations and orientations, we decide to invoke fine-tuned perception and grasp-estimation models from the robot control scripts. The limited-scope models are much smaller and faster, and thus can be run locally at the robot – rather than communicating with an inference server at a remote location. We also found these smaller models more predictable.

Some of the KPIs are directly incorporated into the code generation system. For instance, the generated robot control scripts are automatically evaluated in simulation for reliability, by executing them in multiple randomized simulation instances. Code generation contributes to speed and safety at the same time: vetted and approved scripts are handled as stand-alone, deterministic code modules; they can be inspected and deployed independently of the process that generated them.

The code generation system is evaluated using a manually curated, small set of example tasks and a fixed budget of generation attempts from natural language task descriptions. The results are derived from the following statistics: *Simulation Generation Success Rate*: how often the system generates usable simulation configurations; *Simulation Generation Partial Success Rate*: how often the system generates a simulation configuration that can be fixed with low effort by human

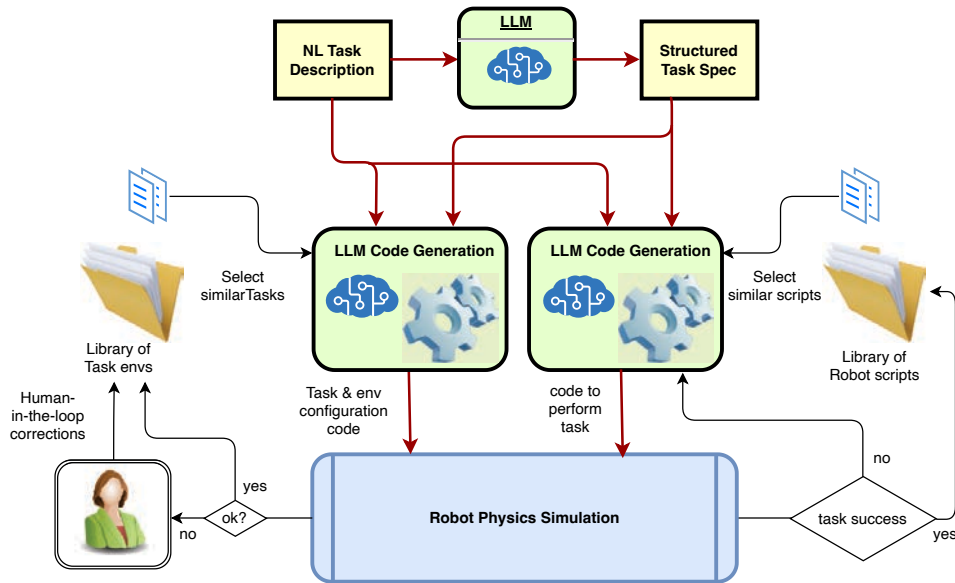


Fig. 7: From NL description, an LLM generates two kinds of scripts: simulation configuration, and high level robot control

intervention (cf.: human-in-the-loop effort/time); and *Script Generation Success Rate*: how often the system generates a robot control script that succeeds across a number of randomized simulation instances.

These metrics vary depending on the code-generation LLMs employed (together with the quality and number of few-shot examples for prompting). The metrics also depend on the sophistication of the physical simulation model, and thus on the generated code. All of these technologies are evolving rapidly, so detailed results at this stage would not be particularly informative. However, an initial proof of concept using GPT-4o as the code generation model for two basic tasks, along with a highly abstracted robot simulation in the *pybullet* environment, demonstrated that the system can generate viable robot control scripts. The results were deemed promising enough to validate the approach, and efforts are currently under way to implement a higher fidelity simulation environment incorporating more realistic robot perception and control primitives. A higher-fidelity simulation does not remove the gap between simulated and real perception; techniques such as visual domain randomization combined with real images [80] can make behavior validated in simulation more robust on the physical line. By outlining the key challenges, detailing the design choices and methods, and presenting an initial proof of concept, this use case aims to offer insights into how code generation AI models could broaden the range of manufacturing environments where industrial robots can economically be deployed.

G. Ontology Generation and Knowledge Graphs

The semiconductor industry requires integration of data generated by interconnected manufacturing processes, supply chains, and sustainability initiatives. Fragmented data

sources, inconsistent data formats, and poor interoperability among systems complicate this process. These issues hinder effective decision-making in core areas such as supply chain management, manufacturing optimization, and environmental impact tracking. For example, supply chain contributors often struggle to trace components and materials due to incompatible data structures. Similarly, isolated silos prevent manufacturing processes from getting real-time data insights. Addressing these challenges requires a common semantic representation, standardized knowledge-sharing mechanisms, and tools to enable efficient collaboration on data-intensive tasks.

The Semantic Web extends traditional web technologies by embedding machine-readable meaning into data through ontologies and knowledge graphs [81]. Ontologies serve as formal vocabularies, defining concepts and relationships in a way that both machines and humans can interpret. Knowledge graphs represent these ontologies as networks of interconnected industrial knowledge. The ontologies' semantic foundation enables systems to reason about complex manufacturing processes, supply chain relationships, and human-machine interactions.

The AIMS5.0 Open Access Platform represents a practical implementation of Semantic Web principles in industrial settings. The platform architecture is specifically tailored to modern semiconductor manufacturing processes and their extended supply chains, where data heterogeneity and interdependence create further operational complexities.

The platform's architecture (illustrated in Figure 8) allows the semantic integration of multiple datasets and ensures secure data exchange. Specific tools help stakeholders to map their data into knowledge graphs for aggregated data analysis and semantic reasoning. Large language models and AI-driven enhancements promote intuitive interactions with

data, reducing operational complexity and users' cognitive overload. This feature is particularly helpful for users who are unfamiliar with semantic technologies.

