

Review

State-of-the-Art on Digital Twin Technologies for Industrial Applications and the Federated Digital Twin Lifecycle Model (F-DTLM)

Janis Peksa ^{1,2,*} and Dmytro Mamchur ^{1,3}

¹ Information Technology Faculty, Turība University, Graudu Street 68, LV-1058 Riga, Latvia; dgmamchur@gmail.com

² Institute of Information Technology, Riga Technical University, Kalku Street 1, LV-1658 Riga, Latvia

³ Computer Engineering and Electronics Department, Kremenchuk Mykhailo Ostrohradskyi National University, Universitetska Street 20, 39600 Kremenchuk, Ukraine

* Correspondence: janis.peksa@turiba.lv

Abstract

Digital Twins (DTs) have emerged as a key technology for sensor-driven cyber–physical systems, enabling such features as real-time monitoring, predictive maintenance, and operational optimization. Despite rapid progress, existing research in the area remains fragmented, mostly addressing only singular aspects, such as data acquisition, modeling, or control, lacking a unified lifecycle-oriented methodology capable of integrating heterogeneous sensor infrastructures, hybrid analytical models, and continuous feedback mechanisms. This paper presents a comprehensive state-of-the-art review of Digital Twin technologies, focusing on sensor-centric architectures, data integration strategies, and hybrid modeling approaches. Based on the identified limitations, a novel Federated Digital Twin Lifecycle Model (F-DTLM) is proposed as a unifying framework for industrial applications. The model structures the DT lifecycle into four iterative phases—Definition and Scoping; Sensor Data and Infrastructure Federation; Hybrid Modeling and State Synchronization; and Operational Optimization and Closed-Loop Control, supported by cross-cutting layers addressing interoperability and governance. The integration of federated sensing infrastructures with hybrid physics-informed and data-driven models enables scalable synchronization between physical and digital systems. A comparative analysis and an illustrative predictive maintenance scenario illustrate the potential applicability of the proposed approach.

Keywords: digital twin; sensor networks; cyber–physical systems; predictive maintenance; hybrid modeling



Academic Editor: Giacomo Innocenti

Received: 30 March 2026

Revised: 6 May 2026

Accepted: 15 May 2026

Published: 17 May 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

The paradigm of the Digital Twin (DT) has emerged as a fundamental pillar of modern cyber-physical systems (CPS), driving the transition toward Industry 4.0 and the human-centric principles of Industry 5.0. Conceptually, a Digital Twin is defined as an exhaustive, real-time, synchronized virtual representation of a physical asset, process, or system. The historical trajectory of this technology dates back to the aerospace sector, specifically to the physical “mirrored systems” utilized by the National Aeronautics and Space Administration (NASA, USA) during the Apollo missions in order to simulate and mitigate the

possibility of catastrophic failures [1]. The conceptual formalization of the digital counterpart was later introduced in 2002 within the context of Product Lifecycle Management (PLM), defining a triadic structure comprising a physical entity, a virtual entity, and the bidirectional data linkages connecting them altogether [2].

In contemporary engineering, the taxonomy of digital representation is strictly delineated by the automation of data flow. A “Digital Model” involves manual data exchange, while a “Digital Shadow” relies on automated, unidirectional telemetry from the physical to the virtual environment [3–6]. The true “Digital Twin” is characterized by a fully automated, bidirectional closed-loop system where the virtual model not only mirrors the physical state but is also able to prescribe actions and autonomously control its physical counterpart.

The realization of high-fidelity DTs is entirely contingent upon a convergence of enabling technologies. The Industrial Internet of Things (IIoT) and advanced sensor networks serve as the nervous system, capturing continuous multimodal telemetry [5,7–9]. Cloud and edge computing architectures provide the necessary computational elasticity to process vast datasets, while Artificial Intelligence (AI) acts as the cognitive core, shifting the utility of the models from purely descriptive simulations to prescriptive, autonomous decision-making engines [9,10]. In the context of modern CPS, the implementation of DT technology is no longer merely an optimization tool; it is an essential requirement for managing the unpredictable, emergent behaviors of highly complex, interconnected systems operating under deep uncertainty.

To support the state-of-the-art synthesis, this study applies a structured review protocol described in Section 2, including database selection, search strings, inclusion and exclusion criteria, screening logic, data extraction categories, and evidence-level classification. Literature analysis revealed that despite rapid DT adoption, there is a lack of consensus on lifecycle methodologies governing Digital Twin development and operation. Many contributions focus on isolated aspects, such as modeling, data acquisition, or visualization, without providing an integrated framework that spans the full lifecycle of a Digital Twin. Furthermore, existing approaches often neglect the explicit integration of hybrid physics–data-driven models, federated data architectures, and closed-loop control mechanisms.

To address these limitations, this paper provides both a structured survey of Digital Twin technologies and a methodological contribution in the form of a Federated Digital Twin Lifecycle Model (F-DTLM). The proposed framework aligns with emerging industrial standards while remaining domain-agnostic and methodologically reproducible.

The main contributions of this paper are:

- A unified workflow of DT data processing;
- A taxonomy of DT approaches;
- A mathematical foundation for DT modeling paradigms;
- A technical architecture aligned with industry standards;
- The proposed Federated Digital Twin Lifecycle Model (F-DTLM);
- An illustrative application scenario.

2. Review Methodology

2.1. Review Objective and Scope

The objective of this review is to provide a comprehensive, up-to-date synthesis of Digital Twin (DT) methodologies. We are specifically focusing on their application within industrial automation and cyber-physical systems. Along the way, we pay special attention to lifecycle-oriented frameworks, hybrid modeling techniques, federated data infrastructures, interoperability standards, and closed-loop operational capabilities. Ultimately, we designed this review to pinpoint existing methodological gaps and to establish the clear

requirements that motivate our proposed framework: the Federated Digital Twin Lifecycle Model (F-DTLM).

Rather than attempting a broad overview of every DT application across all fields, we intentionally narrowed our scope to methodologies that offer real, transferable value to industrial environments. To be included, studies had to address at least one of the following key areas:

- Lifecycle design;
- Industrial implementation;
- Hybrid physics- and data-driven modeling;
- Interoperability standards;
- Edge-cloud or federated infrastructures;
- Semantic integration;
- Governance;
- Predictive maintenance;
- Closed-loop control.

To directly connect our literature synthesis with the methodological contributions of this paper, we guided our review using five core research questions:

- RQ1. How do existing DT frameworks explicitly define lifecycle phases and methodological steps for industrial and cyber-physical systems?
- RQ2. How are physics-based, data-driven, and hybrid modeling methods integrated within current approaches?
- RQ3. To what extent do these frameworks support federated data infrastructures, semantic interoperability, and alignment with industry standards?
- RQ4. How are closed-loop processes—such as monitoring, prediction, optimization, and control—represented in current DT methodologies?
- RQ5. What methodological gaps remain in current lifecycle approaches, and how can this information be employed in the development of our F-DTLM?

To ensure our reporting remains rigorous and transparent, the reporting structure was guided by PRISMA 2020 and PRISMA-ScR principles for transparent identification, screening, and synthesis.

2.2. Search Strategy and Information Sources

We conducted a structured literature search across major scientific databases and publisher platforms, focusing heavily on computer science, electrical engineering, industrial automation, and manufacturing. Our primary sources included IEEE Xplore, Scopus, Web of Science Core Collection, ScienceDirect, SpringerLink, ACM Digital Library, and MDPI.

We limited our primary search to publications spanning from January 2015 to April 2026. We chose 2015 as our starting point because the rapid surge in DT research closely aligns with the maturation of Industry 4.0, the Industrial Internet of Things (IIoT), and cloud/edge computing that began around that time. However, we also manually included older, foundational texts when necessary to provide historical or conceptual completeness, particularly for core definitions of the Digital Model, Digital Shadow, and Digital Twin.

We tailored our baseline search string to the specific syntax of each database, generally following this general query structure:

“digital twin” OR “digital twins”) AND (“lifecycle” OR “life cycle” OR “framework” OR “methodology” OR “architecture” OR “reference architecture”) AND (“industrial automation” OR “manufacturing” OR “cyber-physical system” OR “predictive maintenance” OR “Industry 4.0”)

To ensure we captured the specific nuances of our review dimensions, we also deployed a set of targeted queries:

- For Federated Systems: (“digital twin” AND “federated” AND (“data” OR “architecture” OR “learning”))
- For Modeling Paradigms: (“digital twin” AND (“physics-informed” OR “hybrid model” OR “grey-box” OR “physics-based” OR “data-driven”))
- For Standards [11–13]: (“digital twin” AND (“RAMI 4.0” OR “ISO 23247” OR “Asset Administration Shell” OR “AAS”))
- For Closed-Loop Operations: (“digital twin” AND (“closed-loop” OR “control” OR “optimization” OR “prescriptive”))

Finally, we supplemented our database searches with a “snowballing” technique—carefully checking both the backward and forward references of highly relevant review papers and standardization studies. This crucial extra step ensured we did not miss any foundational definitions, reference architectures, or major lifecycle-oriented contributions.

Adopted search strategy summarized in Table 1.

Table 1. Search strategy used for the structured literature review.