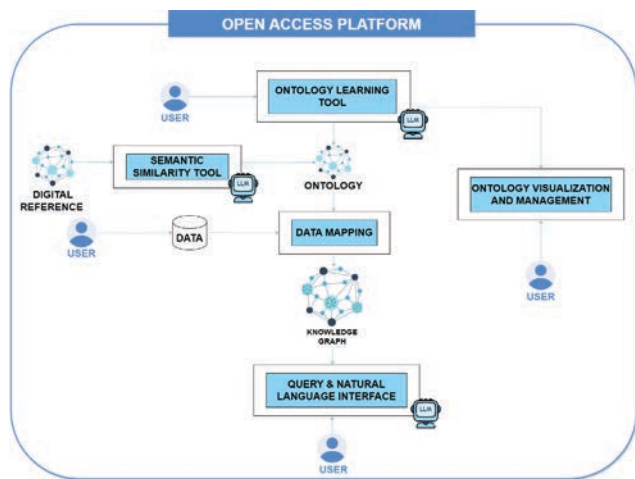


Fig. 8: Open Access Platform architecture

The AIMS5.0 platform architecture consists of two primary functional areas: a knowledge sharing infrastructure and a secure data transfer framework. In both areas LLM-based AI tools help reduce technical complexity, streamline workflows, and enable intuitive interaction with data.

The knowledge sharing infrastructure enables the creation, alignment, and management of semantic models through ontologies and knowledge graphs. Users can create domain-specific ontologies tailored to their datasets, even without prior expertise in semantic modeling. Tools like Chowlk assist in automatically converting data schemas into formal ontology files, simplifying the traditionally complex process of ontology creation. We added LLM-based tools for schema extraction from unstructured data, further reducing users' cognitive burden, and increasing accessibility for non-experts.

To ensure interoperability, user-created ontologies are aligned with the *Digital Reference* model [82], shown in Figure 9, which provides a comprehensive semantic representation of the semiconductor supply chain – comprising sub-ontologies for products, sensors, organizations, and business processes. This alignment minimizes redundancy, promotes reuse, and ensures seamless data exchange across stakeholders.

After ontology creation or alignment, datasets are semantically mapped to the data models. Visual mapping tools, such as OpenRefine, offer an intuitive graphical interface for users with limited technical knowledge. Expert users can rely on flexible, traditional mapping tools for greater control. Following semantic mapping, knowledge graphs are generated to represent the relationships among data entities, enabling semantic reasoning and analysis. Knowledge graphs are stored in graph databases, which support efficient querying through SPARQL and Cypher endpoints.

Large language models simplify user interactions with the knowledge graphs. For example, users can now pose questions in natural language, which are translated into SPARQL or Cypher queries. This feature allows access to semantic

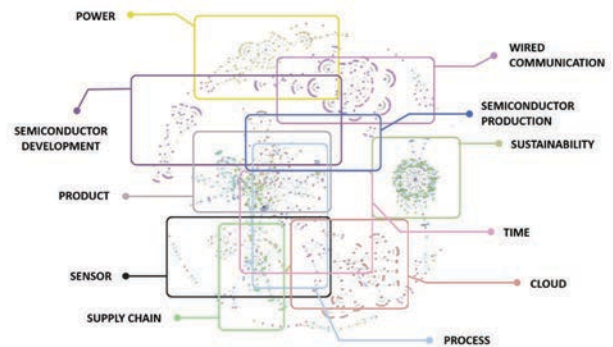


Fig. 9: Digital Reference Model of semantic representations

technologies for non-technical users and a wider range of stakeholders. By lowering technical barriers, these enhancements reduce cognitive overload and speed up the adoption of semantic, data-driven approaches.

The secure data transfer framework complements the knowledge sharing infrastructure by ensuring encrypted and controlled access to sensitive industrial data. Semantic integration across distributed datasets allows for collaborative analysis of complex ecosystems, such as interconnected supply chains. The framework also ensures that stakeholders can securely share insights derived from their knowledge graphs, promoting trust and collaboration.

The development of the AIMS5.0 platform has highlighted the importance of simplifying workflows for users. Early efforts show that LLM-driven enhancements, such as natural language querying and automated ontology creation, can greatly reduce cognitive overload and improve accessibility for non-technical stakeholders. Semantic integration, particularly through alignment with the Digital Reference model, highlights the value of interoperability and collaboration across fragmented datasets. Intuitive tools for mapping and visualizing knowledge graphs are proving critical for encouraging adoption and enabling data-driven decision-making.

However, several challenges remain. Ontology creation still demands considerable domain expertise and effort, which can slow down progress. Aligning legacy systems with the semantic framework is difficult, especially when data is incomplete or inconsistent. Some users, particularly those with limited technical expertise, may struggle to grasp semantic modeling concepts and interact with knowledge graphs. While LLM-based natural language querying addresses these issues, its effectiveness has not yet been fully validated. As industrial requirements evolve, scalability and long-term maintenance of ontologies and knowledge graphs become another key focus area.

Large language models have significant potential to address many of the platform's shortcomings. In ontology creation, LLMs can automate schema extraction from unstructured datasets, reducing the dependency on domain experts and accelerating the modeling process [83]. LLMs can improve ontology alignment by identifying semantic similarities between concepts, enabling seamless integration of multiple

ontologies [84].

LLM-driven enhancements can further simplify querying by translating natural language questions into SPARQL or Cypher queries, making semantic tools more accessible to non-technical users. Beyond querying, LLMs can support knowledge graph population through tasks such as entity recognition and dependency parsing, streamlining the integration of new datasets into the platform.