Search Block	Main Terms	Purpose
Core Digital Twin (DT) concept	“digital twin”, “digital twins”	Identification of DT-related publications
Lifecycle and methodology	“lifecycle”, “life cycle”, “framework”, “methodology”, “reference architecture”	Identification of lifecycle-oriented and methodological studies
Industrial context	“industrial automation”, “manufacturing”, “cyber-physical system”, “Industry 4.0”	Restriction to industrial and CPS-relevant studies
Modeling paradigm	“physics-based”, “data-driven”, “hybrid model”, “grey-box”, “physics-informed”	Identification of modeling approaches
Federation and interoperability	“federated”, “AAS”, “RAMI 4.0”, “ISO 23247”, “OPC UA”, “MQTT”	Identification of integration and interoperability studies
Operational use	“predictive maintenance”, “RUL”, “closed-loop”, “optimization”, “control”	Identification of applied and operational DT studies

2.3. Inclusion and Exclusion Criteria

Before starting a screening process, we have defined exclusion and inclusion criteria to ensure consistency and reduce selection bias (Table 2). A publication was included if it met at least one of the inclusion criteria and did not meet any exclusion criteria.

Table 2. Inclusion and exclusion criteria.

Type	Criterion
Inclusion	The study proposes, analyzes, or compares a Digital Twin lifecycle model, framework, methodology, or reference architecture.

Table 2. *Cont.*

Type	Criterion
Inclusion	The study addresses Digital Twins in industrial automation, manufacturing, cyber–physical systems, predictive maintenance, or closely related engineering contexts.
Inclusion	The study discusses physics-based, data-driven, or hybrid DT modeling, including Physics-Informed Neural Networks, grey-box models, or uncertainty-aware synchronization.
Inclusion	The study addresses DT interoperability, standards, or integration technologies such as ISO 23247, RAMI 4.0, AAS, OPC UA, MQTT, edge computing, cloud computing, or federated architectures.
Inclusion	The study provides conceptual, simulation-based, experimental, or industrial evidence relevant to DT deployment or lifecycle operation.
Exclusion	The term “Digital Twin” is used only superficially, without methodological, architectural, or modeling relevance.
Exclusion	The paper focuses only on visualization, CAD modeling, or static simulation without synchronization, lifecycle, or operational relevance.
Exclusion	The study is purely domain-specific and does not provide transferable methodological or architectural insight.
Exclusion	The publication is a short abstract, editorial, presentation-only item, or non-peer-reviewed document, except for standards used as technical references.
Exclusion	The full text is unavailable or the paper is not written in English.
Exclusion	Duplicate records across databases.

2.4. Screening and Selection Procedure

We designed our screening process around a PRISMA-inspired logic, carefully adapting it to suit the specific needs of a structured engineering review. Our approach followed a systematic, five-step pipeline:

1. Identification. We gathered initial records from our selected databases and publisher platforms;
2. Deduplication. We purged any duplicate entries to create a clean working dataset;
3. Initial Screening. We reviewed the titles and abstracts of the remaining records against our strict inclusion and exclusion criteria;
4. Full-Text Assessment. For the papers that passed the initial filter, we conducted a thorough full-text evaluation to confirm their eligibility;
5. Final Inclusion. The studies that met all of our criteria were ultimately included in our qualitative synthesis.

To visualize this process, we have outlined the exact breakdown of our screening phases below (Figure 1).

2.5. Data Extraction and Classification Framework

Once the final set of 52 studies was established, we extracted structured data from each paper using a predefined classification framework. Our intention with this step was to move well beyond a simple, descriptive overview of the literature. Instead, we wanted to build a robust foundation that allowed for a rigorous, analytical comparison of how these existing approaches actually function and compare against one another. Adopted data extraction framework summarized in Table 3.

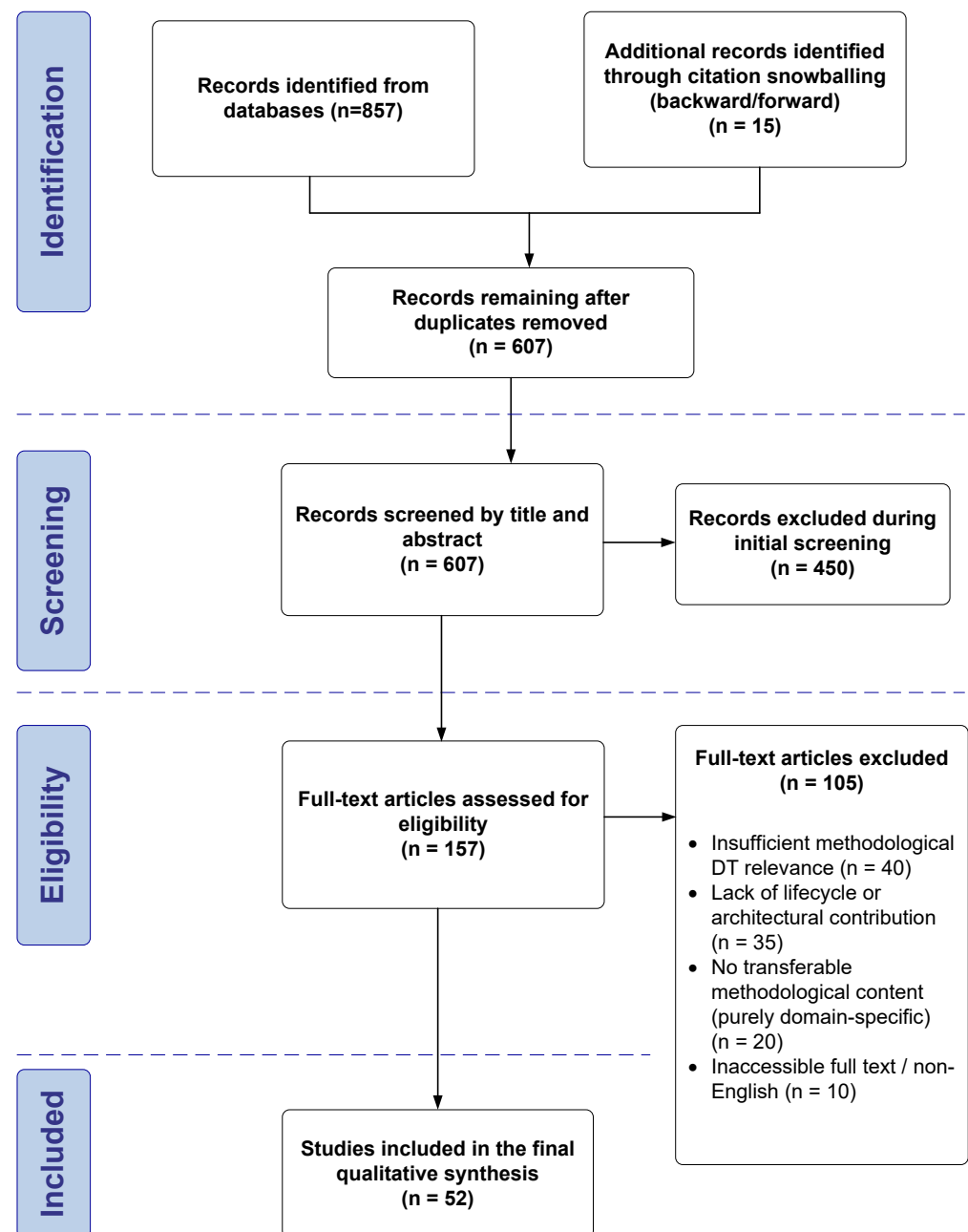


Figure 1. PRISMA-style flow diagram of the literature identification, screening, eligibility assessment and inclusion process.

Table 3. Data extraction framework used for the state-of-the-art synthesis.

Extraction Category	Description
Bibliographic information	Authors, year, publication type, journal/conference, publisher
DT representation type	Digital Model, Digital Shadow, Digital Twin, or unclear
Application domain	Manufacturing, energy, predictive maintenance, healthcare, smart city, environmental systems, or cross-domain
Lifecycle coverage	Scoping, data acquisition, integration, modeling, synchronization, prediction, optimization, control, refinement

Table 3. *Cont.*

Extraction Category	Description
Modeling paradigm	Physics-based, data-driven, hybrid, probabilistic, or unspecified
Data architecture	Centralized, edge-based, cloud-based, edge–cloud hybrid, federated, semantic
Standards and Interoperability	ISO 23247, RAMI 4.0, AAS, OPC UA, MQTT, other standards, or none
Feedback capacity	No feedback, human-in-the-loop feedback, semi-automated feedback, fully automated closed-loop control
Governance and trust	Cybersecurity, access control, model validation, versioning, auditability, regulatory compliance
Validation level	Conceptual, simulation, laboratory prototype, industrial deployment, review/synthesis
Main limitation	Scalability, data dependency, interoperability, explainability, computational cost, validation gap, governance gap
Relevance to F-DTLM	Which methodological requirement or gap the study supports

This extraction framework was used to classify the reviewed literature according to methodological contribution.

2.6. Evidence-Level Classification

To accurately assess the actual maturity of current DT contributions, we categorized the included studies based on the level of evidence backing their claims. We introduced this classification metric (Table 4) because much of the existing DT literature proposes architectures or conceptual frameworks without providing empirical or industrial validation.

Table 4. Evidence-level classification of reviewed Digital Twin studies.

Evidence Level	Category	Description
L1	Conceptual	Definitions, conceptual frameworks, taxonomies, or theoretical models without implementation
L2	Architectural	Reference architecture, software stack, or system design without experimental validation
L3	Simulation-based	Validation through simulation or virtual experiments
L4	Laboratory prototype	Controlled experimental implementation or testbench validation
L5	Industrial deployment	Real-world industrial case study or deployment with operational data
L6	Review/Synthesis	Systematic review, structured review, meta-synthesis, or survey

This evidence-level classification allows our review to draw a clear line between conceptual completeness and empirical maturity. After all, a framework might be conceptually comprehensive but entirely lack real-world industrial testing. Acknowledging this distinction helps us transparently position our own F-DTLM as the model we are proposing as a methodological framework synthesized from the literature and illustrated through

an application scenario, with the understanding that full industrial validation remains an important direction for future work.

2.7. Synthesis Method

We synthesized the selected studies using a thematic and comparative approach, focusing our analysis on five key dimensions that map directly back to our initial research questions:

1. Lifecycle structure. Does the study explicitly define comprehensive DT lifecycle phases, or does it merely discuss isolated components?
2. Modeling approach. Does the DT rely on physics-based, data-driven, or hybrid modeling techniques?
3. Data and interoperability architecture. Does the framework utilize centralized, edge/cloud, federated, or semantic data integration?
4. Operational feedback. Does the DT support descriptive monitoring, predictive analytics, prescriptive optimization, or full closed-loop control?
5. Governance and trust. Does the study tackle critical issues like cybersecurity, explainability, model validation, versioning, access control, or regulatory compliance?

We carried out this synthesis in two distinct stages. First, we grouped the literature into broad thematic clusters: DT definitions and maturity levels, lifecycle methodologies, modeling paradigms, technical architectures, federated and semantic integration, governance and security, and predictive maintenance applications. Second, we systematically compared these existing approaches against the specific methodological requirements necessary for a lifecycle-oriented industrial DT framework. We then used the resulting gap analysis to directly inform and derive the design requirements for the F-DTLM.

2.8. Derivation of Methodological Requirements

Our synthesis of the literature resulted in six core methodological requirements necessary for building a lifecycle-oriented DT framework suited for industrial automation and cyber-physical systems, summarized in Table 5.

Table 5. Methodological requirements derived from the state-of-the-art review.

Identified Limitation in Existing DT Literature	Derived Methodological Requirement	F-DTLM Response
Many DT studies address only isolated stages such as modeling, monitoring, or visualization.	A DT methodology should define explicit lifecycle phases from scoping to operation and refinement.	Four-phase lifecycle: Definition and Scoping; Data and Infrastructure Federation; Hybrid Modeling and Synchronization; Operational Optimization and Control.
Data integration is often centralized, platform-specific, or weakly described.	A DT methodology should support distributed and federated data infrastructures.	Data and Infrastructure Federation phase.
Physics-based and data-driven models are often treated separately.	A DT methodology should support hybrid modeling and model fusion.	Hybrid Modeling and Synchronization phase.
Standards are frequently discussed but not operationalized across the lifecycle.	A DT methodology should align with standards and semantic interoperability mechanisms.	Semantic Interoperability cross-cutting layer.
Governance, cybersecurity, and model validation are often external to the lifecycle.	Trust mechanisms should be embedded across all lifecycle stages.	Governance and Trust cross-cutting layer.
Many implementations remain digital shadows rather than closed-loop twins.	A DT methodology should explicitly include operational optimization and feedback control.	Operational Optimization and Control phase.

This requirement-driven synthesis forms the conceptual bedrock for the F-DTLM framework that we will formally introduce in Section 6.

2.9. Limitations of the Review Methodology

While we structured and conducted this review using transparent, rigorous criteria, we must acknowledge a few inherent limitations.

First, because our focus was strictly on lifecycle-oriented DT methodologies relevant to industrial automation and cyber-physical systems, we only included highly specialized, domain-specific studies if they offered broadly transferable methodological insights. Second, we purposefully emphasized peer-reviewed literature and technical standards; as a result, we did not systematically analyze industrial white papers or vendor documentation. Third, DT terminology is still not fully standardized across the field. Some relevant studies might use alternative phrasing, such as “cyber-physical model,” “virtual commissioning,” “asset model,” or “digital shadow.” To mitigate this risk, we relied heavily on backward and forward citation checking to catch relevant papers that our database keyword searches might have missed. Finally, it is important to note that this review provides a structured qualitative synthesis rather than a statistical meta-analysis. Given that the reviewed studies vary wildly in their core objectives, methods, application domains, and levels of validation, direct quantitative aggregation would be unsuitable and ultimately unhelpful.

3. State-of-the-Art Synthesis of Digital Twin Technologies

3.1. Evolution and Terminological Boundaries of Digital Twin Concepts

The DT concept has evolved over the years. What began as simple physical mirroring and engineering simulation has transformed into continuously synchronized cyber-physical systems that can actively monitor, predict, optimize, and provide operational feedback. Historically, the idea of using a counterpart system to help make decisions in inaccessible or high-risk environments is strongly tied to aerospace engineering, most notably NASA’s use of mirrored systems for mission support [1,14,15]. Later, as the DT concept was formalized within Product Lifecycle Management (PLM), the focus shifted to the dynamic relationship between a physical entity, its virtual representation, and the critical data link connecting the two [16,17]. With the explosive growth of the Industrial Internet of Things (IIoT), cloud and edge computing, and AI, the DT evolved from a static, design-stage model into a dynamic, operational cyber-physical technology used for real-time industrial monitoring and optimization [15–24].

For the purposes of this state-of-the-art review, however, this historical evolution is primarily relevant because it highlights the crucial shift from static representations to fully synchronized, feedback-capable systems. We are not treating this history as our central contribution; rather, we are using it as a conceptual baseline to distinguish between different maturity levels of digital representation. This distinction matters because many systems loosely labeled as “Digital Twins” in the literature are, strictly speaking, just Digital Models or Digital Shadows. Figure 2 summarizes DT concept evolution throughout the years.

As illustrated in Figure 3, we can classify digital representations into three principal forms based on their level of automation and the direction of their data exchange [3,4,6,25]. A Digital Model is a static or manually updated virtual representation of a physical object. For example, a CAD model of an induction motor used during the design phase is a Digital Model if changes in the physical asset do not automatically update the virtual representation. A Digital Shadow introduces automated one-way data flow from the physical asset to the digital representation. A typical industrial example is a SCADA or dashboard system that displays motor current, vibration, or temperature in real time but does not send optimized control commands back to the motor drive. A Digital Twin, in

the strict sense, requires bidirectional data exchange: sensor data update the virtual model, while the model supports recommendations, parameter adjustment, maintenance decisions, or automated control actions [3–6,25].

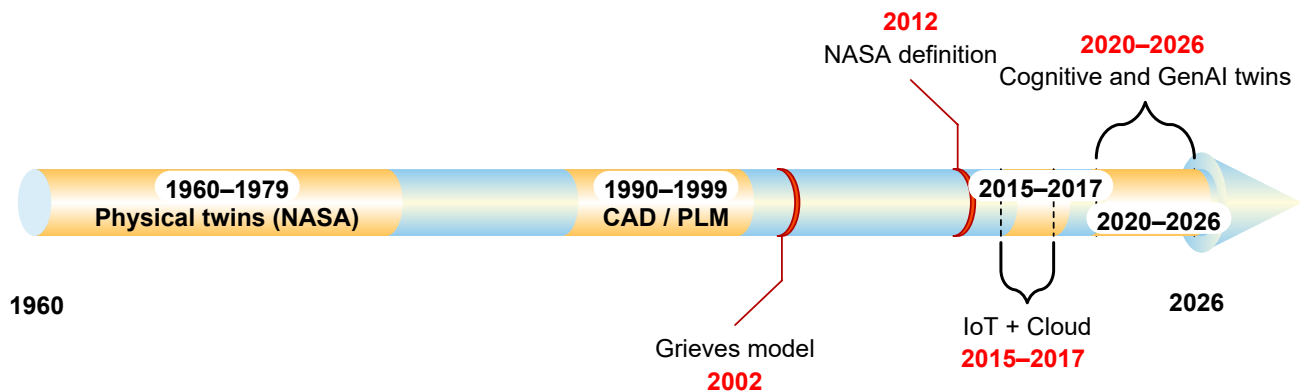


Figure 2. Historical evolution of Digital Twin concepts from physical redundancy in aerospace engineering to modern cognitive and AI-enabled Digital Twins.

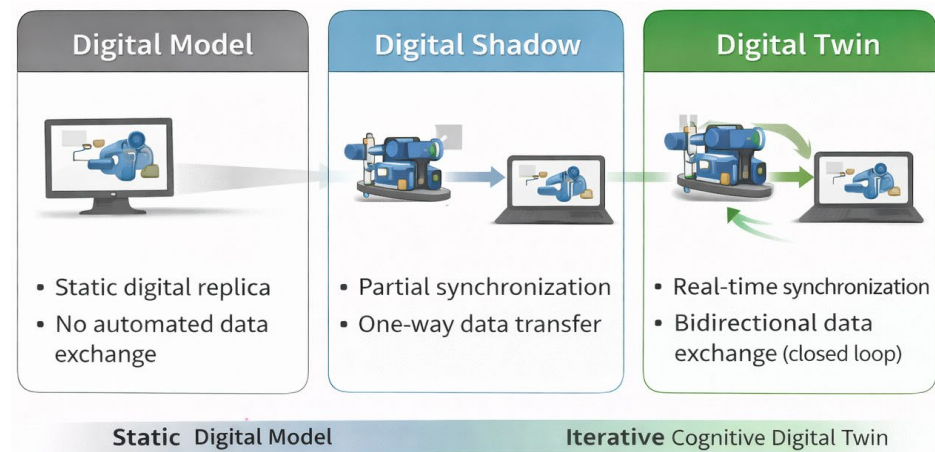


Figure 3. Taxonomy of digital representations.