The AIMS5.0 Open Access Platform exemplifies the transformative potential of Semantic Web and AI technologies in addressing challenges of the semiconductor industry. By enabling semantic integration, knowledge sharing, and advanced reasoning, the platform has improved supply chain transparency, optimized manufacturing processes, and supported sustainability efforts. With the integration of LLM-driven enhancements, the platform is poised to further reduce complexity, improve usability, and drive growth and sustainability across the industry.

H. Testing LLMs with Eclipse Arrowhead-compliant AI-gym

Testing Large Language Models (LLMs) presents unique challenges in evaluating their robustness, ethics, biases, trustworthiness, and ability to adapt to dynamic real-world scenarios. Traditional testing methods for LLMs, such as static benchmarks and isolated evaluation metrics [85], have notable limitations. Chang et al. [86], highlight that current LLMs have limited capabilities in reasoning and robustness. They also underscore the pressing need for modern evaluation systems, so that the models' capabilities and constraints can be assessed more accurately. This calls for more robust evaluation frameworks that test LLMs in interactive, diverse, and evolving settings. The AI-gym concept offers a promising solution by providing simulated, interactive environments where LLMs can be tested across diverse tasks and evolving contexts. These environments allow for systematic benchmarking, continuous feedback loops, and safe exploration of complex scenarios, enabling developers to better assess and improve LLM performance in controlled yet flexible settings.

In this section we present an AI-gym implementation – a controlled environment designed for developing, training, and testing of AI models under in simulated real-world industrial scenarios. The framework enables iterative model refinement, accurate testing, and seamless integration with Industrial Internet-of-Things (IIoT) systems by integrating Apache Zeppelin [87], Eclipse Arrowhead [88], and IIoT data streams tools. Its features include dynamic interaction with live data, interoperability with existing IIoT frameworks, reduced deployment risks by model testing in production-like conditions, enhanced model robustness through real-time performance evaluations, and accelerated software development.

The AI-gym architecture as depicted in Figure 10 consists of three-layers that combines Apache Zeppelin, Eclipse Arrowhead, and IIoT components:

The Model Development Layer focuses on AI model development and quality assurance. Apache Zeppelin – the open-source notebook developed for collaborative data analysis – enables developers to write, test, and visualize their code

interactively. Python, Scala, and various other supported interpreters allow experimentation with a variety of machine learning frameworks. Developers can iteratively refine their models with feedback from data visualizations and performance metrics. Zeppelin's integrated quality checks help catching bugs before proceeding to the next stage of the AI development pipeline.

The Model Integration Layer uses the Eclipse Arrowhead Adapter to incorporate the AI models into larger IIoT ecosystems. The role of this layer is to connect Zeppelin to Eclipse Arrowhead's service-oriented architecture. The adapter merges AI solutions into Arrowhead's local cloud IIoT ecosystem, efficiently orchestrating data flow and service discovery. This layer also handles the security and interoperability functions, ensuring that the AI models adhere to industry standards while interacting with IIoT devices and frameworks.

Finally, the IIoT Data Layer tests the AI models under realistic conditions using Eclipse Arrowhead Local Cloud's IIoT data streams. This layer simulates production-like environments where AI models are exposed to dynamic inputs from IIoT devices: sensors, actuators, and industrial machines. The models' performance, scalability, and reliability are tested under varying conditions. Only authorized services can interact with the models, safeguarding sensitive data and operations by using Arrowhead's secure and modular architecture.

V. CONCLUSIONS

This paper presented a qualitative multi-case view of how LLMs and AI copilots can be integrated into industrial engineering workflows within the AIMS5.0 project. The cases cover different stages of the industrial lifecycle and range from conceptual designs to proofs of concept and pilot implementations. Taken together, they show how LLM-based systems can support engineers through knowledge access, natural-language interaction, software assistance, and controlled interaction with industrial systems.

Several technical patterns recur across the cases. Retrieval-Augmented Generation and semantic search provide access to domain-specific knowledge, while multi-agent approaches extend LLMs from question answering toward service orchestration. Other cases use LLMs to generate engineering artifacts or simplify interaction with semantic technologies. As these systems move closer to executable code or operational services, verification, access control, simulation, and human supervision become increasingly important.

The implementation experience also reveals a common set of challenges. Industrial knowledge is often distributed across heterogeneous, evolving, and multimodal sources. This makes document preparation, indexing, version management, and data currency central concerns. Data confidentiality and fine-grained access control are equally important when proprietary information is involved. In addition, factual correctness remains difficult to guarantee in specialized domains. Integration with legacy systems can create further difficulties, especially when interfaces are poorly documented or inconsistent.

The maturity of the demonstrators differ across the shown use cases. Most of the solutions are still at the proof-of-concept stage, and some have reached pilot use. The findings

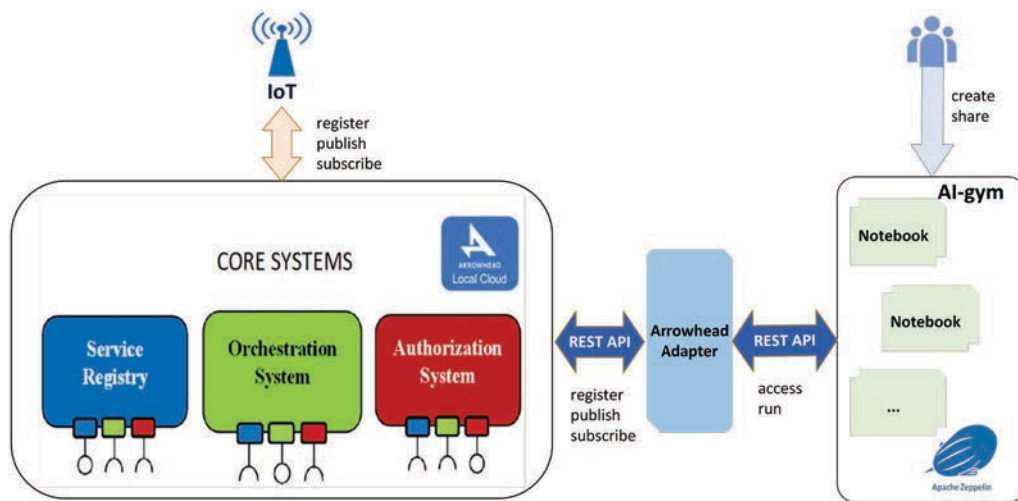


Fig. 10: AI-gym architecture.