This taxonomy carries real methodological weight. Digital Models are built for design and simulation; Digital Shadows excel at monitoring and diagnostics; true Digital Twins are engineered for prediction, optimization, and feedback. Therefore, DT maturity should not be evaluated solely by how complex its virtual model is. Instead, we must evaluate its degree of synchronization, decision integration, and closed-loop capabilities. This core idea is the driving force behind our Federated Digital Twin Lifecycle Model (F-DTLM), because a robust lifecycle methodology becomes necessary the moment a DT moves past simple visualization and starts making operational decisions.

3.2. Digital Twin Architectures as System-of-Systems

It is vital to understand that a Digital Twin is never just a single software module or isolated simulation model. Instead, it is a complex “system-of-systems” that weaves together physical assets, data acquisition infrastructure, communication protocols, virtual models, analytics engines, semantic representations, and feedback mechanisms [26,27]. Based on the provided literature analysis, we can identify five major architectural dimensions: the physical system, the data acquisition and communication layer, the virtual model, the analytics engine, and the feedback-control mechanism.

The physical system corresponds to the real-world asset, process, or infrastructure being represented. In ISO 23247 terminology, this often corresponds to the Observable Man-

ufacturing Element (OME), ranging from machines and tools to sensors and production resources [28]. The data acquisition layer gathers a wide mix of operational telemetry—like vibration, current, voltage, temperature, pressure, speed, energy consumption, and environmental variables—from sensors, PLCs, SCADA systems, industrial controllers, embedded devices, and historical logs. The communication layer enables synchronization between physical and digital entities and is typically implemented using industrial protocols and middleware such as OPC UA, MQTT, 5G/6G communication, REST APIs, and edge-to-cloud streaming mechanisms [29]. The virtual model layer contains the digital representation of the physical asset. Importantly, this model is not necessarily a single mathematical object; it may combine geometric models, physics-based equations, data-driven predictors, rule-based logic, semantic asset descriptions, and reduced-order or surrogate models. The analytics layer performs data fusion, state estimation, model updating, anomaly detection, prediction, optimization, and decision support. Bayesian updating, Kalman filtering, probabilistic inference, and dynamic state estimation are commonly used to align virtual states with incoming telemetry [30,31]. Finally, the feedback-control layer transforms analytical outputs into operator recommendations, maintenance actions, optimized set-points, or control commands.

3.3. Data Processing Workflow and Closed-Loop Operation

At its core, the DT paradigm runs on a structured data processing workflow. This pipeline takes raw physical telemetry and turns it into operational intelligence—and, in the most advanced applications, into prescriptive or autonomous action. This workflow unfolds across a sequence of five distinct stages represented in Figure 4 [32–34].

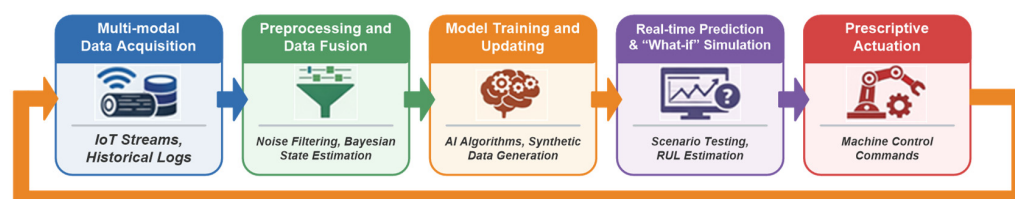


Figure 4. Workflow of Digital Twin data processing.

As shown in Figure 4, the workflow begins with multi-modal data acquisition, where measurements are collected from IoT sensors, industrial control systems, embedded devices, machine logs, and historical databases. These streams typically include physical measurements such as vibration, temperature, pressure, current, voltage, rotational speed, energy consumption, and environmental conditions. The literature consistently shows that DT quality depends strongly on the reliability, temporal resolution, and semantic consistency of the data layer [32,35].

The second stage, preprocessing and data fusion, transforms raw heterogeneous data into an analytically meaningful representation. This stage includes signal filtering, time alignment, outlier detection, missing-data handling, normalization, feature extraction, and multi-sensor fusion. Because industrial DTs often combine measurements with different sampling rates, noise levels, and uncertainty characteristics, preprocessing is not merely a technical preparation step. It directly affects the validity of downstream modeling, prediction, and control. Probabilistic inference and Bayesian state estimation are especially relevant when the physical asset is only partially observable or when sensor data are noisy [34,36,37].

The third stage, model training and updating, forms the computational core of the DT. Processed data are used to calibrate physics-based models, train data-driven predictors, update hybrid models, or adjust semantic representations. A defining property of DTs is

that model updating is continuous or recurrent rather than one-time. This is necessary because physical systems age, operating regimes change, sensors drift, and degradation mechanisms evolve over time [36–38].

The fourth stage, real-time prediction and what-if simulation, enables the DT to move beyond monitoring. Typical functions include anomaly detection, fault diagnosis, remaining useful life (RUL) estimation, operational forecasting, and virtual scenario testing. What-if simulation allows candidate actions to be evaluated *in silico* before they are applied to the physical system, reducing operational risk and supporting proactive decision-making [33,37,39].

The final stage, prescriptive actuation, closes the cyber–physical loop by translating DT insights into decisions, recommendations, or machine-readable control commands. Depending on the application and risk level, the feedback may be fully automated, semi-automated, human-approved, or purely advisory. In high-risk industrial systems, human-in-the-loop and human-on-the-loop configurations are often more realistic than full autonomy. Advanced DTs may integrate model predictive control, reinforcement learning, or optimization algorithms to support operational adaptation [39,40].

3.4. Taxonomy of Digital Twin Approaches

The conceptual landscape of Digital Twin research has evolved into a multi-dimensional taxonomy encompassing levels of system integration, modeling paradigms, and application domains. As illustrated in Figure 5, this taxonomy provides a structured framework for understanding the diversity of DT implementations and their methodological foundations. Such classification is widely adopted in the literature to clarify the distinctions between different DT maturity levels and to position emerging approaches within a coherent scientific context [32–34].

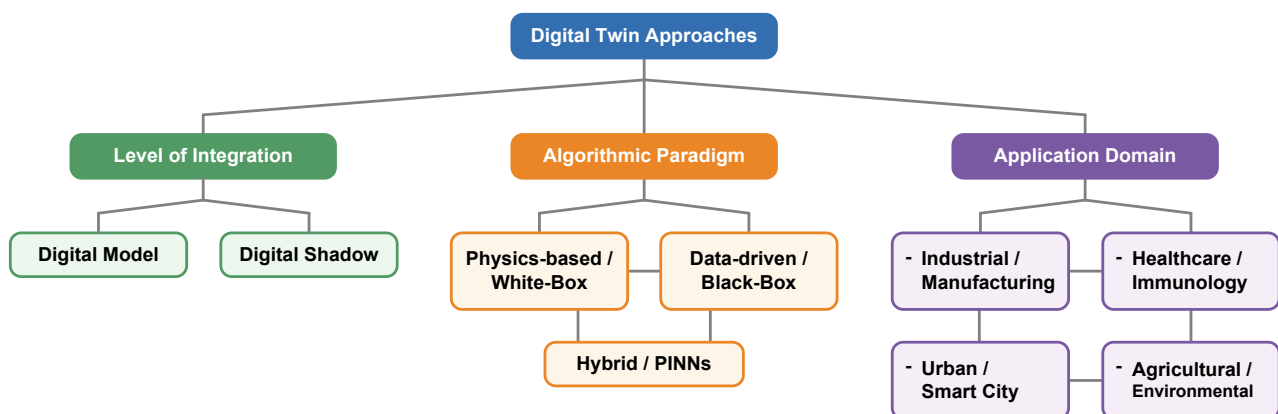


Figure 5. Taxonomy of Digital Twin approaches.

Breaking down this taxonomy further, the first dimension we must consider is the level of integration. This maps directly back to the Digital Model, Digital Shadow, and Digital Twin distinctions we defined earlier. Importantly, this is not just an argument over semantics. The integration level fundamentally dictates the system’s data architecture, its synchronization requirements, and its overall operational maturity. For instance, a real-time monitoring dashboard might be a valuable engineering tool, but if it only absorbs incoming data and cannot push operational feedback back to the physical system, we must accurately classify it as a Digital Shadow rather than a fully fledged Digital Twin [32,41].

Our second dimension is the algorithmic paradigm. When building a DT, the modeling approach generally falls into one of three categories: physics-based, data-driven, or hybrid. Physics-based models are grounded in explicit scientific laws, relying on the principles of

mechanics, thermodynamics, electromagnetics, or fluid dynamics. On the other end of the spectrum, data-driven models lean heavily on machine learning, statistical analysis, and AI to uncover hidden relationships within historical and streaming data. Hybrid models offer a powerful middle ground, blending both approaches. For example, a hybrid system might use physics-based rules to establish structural constraints and boundaries, while deploying machine-learning algorithms as surrogate models or to correct residual errors [37,42–45].

The third dimension is the application domain. In industrial and manufacturing settings, DTs are widely used tools for predictive maintenance, process optimization, production planning, and lifecycle management. Yet, their reach extends far beyond the factory floor. Healthcare DTs are paving the way for patient-specific modeling and personalized treatment planning. Urban and smart-city DTs weave together complex data across infrastructure, mobility, environment, and energy networks. Meanwhile, agricultural and environmental DTs excel at monitoring, forecasting, and optimizing scarce resources [33,34,46]. However, this sheer diversity of domains introduces a methodological trade-off. If a DT lifecycle model is too rigidly tied to one specific domain, it becomes incredibly difficult to transfer to another. Conversely, if a framework is overly abstract, it loses its practical implementation value entirely.

Ultimately, the main takeaway from this taxonomy is that DT design decisions are deeply interdependent. You simply cannot isolate them. Pushing for higher levels of integration naturally demands much stronger synchronization, interoperability, and governance mechanisms. Opting for a hybrid modeling approach only makes sense, and only works, if you possess both robust physical knowledge and rich operational data. Furthermore, achieving true closed-loop control requires more than just predictive accuracy; it demands strict safety constraints, decision validation, and airtight, trusted communication channels.

This interconnected reality strongly reinforces the core rationale behind our F-DTLM: a truly effective industrial DT methodology must seamlessly coordinate the integration level, modeling paradigm, data architecture, governance, and operational feedback, rather than treating them as a disconnected checklist of independent design choices.

Thus, the taxonomy depicted in Figure 5 provides a comprehensive framework for categorizing Digital Twin approaches along three key dimensions: integration level, algorithmic paradigm, and application domain. By organizing the diverse body of DT research into a structured hierarchy, this taxonomy facilitates a deeper understanding of methodological trade-offs and supports the systematic design and evaluation of DT systems. It also highlights the ongoing evolution of the field toward more integrated, hybrid, and application-specific solutions that bridge the gap between physical modeling and data-driven intelligence.

3.5. Modeling Paradigms: Physics-Based, Data-Driven, and Hybrid Digital Twins

Figure 6 presents a comparative analysis of two dominant modeling paradigms in DT systems: physics-based (white-box) and AI-based (data-driven, black-box) approaches. Analysis has been provided across five critical evaluation dimensions: geometric/physical fidelity, real-time inference speed, interpretability, scalability, and computational cost. This comparison reflects a central theme in DT research: the trade-off between model transparency and physical consistency on the one hand, and computational efficiency and adaptability on the other. The literature consistently emphasizes that no single paradigm is universally optimal; instead, the choice depends on application requirements, system complexity, and operational constraints [32–34].

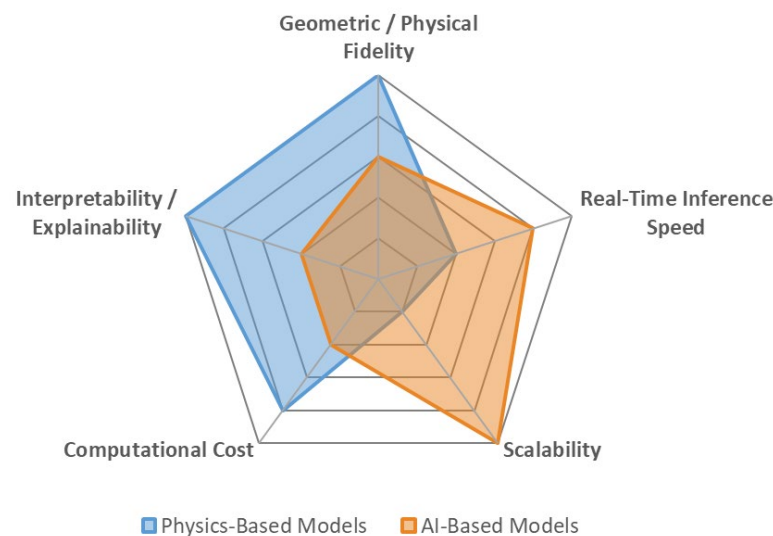


Figure 6. Comparison of AI-based and physics-based Digital Twins.

The first dimension, geometric and physical fidelity, is one of the strongest points of physics-based models. These models are derived from first principles, such as conservation laws and constitutive equations, enabling them to capture system behavior with high accuracy and physical consistency. Techniques such as FEM and CFD are capable of modeling complex spatial and temporal dynamics with fine granularity. As a result, physics-based DTs are particularly suitable for safety-critical systems and scenarios requiring extrapolation beyond observed data. Studies highlight that these models maintain validity even under previously unseen operating conditions, provided the governing equations remain applicable [32,42]. In contrast, AI-based models approximate system behavior through statistical learning, which may limit their ability to generalize outside the training data distribution.

The second dimension, real-time inference speed, represents a key advantage of AI-based approaches. Once trained, machine learning models, such as neural networks, can perform inference with minimal computational overhead, enabling rapid predictions suitable for real-time monitoring and control. This is particularly important in industrial environments where low-latency decision-making is required. In contrast, physics-based models often involve solving complex systems of differential equations, which can be computationally intensive and time-consuming. Although reduced-order modeling (ROM) techniques can mitigate this limitation, achieving real-time performance is still a challenge for high-fidelity simulations [37,39].

Interpretability and explainability constitute another critical axis of comparison. Physics-based models are inherently interpretable because their structure directly reflects physical laws and system parameters. This transparency facilitates validation, debugging, and trust in decision-making processes, which is essential in regulated industries such as aerospace, healthcare, and energy. Conversely, AI-based models, particularly deep learning architectures, are often considered “black boxes,” as their internal representations are not easily interpretable. This lack of transparency can hinder adoption in applications where accountability and explainability are required. However, recent advances in explainable AI (XAI) are beginning to address these limitations, although they do not yet fully match the clarity of physics-based formulations [33,44].

The fourth dimension, scalability, highlights the strengths of AI-based models in handling large-scale, high-dimensional systems. Data-driven approaches can efficiently process vast amounts of sensor data and scale across distributed architectures, making them well-suited for applications such as smart cities, industrial IoT and cloud-based

DT platforms. In contrast, physics-based models often face scalability challenges due to their computational complexity and the need for detailed system-specific parameterization. Extending such models to large or highly heterogeneous systems can be resource-intensive and time-consuming [34,35].

Finally, computational cost reflects the trade-off between training and inference phases. Physics-based models typically incur high computational costs during simulation, particularly for complex systems requiring fine discretization. AI-based models, on the other hand, require significant computational resources during the training phase but offer low-cost inference once deployed. This distinction has important implications for system design—physics-based approaches may be preferable when training data are scarce but computational resources are available, while AI-based methods are advantageous in environments, where large data flow is available, and which require rapid decision-making [39,43].

The comparative analysis in Figure 6 underscores that physics-based and AI-based DT approaches occupy complementary positions within the DT design space. Physics-based models excel in fidelity, interpretability, and extrapolation capability, while AI-based models provide superior speed, scalability, and adaptability. This contrast between approaches has led to the emergence of hybrid Digital Twin architectures, which aim to integrate the strengths of both paradigms. By embedding physical constraints into machine learning models or coupling reduced-order physics models with data-driven components, hybrid approaches achieve a balance between accuracy, efficiency, and robustness. Physics-Informed Neural Networks are a notable example of this trend, as they incorporate governing equations into the learning process, thereby improving generalization and reducing data requirements [37,45].

Looking more closely at the distinct modeling paradigms, physics-based DTs are anchored in first-principles equations derived directly from established physical laws. In the engineering world, this means relying on ordinary and partial differential equations, finite element models, computational fluid dynamics, electromagnetic models, thermal networks, or reduced-order models. Their greatest asset is interpretability—the model's variables and parameters map directly to real-world physical properties. Furthermore, as long as the underlying physical assumptions hold true, these models are highly reliable when you need to extrapolate outside of historically observed data [37,42,47]. The tradeoff, however, is performance. High-fidelity physics models can be incredibly computationally expensive and notoriously difficult to synchronize in real time, particularly when dealing with high-dimensional or complex multiphysics systems.

On the flip side, data-driven DTs learn entirely by observing historical and real-time system behavior. To do this, we lean on machine learning, deep learning, recurrent and convolutional neural networks, transformers, Gaussian processes, and reinforcement learning. These tools are highly effective for diagnostics, anomaly detection, estimating remaining useful life (RUL), visual inspection, and process optimization [33,43,48]. Their main strengths are low-latency inference (once the model is trained) and an uncanny ability to map highly complex, nonlinear relationships without needing a fundamental blueprint of the system's physics. However, they have significant blind spots. They are wholly dependent on the quality and representativeness of their training data, leaving them highly vulnerable to distribution shifts. They also suffer from limited explainability—the classic “black box” problem—and their performance can rapidly degrade if the physical asset wanders into operating regimes that were not represented in the training dataset [43,48,49].

This brings us to hybrid DTs, often referred to as “grey-box” models. These aim for the best of both worlds by fusing rigorous physical knowledge with flexible data-driven learning. The most common strategies here include residual learning, physics-guided machine learning, surrogate modeling, Bayesian updating, and Physics-Informed Neural Networks

(PINNs). In a residual setup, a physics-based model captures the known dynamics, while a machine-learning model runs alongside it to estimate any unmodeled effects or deviations caused by physical degradation. Meanwhile, PINNs literally embed physical constraints right into a neural network's loss function, ensuring the output remains physically consistent while drastically reducing the volume of training data required [37,44,45].

But hybrid modeling is not a universal solution. As recent literature indicates, PINNs and similar methods still wrestle with significant hurdles. These include training instability, tricky loss balancing, high computational costs, extreme sensitivity to boundary conditions, and scaling issues when applied to high-dimensional industrial setups. Consequently, extracting real value from a hybrid model demands rigorous model selection, strict validation, robust uncertainty quantification, and meticulous lifecycle management.