of this paper are therefore best understood as implementation experience, design patterns, and practical lessons.

From an Industry 5.0 perspective, the clearest contribution of the presented systems is to human-centric engineering. They can make complex technical knowledge easier to access, reduce barriers to specialized tools, and support engineers while preserving human oversight. Some cases also contribute to resilience through verification, compliance support, controlled testing, and adaptation to heterogeneous systems. Sustainability effects are generally more indirect and arise through improved knowledge reuse, reduced rework, maintainability, and more efficient engineering processes. These effects require further empirical evaluation before broader conclusions can be drawn.

Future work should therefore focus on systematic evaluation in realistic industrial settings. Comparable measures are needed for response quality, task success, human effort, robustness, and operational impact. Further research is also required on secure and maintainable RAG architectures, domain adaptation, evolving knowledge bases, and verification of generated engineering artifacts. Longitudinal studies with industrial users would help distinguish projected benefits from effects that can be measured in routine operation.

Overall, the AIMS5.0 experience suggests that LLMs can become useful components of Industry 5.0 when they are embedded in well-designed technical and organizational safeguards. Their practical value depends not only on the capabilities of the language model, but also on the quality of the underlying knowledge, reliable system integration, explicit verification mechanisms, and clearly defined human responsibility. In this role, LLM-based systems are best viewed as tools that augment industrial expertise rather than replace it.

ACKNOWLEDGMENT

The research leading to these results was conducted in the AIMS5.0 project, which received funding from the Chips Joint Undertaking under grant agreement No. 101112089 and from the partners' national funding programmes.

REFERENCES

- [1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *CoRR*, vol. abs/1706.03762, 2017. [Online]. Available: <http://arxiv.org/abs/1706.03762>
- [2] T. B. Brown, "Language models are few-shot learners," *arXiv preprint arXiv:2005.14165*, 2020.
- [3] R. Hendel, M. Geva, and A. Globerson, "In-context learning creates task vectors," *arXiv preprint arXiv:2310.15916*, 2023.
- [4] S. Lu, I. Bigoulaeva, R. Sachdeva, H. T. Madabushi, and I. Gurevych, "Are emergent abilities in large language models just in-context learning? arxiv," *arXiv preprint arXiv:2309.01809*, 2023.
- [5] C. Zhou, J. He, X. Ma, T. Berg-Kirkpatrick, and G. Neubig, "Prompt consistency for zero-shot task generalization," *arXiv preprint arXiv:2205.00049*, 2022.
- [6] V. Sanh, A. Webson, C. Raffel, S. H. Bach, L. Sutawika, Z. Alyafeai, A. Chaffin, A. Stiegler, T. L. Scao, A. Raja et al., "Multitask prompted training enables zero-shot task generalization," *arXiv preprint arXiv:2110.08207*, 2021.
- [7] Z. G. Yang and N. Ligeti-Nagy, "Improve performance of finetuning language models with prompting," *Infocommunications Journal*, vol. XV, no. Special Issue on Applied Informatics, pp. 62–68, 2023.
- [8] J. White, Q. Fu, S. Hays, M. Sandborn, C. Olea, H. Gilbert, A. Elnashar, J. Spencer-Smith, and D. C. Schmidt, "A prompt pattern catalog to enhance prompt engineering with chatgpt," *arXiv preprint arXiv:2302.11382*, 2023.
- [9] J. Zamfirescu-Pereira, R. Y. Wong, B. Hartmann, and Q. Yang, "Why johnny can't prompt: how non-ai experts try (and fail) to design llm prompts," in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 2023, pp. 1–21.
- [10] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou et al., "Chain-of-thought prompting elicits reasoning in large language models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 24 824–24 837, 2022.
- [11] S. Yao, D. Yu, J. Zhao, I. Shafran, T. L. Griffiths, Y. Cao, and K. Narasimhan, "Tree of thoughts: Deliberate problem solving with large language models," 2023.
- [12] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery, and D. Zhou, "Self-consistency improves chain of thought reasoning in language models," 2023.
- [13] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, "ReAct: Synergizing Reasoning and Acting in Language Models," *International Conference on Learning Representations (ICLR)*, Jan. 2023.
- [14] L. Gao, A. Madaan, S. Zhou, U. Alon, P. Liu, Y. Yang, J. Callan, and G. Neubig, "Pal: Program-aided language models," in *International Conference on Machine Learning*. PMLR, 2023, pp. 10 764–10 799.