So, we can summarize that physics-based models are preferable when physical laws are well understood, data are scarce, interpretability is mandatory, and extrapolation is required. Data-driven models are preferable when large volumes of representative operational data are available, real-time inference is required, and the main goal is pattern recognition or prediction. Hybrid models are preferable when partial physical knowledge exists, operational data are available, and both interpretability and adaptive prediction are needed.

The radar chart presented in Figure 6 provides a visual synthesis of the fundamental trade-offs between physics-based and AI-based Digital Twins. Rather than representing competing alternatives, these paradigms should be viewed as complementary tools within a unified DT framework. The ongoing convergence toward hybrid methodologies reflects the broader evolution of DT systems toward more intelligent, adaptive, and context-aware cyber-physical solutions. This convergence is widely recognized in contemporary research as a key enabler for next-generation DT applications across industrial, healthcare, and environmental domains [32–35].

Table 6 summarizes the derived comparison outcomes, comparing the discussed DT paradigms.

Table 6. Practical comparison of Digital Twin modeling paradigms.

Criterion	Physics-Based Models	Data-Driven Models	Hybrid/Grey-Box Models
Main knowledge source	Physical laws and engineering models	Historical and streaming data	Physical laws combined with operational data
Typical methods	ODEs, PDEs, FEM, CFD, ROM	ML, DL, CNN, LSTM, transformers, reinforcement learning	Residual learning, PINNs, physics-guided ML, Bayesian updating
Interpretability	High	Low to moderate	Moderate to high
Data dependency	Low to moderate	High	Moderate
Extrapolation capability	Strong if assumptions remain valid	Weak outside training distribution	Moderate to strong when physics is informative
Real-time capability	Limited without model reduction	High after training	Potentially high, depending on implementation
Main limitation	Computational burden and modeling effort	Explainability, drift, data quality	Integration complexity, training stability, validation difficulty
Most suitable scenario	Data-scarce and physics-rich systems	Data-rich monitoring and prediction	Industrial DTs requiring both trust and adaptability

3.6. State-of-the-Art Limitations in DT Lifecycle Methodologies

The literature review reveals a clear trend: while individual DT enabling technologies, modeling approaches, and domain-specific applications have matured significantly, comprehensive, lifecycle-oriented methodologies remain heavily fragmented. Too often, papers focused on a single component, whether that is sensing, visualization, architecture, or predictive maintenance, without explaining how to weave these isolated pieces together across the twin's entire lifespan, from initial scoping to continuous, real-world refinement.

We identified several major methodological gaps in the current state of the art, discussed below.

The first gap is the incomplete lifecycle coverage. Current models frequently describe the moving parts of a DT, but they fail to provide actionable, reproducible steps for actually building and running one. We are missing clear guidelines for scoping the twin, selecting data sources, defining model granularity, synchronizing states, validating outputs, and continuously updating the system during operation. This leaves a frustrating gap between high-level conceptual definitions and practical, on-the-ground implementation.

The next gap relates to rigid data infrastructures. Many existing DT systems lean heavily on centralized data pipelines or highly platform-specific architectures. The reality of industrial automation, however, is inherently distributed. Factory floors are filled with scattered sensors, legacy controllers, PLC/SCADA systems, disparate enterprise databases, maintenance logs, and engineering models. Forcing all of this data into a centralized funnel often introduces severe latency, scalability bottlenecks, and thorny issues around data ownership and cybersecurity. While federated data integration and semantic abstraction offer the ideal solution, they are rarely operationalized effectively in current lifecycle approaches.

Another gap concerns unclear model integration. While physics-based, data-driven, and hybrid models are widely discussed, very few methodologies actually explain how to select, combine, synchronize, validate, and update these paradigms within a live DT environment. This is a critical oversight for industrial DTs, because real-world systems almost always require a delicate balance of physical interpretability and data-driven adaptability.

The next one reveals the lack of true closed-loop operation. As noted earlier in our taxonomy, many systems billed as “Digital Twins” are actually just Digital Shadows. They excel at monitoring and prediction but stop short of feeding operational decisions or control commands back into the physical system. A true DT requires rigorous mechanisms for validating decisions, checking safety constraints, authorizing actuation, and managing either human-in-the-loop or fully autonomous feedback.

Finally, we revealed insufficient governance and trust in many reviewed works. Because industrial DTs can directly dictate maintenance schedules, production flows, energy use, and physical safety, trust is essential. Features like robust cybersecurity, model versioning, data lineage tracking, clear validation statuses, explainability, access control, and comprehensive auditability cannot be treated as optional afterthoughts. They must be treated as fundamental lifecycle requirements.

These limitations represent the core motivation behind the Federated Digital Twin Lifecycle Model (F-DTLM) we propose later in this manuscript. To be clear, our contribution is not the invention of isolated elements like hybrid modeling, federated architectures, or closed-loop feedback—these concepts are already well-documented in the literature. Instead, the true value of the F-DTLM lies in structurally integrating these proven elements into a cohesive, actionable, and explicit lifecycle methodology.

3.7. Summary of the State-of-the-Art Synthesis

Our synthesis of the state-of-the-art literature reveals a clear duality: while DT research has achieved remarkable maturity regarding conceptual definitions, enabling technologies, modeling paradigms, and domain-specific applications, the field remains deeply fragmented on a methodological level. Existing studies offer immense value when tackling isolated components like architecture, sensing, modeling, predictive maintenance, interoperability, and control. Yet, very few of these contributions successfully weave those elements together into a cohesive, reproducible lifecycle methodology.

Ultimately, we can draw three distinct conclusions from this literature synthesis.

First, we must start evaluating DTs based on their overall lifecycle maturity, rather than judging them solely on their model fidelity or visual rendering capabilities.

Second, if we want to build genuinely scalable industrial DTs, hybrid modeling and federated data integration are no longer just optional features—they have become absolute, central requirements.

Third, we need a fundamental shift in perspective: interoperability, robust governance, and closed-loop feedback control must be elevated from mere implementation details to foundational, lifecycle-level requirements.

4. Theoretical and Mathematical Foundations of Digital Twins

Digital Twins are increasingly recognized as a core paradigm for modeling, monitoring, and controlling cyber–physical systems through continuous integration of data and models [32,37,41,42]. From a theoretical standpoint, a DT can be formulated as a dynamic, data-assimilative system that combines physical laws, observational data, and computational intelligence to represent and predict the behavior of a physical asset in real time. The mathematical foundation of DTs is commonly structured around three principal modeling paradigms: physics-based (white-box), data-driven (black-box), and hybrid (grey-box) approaches [41–44,47,48].

In the context of this paper, the mathematical formulations presented in this section are not treated as isolated abstractions. Rather, they provide the analytical mechanisms required to address the methodological gaps identified in the state-of-the-art synthesis. Specifically, they support: (i) formal scoping of system states, inputs, outputs, and measurable variables; (ii) definition of the data streams required for synchronization; (iii) integration of physics-based, data-driven, and hybrid models; (iv) recursive state estimation and uncertainty-aware updating; and (v) prediction, simulation, and operational decision support. These requirements are later operationalized in Section 6 through the proposed Federated Digital Twin Lifecycle Model (F-DTLM). Therefore, Section 4 should be read as the theoretical bridge between the state-of-the-art review and the lifecycle methodology introduced later.

4.1. General State-Space Formulation

A Digital Twin can be generically described using a nonlinear state-space representation. Let $x(t) \in \mathbb{R}^n$ denote the system state, $u(t) \in \mathbb{R}^m$ the input, and $y(t) \in \mathbb{R}^p$ the observed output. The system dynamics can be expressed as:

$$\dot{x}(t) = f(x(t), u(t), \theta, t) + w(t), \quad (1)$$

$$y(t) = h(x(t), u(t), \theta, t) + v(t), \quad (2)$$

where θ represents model parameters, while $w(t)$ and $v(t)$ denote process and measurement noise, respectively. In discrete-time form:

$$x_{k+1} = f(x_k, u_k, \theta) + w_k, \quad (3)$$

$$y_k = h(x_k, u_k, \theta) + v_k. \quad (4)$$

The DT objective is to estimate the hidden state x_k identify parameters θ , and predict future system behavior using continuously updated observations $y_{1:k}$ [37,42]. The specific structure of functions $f(\cdot)$ and $h(\cdot)$ defines the modeling paradigm.

At the lifecycle-design level, Equations (1)–(4) support the formal definition of system states, measurable outputs, inputs, disturbances, and model parameters. These variables later determine the scope of the DT and the data streams required for synchronization. From a data-infrastructure perspective, the output equation defines which measurements must be acquired, time-aligned, semantically described, and made available to the DT model. For hybrid synchronization, residual modeling provides a mechanism for combining physics-based predictions with data-driven corrections. For operational decision support, the synchronized state provides the basis for prediction, what-if simulation, RUL estimation, optimization, and control recommendations.

4.2. Physics-Based Modeling (White-Box)

Physics-based DTs rely on first-principles modeling, where system dynamics are derived from physical laws such as conservation equations, thermodynamics, and continuum mechanics [32,47]. These models are typically expressed using ordinary or partial differential equations.