- [15] N. Shinn, F. Cassano, A. Gopinath, K. Narasimhan, and S. Yao, “Reflexion: Language agents with verbal reinforcement learning,” *Advances in Neural Information Processing Systems*, vol. 36, 2024.
- [16] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, M. Wang, and H. Wang, “Retrieval-augmented generation for large language models: A survey,” 2024. [Online]. Available: <https://arxiv.org/abs/2312.10997>
- [17] M. Antal, “Evaluation of embedding models for Hungarian question-answer retrieval on domain-specific and public benchmarks,” *Infocommunications Journal*, vol. XVII, no. 4, pp. 2–11, 2025.
- [18] P. Sarthi, S. Abdullah, A. Tuli, S. Khanna, A. Goldie, and C. D. Manning, “Raptor: Recursive abstractive processing for tree-organized retrieval,” in *International Conference on Learning Representations (ICLR)*, 2024.
- [19] T. T. Procko and O. Ochoa, “Graph retrieval-augmented generation for large language models: A survey,” in *2024 Conference on AI, Science, Engineering, and Technology (AIxSET)*, 2024, pp. 166–169.
- [20] A. Singh, A. Ehtesham, S. Kumar, and T. T. Khoei, “Agentic retrieval-augmented generation: A survey on agentic rag,” 2025. [Online]. Available: <https://arxiv.org/abs/2501.09136>
- [21] X. Hou, Y. Zhao, Y. Liu, Z. Yang, K. Wang, L. Li, X. Luo, D. Lo, J. Grundy, and H. Wang, “Large language models for software engineering: A systematic literature review,” *ACM Trans. Softw. Eng. Methodol.*, Sep. 2024, just Accepted. [Online]. Available: <https://doi.org/10.1145/3695988>
- [22] Z. Zhang, C. Chen, B. Liu, C. Liao, Z. Gong, H. Yu, J. Li, and R. Wang, “Unifying the Perspectives of NLP and Software Engineering: A Survey on Language Models for Code,” p. arXiv:2311.07989, Nov. 2023.
- [23] X. Wang, B. Li, Y. Song, F. F. Xu, X. Tang, M. Zhuge, J. Pan, Y. Song, B. Li, J. Singh et al., “Openhands: An open platform for ai software developers as generalist agents,” in *The Thirteenth International Conference on Learning Representations*, 2024.
- [24] T. Ridnik, D. Kredo, and I. Friedman, “Code generation with alphacodium: From prompt engineering to flow engineering,” 2024. [Online]. Available: <https://arxiv.org/abs/2401.08500>
- [25] S. Peng, E. Kalliamvakou, P. Cihon, and M. Demirel, “The impact of ai on developer productivity: Evidence from github copilot,” 2023. [Online]. Available: <https://arxiv.org/abs/2302.06590>
- [26] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. de Oliveira Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman, A. Ray, R. Puri, G. Krueger, M. Petrov, H. Khlaaf, G. Sastry, P. Mishkin, B. Chan, S. Gray, N. Ryder, M. Pavlov, A. Power, L. Kaiser, M. Bavarian, C. Winter, P. Tillet, F. P. Such, D. Cummings, M. Plappert, F. Chantzis, E. Barnes, A. Herbert-Voss, W. H. Guss, A. Nichol, A. Paino, N. Tezak, J. Tang, I. Babuschkin, S. Balaji, S. Jain, W. Saunders, C. Hesse, A. N. Carr, J. Leike, J. Achiam, V. Misra, E. Morikawa, A. Radford, M. Knight, M. Brundage, M. Murati, K. Mayer, P. Welinder, B. McGrew, D. Amodei, S. McCandlish, I. Sutskever, and W. Zaremba, “Evaluating large language models trained on code,” 2021. [Online]. Available: <https://arxiv.org/abs/2107.03374>
- [27] J. Thorne, A. Vlachos, C. Christodoulopoulos, and A. Mittal, “Fever: a large-scale dataset for fact extraction and verification,” *arXiv preprint arXiv:1803.05355*, 2018.
- [28] A. Pagnoni, V. Balachandran, and Y. Tsvetkov, “Understanding factuality in abstractive summarization with frank: A benchmark for factuality metrics,” *arXiv preprint arXiv:2104.13346*, 2021.
- [29] B. Goodrich, V. Rao, P. J. Liu, and M. Saleh, “Assessing the factual accuracy of generated text,” in *proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 166–175.
- [30] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, “Squad: 100,000+ questions for machine comprehension of text,” *arXiv preprint arXiv:1606.05250*, 2016.
- [31] S. Min, K. Krishna, X. Lyu, M. Lewis, W.-t. Yih, P. W. Koh, M. Iyyer, L. Zettlemoyer, and H. Hajishirzi, “FactScore: Fine-grained atomic evaluation of factual precision in long form text generation,” *arXiv preprint arXiv:2305.14251*, 2023.
- [32] A. Wang, K. Cho, and M. Lewis, “Asking and answering questions to evaluate the factual consistency of summaries,” *arXiv preprint arXiv:2004.04228*, 2020.
- [33] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, “BertScore: Evaluating text generation with bert,” 2020.
- [34] W. Kryściński, B. McCann, C. Xiong, and R. Socher, “Evaluating the factual consistency of abstractive text summarization,” *arXiv preprint arXiv:1910.12840*, 2019.
- [35] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano et al., “Training verifiers to solve math word problems,” *arXiv preprint arXiv:2110.14168*, 2021.
- [36] S. Han, H. Schoelkopf, Y. Zhao, Z. Qi, M. Riddell, L. Benson, L. Sun, E. Zubova, Y. Qiao, M. Burtell et al., “Folio: Natural language reasoning with first-order logic,” *arXiv preprint arXiv:2209.00840*, 2022.
- [37] J. Liu, L. Cui, H. Liu, D. Huang, Y. Wang, and Y. Zhang, “Logiqa: A challenge dataset for machine reading comprehension with logical reasoning,” *arXiv preprint arXiv:2007.08124*, 2020.
- [38] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “Bleu: a method for automatic evaluation of machine translation,” in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.
- [39] C.-Y. Lin, “ROUGE: A package for automatic evaluation of summaries,” in *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, Jul. 2004, pp. 74–81. [Online]. Available: <https://aclanthology.org/W04-1013>
- [40] W.-L. Chiang, L. Zheng, Y. Sheng, A. N. Angelopoulos, T. Li, D. Li, H. Zhang, B. Zhu, M. Jordan, J. E. Gonzalez et al., “Chatbot arena: An open platform for evaluating llms by human preference, 2024,” URL <https://arxiv.org/abs/2403.04132>, vol. 2, no. 9, 2024.
- [41] Y. Huang, Q. Zhang, L. Sun et al., “Trustgpt: A benchmark for trustworthy and responsible large language models,” *arXiv preprint arXiv:2306.11507*, 2023.
- [42] X. Liu, H. Yu, H. Zhang, Y. Xu, X. Lei, H. Lai, Y. Gu, H. Ding, K. Men, K. Yang et al., “Agentbench: Evaluating llms as agents,” *arXiv preprint arXiv:2308.03688*, 2023.
- [43] S. Dahdal, L. Colombi, M. Brina, A. Gilli, M. Tortonesi, M. Vignoli, and C. Stefanelli, “An MLOps framework for GAN-based fault detection in Bonfiglioli’s EVO plant,” *Infocommunications Journal*, vol. XVI, no. 2, pp. 2–10, 2024.
- [44] J. Yang, H. Jin, R. Tang, X. Han, Q. Feng, H. Jiang, S. Zhong, B. Yin, and X. Hu, “Harnessing the power of llms in practice: A survey on chatgpt and beyond,” *ACM Transactions on Knowledge Discovery from Data*, vol. 18, no. 6, pp. 1–32, 2024.
- [45] A. Mailach, S. Simon, J. Dorn, and N. Siegmund, “Practitioners’ discussions on building llm-based applications for production,” *arXiv preprint arXiv:2411.08574*, 2024.
- [46] H. Zhou, C. Hu, Y. Yuan, Y. Cui, Y. Jin, C. Chen, H. Wu, D. Yuan, L. Jiang, D. Wu et al., “Large language model (llm) for telecommunications: A comprehensive survey on principles, key techniques, and opportunities,” *arXiv preprint arXiv:2405.10825*, 2024.
- [47] Y. Xia, J. Kim, Y. Chen, H. Ye, S. Kundu, C. Hao, and N. Talati, “Understanding the performance and estimating the cost of llm finetuning,” *arXiv preprint arXiv:2408.04693*, 2024.
- [48] S. Shekhar, T. Dubey, K. Mukherjee, A. Saxena, A. Tyagi, and N. Kotla, “Towards optimizing the costs of llm usage,” *arXiv preprint arXiv:2402.01742*, 2024.
- [49] Y. Ma, S. Zheng, Z. Yang, P. Zheng, J. Leng, and J. Hong, “Leveraging large language models in next generation intelligent manufacturing: Retrospect and prospect,” *Journal of Manufacturing Systems*, vol. 82, pp. 809–840, 2025.
- [50] C. Zhang, Q. Xu, Y. Yu, G. Zhou, K. Zeng, F. Chang, and K. Ding, “A survey on potentials, pathways and challenges of large language models in new-generation intelligent manufacturing,” *Robotics and Computer-Integrated Manufacturing*, vol. 92, p. 102 883, 2025.
- [51] Y. Li, H. Zhao, H. Jiang, Y. Pan, Z. Liu, Z. Wu, P. Shu, J. Tian, T. Yang, S. Xu, Y. Lyu, P. Blenk, J. Pence, J. Rupram, E. Banu, N. Liu, L. Wang, W. Song, X. Zhai, K. Song, D. Zhu, B. Li, X. Wang, and T. Liu, “Large language models for manufacturing,” *arXiv preprint arXiv:2410.21418*, 2024.
- [52] M. Sadeqi Bajestani, D. Mun, and D. Kim, “Human-in-the-loop and large language models in smart manufacturing: Current applications, challenges, and perspectives,” *Journal of Manufacturing Systems*, vol. 86, pp. 913–941, 2026.
- [53] H. Kozirolek, S. Gruener, and V. Ashiwal, “ChatGPT for PLC/DCS control logic generation,” in *2023 IEEE 28th International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2023, pp. 1–8, arXiv:2305.15809.
- [54] M. Fakhri, R. Dharmaji, Y. Moghaddas, G. Quiros, T. Ogundare, and M. A. Al Faruque, “LLM4PLC: Harnessing large language models for verifiable programming of PLCs in industrial control systems,” in *Proceedings of the 46th International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, 2024, arXiv:2401.05443.