A representative formulation for mechanical systems is:

$$M(\theta)\ddot{q}(t) + C(\theta)\dot{q}(t) + K(\theta)q(t) = F(t), \quad (5)$$

where M , C and K denote mass, damping, and stiffness matrices, respectively. For distributed systems, PDE-based formulations are often used:

$$\frac{\partial x}{\partial t} = \nabla \cdot (D(x, \theta) \nabla x) + s(x, u, \theta, t). \quad (6)$$

Parameter estimation is typically formulated as an optimization problem:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \sum_{k=1}^N \|y_k - \hat{y}_k(\theta)\|^2, \quad (7)$$

or in Bayesian form:

$$p(\theta|y_{1:N}) \propto p(y_{1:N}|\theta)p(\theta). \quad (8)$$

Physics-based DTs provide high interpretability and strong extrapolation capabilities, as their predictions remain consistent with governing laws. However, they often suffer from high computational complexity, particularly in real-time applications, requiring reduced-order models or surrogate approximations [47,50].

4.3. Data-Driving Modeling (Black-Box)

Data-driven DTs approximate system behavior directly from data using statistical learning and artificial intelligence techniques [41,43,48]. Instead of explicitly modeling physical laws, these approaches learn a mapping:

$$\hat{y} = g_{\phi}(z), \quad (9)$$

where z represents input features and ϕ denotes model parameters. For dynamic systems:

$$x_{k+1} = f_{\phi}(x_k, u_k), y_k = h_{\phi}(x_k, u_k) \quad (10)$$

Model training is commonly formulated as empirical risk minimization:

$$\hat{\phi} = \operatorname{argmin}_{\phi} \frac{1}{N} \sum_{i=1}^N L(y_i, g_{\phi}(z_i)) + \lambda \Omega(\phi), \quad (11)$$

where L is a loss function, and $\Omega(\phi)$ is a regularization term.

Data-driven models are highly flexible and capable of capturing nonlinear relationships in complex systems. They are particularly effective for real-time inference due to low computational cost after training. However, their main limitations include reduced interpretability and poor generalization outside the training data distribution [48,49].

4.4. Hybrid Modeling (Grey-Box)

Hybrid DTs combine physical models with data-driven components to leverage the strengths of both paradigms [43,44,48]. One common formulation is residual modeling:

$$x_{k+1} = f_{phys}(x_k, u_k, \theta) + r_{\phi}(x_k, u_k), \quad (12)$$

where $r_{\phi}(\cdot)$ represents a learned correction term.

A more advanced approach involves Physics-Informed Neural Networks (PINNs), where physical laws are embedded into the training process. The loss function is defined as:

$$L(\phi) = \lambda_d L_{data} + \lambda_p L_{physics} + \lambda_b L_{BC/IC}, \quad (13)$$

with the physics constraint given by:

$$L_{physics} = \frac{1}{N_r} \sum_{j=1}^{N_r} \left\| \frac{\partial u_{\phi}}{\partial t}(x_j, t_j) - N(u(x_j, t_j), x_j, t_j) \right\|^2, \quad (14)$$

Hybrid models also employ state estimation techniques such as the Kalman filter:

$$\hat{x}_{k|k-1} = f_{phys}(\hat{x}_{k-1|k-1}, u_{k-1}), \quad (15)$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k (y_k - h(\hat{x}_{k|k-1})). \quad (16)$$

These approaches enable continuous model updating and uncertainty-aware estimation, making hybrid DTs particularly suitable for real-time industrial applications [37,45].

4.5. Uncertainty Quantification and Model Updating

A key theoretical aspect of DTs is uncertainty handling. Bayesian filtering provides a unified framework:

$$p(x_k, \theta | y_{1:k}) \propto p(y_k | x_k, \theta) p(x_k, \theta | y_{1:k-1}), \quad (17)$$

which enables recursive updating as new data become available [37,45]. This formulation supports both parameter estimation and state inference under uncertainty.

4.6. Relationship Between Mathematical Formulations and Lifecycle-Oriented DT Requirements

To clarify how the preceding formulations are used within the proposed lifecycle methodology, Table 7 maps the mathematical concepts to the corresponding F-DTLM phases.

Table 7. Relationship between mathematical formulations and F-DTLM phases.

Mathematical Formulation	Main Role in DT Modeling	Lifecycle-Oriented Requirement Supported	Later Operationalization in F-DTLM
State-space model, Equations (1)–(4)	Defines states, inputs, outputs, dynamics, and observations	DT scoping and synchronization	Definition and Scoping; Hybrid Modeling and Synchronization
Physics-based model, Equations (5) and (6)	Represents known physical behavior	Interpretable modeling and simulation	Hybrid Modeling and Synchronization
Parameter estimation, Equations (7) and (8)	Calibrates model parameters from observations	Model updating and uncertainty-aware calibration	Hybrid Modeling and Synchronization
Data-driven model, Equations (9)–(11)	Learns nonlinear mappings from operational data	Predictive analytics from federated data	Data and Infrastructure Federation; Hybrid Modeling and Synchronization
Residual hybrid model, Equation (12)	Combines physics-based dynamics with learned corrections	Hybrid model fusion	Hybrid Modeling and Synchronization
PINN loss, Equations (13) and (14)	Embeds physical constraints into learning	Physics-guided learning	Hybrid Modeling and Synchronization
State estimation, Equations (15) and (16)	Corrects predicted states using measurements	Real-time synchronization	Hybrid Modeling and Synchronization; Operational Optimization and Control
Bayesian updating, Equation (17)	Updates state/parameter beliefs under uncertainty	Uncertainty-aware decision support	Governance and Trust; Operational Optimization and Control

The formulations summarized in Table 7 define the analytical capabilities required for lifecycle-oriented DT implementation. Section 5 next discusses the technical architecture and software stack needed to deploy these capabilities, while Section 6 integrates both the mathematical and architectural layers into the proposed F-DTLM framework.

4.7. Comparative Analysis

Theoretical differences between DT modeling paradigms are summarized in Table 8.

Table 8. Comparison of Digital Twin modeling paradigms.

Criterion	Physics-Based	Data-Driven	Hybrid
Model basis	First-principles laws	Data-driven approximation	Combined physics and data
Mathematical form	ODEs, PDEs	ML/DL models	Residual models, PINNs
Interpretability	High	Low	Moderate–High
Data requirement	Low–Moderate	High	Moderate
Extrapolation	Strong	Weak	Moderate–Strong

Table 8. Cont.

Criterion	Physics-Based	Data-Driven	Hybrid
Real-time capability	Limited	High	High
Computational cost	High	Training high, inference low	Moderate
Robustness	High (physics-consistent)	Sensitive to data shift	Balanced

5. Technical Architecture and Software Stack for Digital Twin System

The implementation of Digital Twin systems relies on a structured, layered technical architecture that integrates physical assets, communication infrastructure, data management, modeling services, and application-level interfaces. Contemporary research emphasizes that such architectures must be modular, interoperable, and standards-compliant to ensure scalability and long-term maintainability in industrial environments [51–54]. The reference architecture illustrated in Figure 7 reflects this layered paradigm and can be aligned with widely accepted standards such as ISO 23247 and RAMI 4.0.

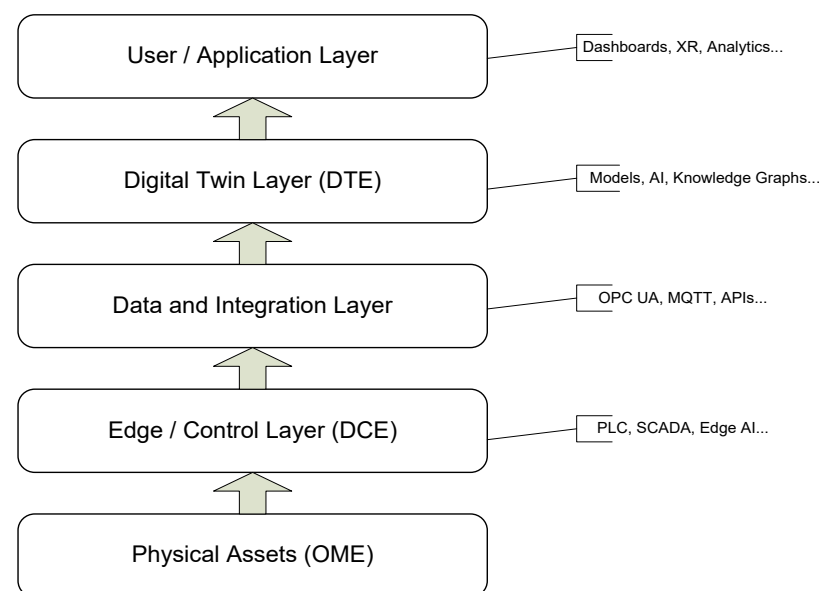


Figure 7. Reference Digital Twin architecture aligned with ISO 23247, illustrating the interaction between physical assets, edge communication, digital twin entities, and user applications.

5.1. Layered Architecture and Mapping to Standards

The architecture in Figure 7 consists of five principal layers: Physical Assets (OME), Edge/Control Layer (DCE), Data and Integration Layer, Digital Twin Layer (DTE), and User/Application Layer. This structure is consistent with ISO 23247, which defines DT systems as interacting entities including observable physical elements, data collection and control components, core digital representations, and user-facing services [53]. Specifically:

- The Physical Assets layer corresponds to real-world systems (machines, sensors, actuators);
- The Edge/Control layer aligns with the Data Collection and Device Control Entity, handling real-time acquisition and actuation via PLCs, SCADA systems, and embedded controllers;

- The Data and Integration layer supports interoperability and data exchange across heterogeneous systems;
- The Digital Twin layer implements models, analytics, simulation, and semantic representations;
- The User/Application layer provides visualization, decision support, and enterprise integration (e.g., dashboards, ERP, PLM).