- [55] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, and A. Zeng, "Code as policies: Language model programs for embodied control," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 9493–9500. [Online]. Available: <https://www.semanticscholar.org/paper/91deaf9d324c8feafc189da0da03e60a60287bca>
- [56] Y. Xia, M. Shenoy, N. Jazdi, and M. Weyrich, "Towards autonomous system: flexible modular production system enhanced with large language model agents," in *2023 IEEE 28th International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2023, pp. 1–8.
- [57] J. Lim, B. Vogel-Heuser, and I. Kovalenko, "Large language model-enabled multi-agent manufacturing systems," in *2024 IEEE 20th International Conference on Automation Science and Engineering (CASE)*, 2024, pp. 3940–3946.
- [58] D. Patel, S. Lin, J. Rayfield, N. Zhou, C. Shyalika, S. R. Yarrabothula, R. Vaculin, N. Martinez, F. O'Donncha, and J. Kalagnanam, "AssetOps-Bench: A real-world evaluation benchmark for AI-driven task automation in industrial asset management," *arXiv preprint arXiv:2506.03828*, 2025.
- [59] M. Liu, T.-D. Ene, R. Kirby, C. Cheng, N. Pinckney *et al.*, "ChipNeMo: Domain-adapted LLMs for chip design," *arXiv preprint arXiv:2311.00176*, 2023.
- [60] E. Brynjolfsson, D. Li, and L. Raymond, "Generative AI at work," *The Quarterly Journal of Economics*, vol. 140, no. 2, pp. 889–942, 2025.
- [61] Siemens AG, "Siemens introduces AI agents for industrial automation," Press release, Automate 2025, Detroit, 12 May 2025. <https://www.automation.com/article/siemens-ai-agents-industrial-automation>, 2025.
- [62] B. Martini, D. Bellisario, and P. Coletti, "Human-centered and sustainable artificial intelligence in Industry 5.0: Challenges and perspectives," *Sustainability*, vol. 16, no. 13, p. 5448, 2024.
- [63] A. Frankó, G. Hollósi, D. Ficzer, and P. Varga, "Synchronized real-time communication in the Eclipse Arrowhead framework," *Infocommunications Journal*, vol. XVIII, no. 2, pp. 91–98, 2026.
- [64] "Regulation (eu) 2024/1689 of the european parliament and of the council of 13 june 2024 laying down harmonised rules on artificial intelligence (artificial intelligence act)," <https://eur-lex.europa.eu/eli/reg/2024/1689/oj/eng>, 2024, official Journal of the European Union, L 168, 12 July 2024.
- [65] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.
- [66] DeepSeek-AI, "DeepSeek-V3 technical report," *arXiv preprint arXiv:2412.19437*, 2024.
- [67] D. Guo *et al.*, "DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning," *Nature*, vol. 645, pp. 633–638, 2025.
- [68] A. Yang *et al.*, "Qwen3 technical report," *arXiv preprint arXiv:2505.09388*, 2025.
- [69] OpenAI, "gpt-oss-120b & gpt-oss-20b model card," *arXiv preprint arXiv:2508.10925*, 2025.
- [70] P. Smeros, V. Emonet, R. Wang, A.-C. Sima, and T. Mendes de Farias, "SPARQL-LLM: Real-time SPARQL query generation from natural language questions," *arXiv preprint arXiv:2512.14277*, 2025.
- [71] S. Tramp, M. Gölo, E. Marx, and P. V. do Carmo, Eds., *Proceedings of the First International TEXT2SPARQL Challenge 2025, co-located with Text2KG at ESWC 2025*, ser. CEUR Workshop Proceedings, vol. 4094. Portorož, Slovenia: CEUR-WS.org, 2025. [Online]. Available: <https://ceur-ws.org/Vol-4094/>
- [72] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019. [Online]. Available: <https://arxiv.org/abs/1810.04805>
- [73] M. Dubey, D. Banerjee, A. Abdelkawi, and J. Lehmann, "Lc-quad 2.0: A large dataset for complex question answering over wikidata and dbpedia," in *The Semantic Web-ISWC 2019: 18th International Semantic Web Conference, Auckland, New Zealand, October 26–30, 2019, Proceedings, Part II 18*. Springer, 2019, pp. 69–78.
- [74] A. Lonkhang and J. Botzheim, "Path planning transformer supervised by improved RRT* with reduced random map size for mobile robots," *Infocommunications Journal*, vol. XVIII, no. 1, pp. 100–109, 2026.
- [75] Z. Yang, L. Li, K. Lin, J. Wang, C.-C. Lin, Z. Liu, and L. Wang, "The dawn of llms: Preliminary explorations with gpt-4v(ision)," 2023. [Online]. Available: <https://arxiv.org/abs/2309.17421>
- [76] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, R. Ring, E. Rutherford, S. Cabi, T. Han, Z. Gong, S. Samangooei, M. Monteiro, J. Menick, S. Borgeaud, A. Brock *et al.*, "Flamingo: a visual language model for few-shot learning," in *Advances in Neural Information Processing Systems*, vol. 35. Curran Associates, Inc., 2022.
- [77] J. Li, D. Li, S. Savarese, and S. Hoi, "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," 2023. [Online]. Available: <https://arxiv.org/abs/2301.12597>
- [78] B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, F. Xia, J. Wu, P. Wohlhart, S. Welker, A. Wahid, Q. Vuong, V. Vanhoucke, H. Tran, R. Soricut, A. Singh, J. Singh, P. Sermanet, P. R. Sanketi, G. Salazar, M. S. Ryoo *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," in *Proceedings of The 7th Conference on Robot Learning*, ser. PMLR, vol. 229, 2023, pp. 2165–2183.
- [79] M. J. Kim *et al.*, "Openvla: An open-source vision-language-action model," in *Proceedings of The 8th Conference on Robot Learning*, ser. PMLR, vol. 270, 2025, pp. 2679–2713.
- [80] A. Béres and B. Gyires-Tóth, "Enhancing visual domain randomization with real images for sim-to-real transfer," *Infocommunications Journal*, vol. XV, no. 1, pp. 15–25, 2023.
- [81] T. Berners-Lee, "The semantic web: A new form of web content that is meaningful to computers will unleash a revolution of new possibilities," *Scientific American*, 2001. [Online]. Available: <https://api.semanticscholar.org/CorpusID:265709654>
- [82] H. Ehm, E. Schoitsch, J. W. van der Weit, N. Ramzy, L. Luo, and D. L. Gruetzner, "Digital reference: a quasi-standard for digitalization in the domain of semiconductor supply chains," *2020 IEEE Conference on Industrial Cyberphysical Systems (ICPS)*, vol. 1, pp. 563–570, 2020. [Online]. Available: <https://api.semanticscholar.org/CorpusID:228092656>
- [83] H. B. Giglou, J. D'Souza, and S. Auer, "Llms4ol: Large language models for ontology learning," *ArXiv*, vol. abs/2307.16648, 2023. [Online]. Available: <https://api.semanticscholar.org/CorpusID:260334867>
- [84] H. B. Giglou, J. D'Souza, F. Engel, and S. Auer, "Llms4om: Matching ontologies with large language models," *ArXiv*, vol. abs/2404.10317, 2024. [Online]. Available: <https://api.semanticscholar.org/CorpusID:269157258>
- [85] G. Naidu, T. Zuva, and E. M. Sibanda, "A review of evaluation metrics in machine learning algorithms," in *Artificial Intelligence Application in Networks and Systems*, ser. Lecture Notes in Networks and Systems, R. Silhavy and P. Silhavy, Eds. Cham: Springer International Publishing, 2023, vol. 724, pp. 15–25.
- [86] Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang, W. Ye, Y. Zhang, Y. Chang, P. S. Yu, Q. Yang, and X. Xie, "A survey on evaluation of large language models," *ACM Transactions on Intelligent Systems and Technology*, vol. 15, no. 3, pp. 1–45, 2024.
- [87] T. A. S. Foundation, "Zeppelin," 23/10/2024. [Online]. Available: <https://zeppelin.apache.org/>
- [88] "Eclipse arrowhead – eclipse arrowhead framework and implementation platform," 25/11/2024. [Online]. Available: <https://arrowhead.eu/>

CONTENTS

Introduction: Industrial AI for Industry 5.0

..... Pál Varga, Kārlis Freivalds, Gian Antonio Susto, and Péter Olszki 3

Machine Learning in Industry 5.0 Pál Varga, and Péter Olszki 5

Image Processing: a Review on Industrial Applications

..... Michał Affek, Manuel Barusco, Diāna Dupļevska, Kārlis Freivalds, Gergely Hollósi,
..... Corinna Kofler, Michał Kopczyński, Ruba Almahasneh, Philip Newton, Guntis V. Strazds,
..... Gian Antonio Susto, Gábor Szűcs, and Marek S. Tataara 25

Anomaly and Fault Detection and Predictive Maintenance: a Review on Industrial Applications

..... Davide Frizzo, Valentina Zaccaria, David Petrovic, Francesco Borsatti, Davide Sartor,
..... Felice Russo, Genoveffa Marcario, and Gian Antonio Susto 44

Reinforcement Learning, Optimization and Scheduling in Industry 5.0: Overview and Case studies

..... Alberto Sinigaglia, Riccardo De Monte, Patrick Stöckermann,
..... Alessandro Immordino, Ilayda Celenk, Willem van Jaarsveld, Ivo Adan, Eric Weijers,
..... Cas Rosman, Bibi de Jong, Kai Schelthoff, Bram Hoekstra, Laura Leja, Gian Antonio Susto,
..... Oskars Teikmanis, and Karlis Freivalds 104

Robotics, Automation, and Process Management for Industrial Operations

..... Fabian Lindner, Iraklis Varlamis, Katrīna Viltrake, Karl-Benedikt Reith, Sebastian Rank,
..... Michał Affek, Germar Schneider, Rosa Leschly, Venugopal Ramadasu, Gerald Reiner,
..... Pascal Herfort, Guntis V. Strazds, and Marek Tataara 119

Large Language Models in Industry 5.0

– Multi-Case Experiences from the AIMS5.0 Project

..... Csaba Hegedűs, Péter Olszki, Máté Bancsics, Dániel Ficzer, Alex Latipov,
..... Mike Holenderski, Guntis V. Strazds, Belal Abu Naim, Yasin Ghafourian, Markus Tauber,
..... Marta Bonik, Chris Damkat, David Dandolo, Chiara Masiero, Manuel Barusco,
..... Gian Antonio Susto, Ruba Almahasneh, and Pál Varga 135

Contact information

President: **FERENC BANCICS** • elnok@hte.hu
Secretary-General: **GÁBOR KOLLÁTH** • kollath.gabor@hte.hu
Operations Director: **PÉTER NAGY** • nagy.peter@hte.hu

Indexing information

Infocommunications Journal is covered by Inspec, Compendex and Scopus.
**Infocommunications Journal is also included in the Thomson Reuters – Web of Science™ Core Collection,
Emerging Sources Citation Index (ESCI)**

Infocommunications Journal

Technically co-sponsored by IEEE Communications Society and IEEE Hungary Section

Supporters

FERENC BANCICS – president, Scientific Association for Infocommunications (HTE)

MTA200

HTE Infocommunications Journal welcomes
the 200th anniversary of the Hungarian Academy of Sciences
and gratefully acknowledges its continuous support
of the journal in recent years.



National Media and Infocommunications Authority | Hungary

The publication was also produced with the support of the NMHH.

Editorial Office (Subscription and Advertisements):

Scientific Association for Infocommunications
H-1051 Budapest, Bajcsy-Zsilinszky str. 12, Room: 502
Phone: +36 1 353 1027 • E-mail: info@hte.hu • Web: www.hte.hu

Articles can be sent also to the following address:

Budapest University of Technology and Economics
Department of Telecommunications and Media Informatics
Phone: +36 1 463 4189 • E-mail: pvarga@tmit.bme.hu

Publisher: PÉTER NAGY

HU ISSN 2061-2079 • Layout: PLAZMA DS • Printed by: FOM Media

www.infocommunications.hu

2026

SPECIAL ISSUE OF THE INFOCOMMUNICATION JOURNAL