In parallel, RAMI 4.0 situates these layers within a three-dimensional framework that combines hierarchy levels, lifecycle phases, and interoperability layers. The Digital Twin corresponds to the digital representation of assets, typically implemented via the Asset Administration Shell (AAS), which standardizes asset metadata, capabilities, and interfaces [52,55–57]. The AAS plays a central role in bridging the Data/Integration and Digital Twin layers by providing a machine-readable semantic structure for asset information.

5.2. Connectivity and Communication Protocols

Interoperability in DT systems depends critically on communication protocols that enable reliable, secure, and scalable data exchange. The literature identifies OPC UA and MQTT as the dominant complementary protocols [54,58,59].

OPC UA (Open Platform Communications Unified Architecture) is widely used in industrial automation due to its support for:

- Platform-independent communication;
- Rich information modeling;
- Built-in security mechanisms.

Mathematically, OPC UA can be interpreted as enabling structured data exchange through graph-based information models. This allows DT systems to operate on semantically enriched data rather than raw signals [58].

MQTT (Message Queuing Telemetry Transport) follows a publish–subscribe paradigm, where data streams are distributed via brokers. Formally, this can be represented as a mapping:

$$T : \text{Publisher} \rightarrow \text{Broker} \rightarrow \text{Subscribers},$$

enabling asynchronous and scalable data dissemination. MQTT is particularly suitable for edge-to-cloud communication due to its low overhead and high scalability [59].

In practice, DT architectures often combine both protocols: OPC UA is used for device-level integration, while MQTT is used for streaming data across distributed systems [54,58,59].

5.3. Platform Ecosystem and Deployment Methods

Modern DT systems are deployed using cloud, edge, or hybrid architectures, depending on latency, computational and security requirements [51,54]. Let T_{lat} denote latency and C_{comp} computational demand. Then:

- Edge deployment minimizes T_{lat} for time-real control;
- Cloud deployment maximizes C_{comp} for large-scale analytics;
- Hybrid deployment balances both.

DT platforms can be broadly categorized into:

1. Industrial IoT platforms—providing connectivity, storage, and visualization.
2. Domain-specific DT platforms—integrating simulation, lifecycle management, and optimization.
3. Standards-based ecosystems—focusing on interoperability (e.g., AAS-based systems).

These platforms are rarely used separately. Instead, DT solutions are typically composed as multi-layer software stacks, combining industrial control systems, middleware, semantic models, and analytics services [54,57,60].

5.4. Comparative Analysis of Standards, Protocols and Platforms

The key components of DT technical architecture are summarized in Table 9.

Table 9. Comparison of Digital Twin architectural standards, protocols, and platform categories.

Category	Example	Role in DT Architecture	Key Features	Advantages	Limitations
Reference Architecture	ISO 23247	DT-specific architecture	Defines DT entities and interactions	Clear DT structure; manufacturing focus	Limited generalization beyond manufacturing
	RAMI 4.0	Industry 4.0 framework	Multi-dimensional architecture (layers, lifecycle, hierarchy)	Strong interoperability and lifecycle integration	Higher complexity
Semantic Model	Asset Administration Shell (AAS)	Digital asset representation	Standardized metadata and interfaces	Interoperability; modularity	Implementation overhead
Communication Protocol	OPC UA	Device-level communication	Secure, structured information models	High interoperability; semantic richness	Higher complexity than lightweight protocols
	MQTT	Data streaming	Publish–subscribe messaging	Lightweight; scalable; real-time capable	Limited semantic structure
Deployment Model	Edge Computing	Local processing	Low latency, real-time control	Fast response; reduced bandwidth	Limited computational power
	Cloud Computing	Centralized processing	High scalability and storage	Powerful analytics; global access	Latency; security concerns
	Hybrid (Edge–Cloud)	Combined architecture	Distributed processing	Balanced performance and scalability	Increased system complexity
Platform Type	Industrial IoT Platforms	Connectivity and data management	Device integration, dashboards	Rapid deployment	Vendor dependency
	Domain-Specific DT Platforms	Simulation and optimization	Integrated DT workflows	High functionality	Limited flexibility
	Standards-Based Ecosystems	Interoperability frameworks	AAS-based modular design	Vendor-neutral; extensible	Requires integration effort

The comparative analysis demonstrates that no single architectural solution dominates across all DT applications. Instead, effective DT implementation requires integration of multiple standards, protocols, and platforms. ISO 23247 provides a DT-centric structural foundation, while RAMI 4.0 ensures alignment with broader Industry 4.0 ecosystems. OPC UA and MQTT enable complementary communication patterns, and hybrid cloud–edge deployments balance latency and computational requirements.

The architecture shown in Figure 7 reflects this integration: it combines physical systems, control infrastructure, data pipelines, semantic representations, and application services into a unified DT ecosystem. Such layered architectures are now considered essential for achieving scalability, interoperability, and real-time responsiveness in modern Digital Twin systems [51–54,57–60].

6. Methodology: Federated Digital Twin Lifecycle Model (F-DTLM)

6.1. Methodology Rationale and Derivation Logic

Despite the rapid advancement of Digital Twin technologies, current implementations remain fragmented, frequently addressing isolated lifecycle stages such as monitoring, simulation, predictive maintenance, visualization, or control without a unified methodological structure. The state-of-the-art synthesis in Section 3 shows that this fragmentation occurs not because individual DT technologies are immature, but because their integration across the full lifecycle remains insufficiently formalized. In particular, existing approaches often lack an explicit connection among lifecycle definition, distributed data infrastructure, hybrid modeling, state synchronization, governance, and closed-loop operation. As illustrated in Figure 8, physics-based models offer high interpretability and physical fidelity but suffer from computational complexity and limited scalability. In contrast, data-driven models provide high scalability and real-time inference capabilities but lack explainability and robustness outside the training domain. Hybrid approaches, particularly physics-informed and residual models, aim to bridge this gap by combining physical knowledge with data-driven adaptability [37,44,45]. However, existing DT methodologies often treat these paradigms in isolation rather than as complementary components within a unified lifecycle.

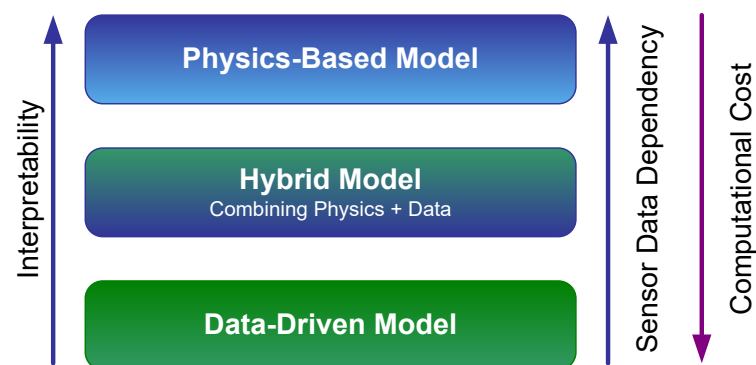


Figure 8. Comparison of physics-based, data-driven, and hybrid modeling paradigms used in Digital Twin implementations, highlighting trade-offs in interpretability, scalability, and data dependency.

The proposed Federated Digital Twin Lifecycle Model (F-DTLM) was therefore constructed using a requirement-driven synthesis of the review findings, rather than as an isolated conceptual framework. The construction procedure consisted of the following steps.

First, the reviewed literature was analyzed according to the extraction categories defined in Section 2, including lifecycle coverage, modeling paradigm, data architecture, interoperability support, feedback capability, governance mechanisms, and validation level. Then, recurring limitations were grouped into methodological requirements: explicit lifecycle definition, federated data integration, hybrid model construction, state synchronization, operational feedback, semantic interoperability, and trust governance. Next, these requirements were mapped into lifecycle phases and cross-cutting layers. Finally, the resulting framework was checked for consistency with the theoretical formulations in Section 4 and the architectural standards and technologies discussed in Section 5.

Thus, F-DTLM is not intended to replace established reference architectures such as ISO 23247 or RAMI 4.0. Rather, it is intended to complement them by providing a lifecycle-oriented methodology that explains how DT development can proceed from problem definition to data federation, hybrid modeling, operational optimization, and iterative refinement. ISO 23247 and RAMI 4.0 define important architectural and interoperabil-

ity principles, while F-DTLM translates such principles into a reproducible sequence of methodological activities.

The framework is based on three design principles:

1. Federation over centralization, because industrial DT data are commonly distributed across sensors, PLC/SCADA systems, edge nodes, enterprise systems, engineering models, and historical maintenance databases;
2. Hybrid modeling over single-paradigm modeling, because real industrial systems often require both physical interpretability and data-driven adaptability;
3. Closed-loop lifecycle continuity, because a full Digital Twin must support not only monitoring and prediction, but also decision feedback, operational adaptation, and continuous refinement.

This derivation logic clarifies the novelty of F-DTLM. The novelty does not lie in introducing federation, hybrid modeling, interoperability, or feedback as isolated concepts. These elements already exist in the literature. Instead, the contribution lies in integrating them into a single lifecycle methodology in which each element is assigned a specific role, input, activity set, output, and governance responsibility.

6.2. Overview of the F-DTLM Framework

The F-DTLM framework structures the Digital Twin lifecycle into four iterative phases, supported by two cross-cutting layers, as illustrated in Figure 9. The lifecycle is inherently cyclic, with outputs from the operational phase continuously feeding back into earlier phases to enable model refinement, infrastructure adaptation, and evolving system objectives.

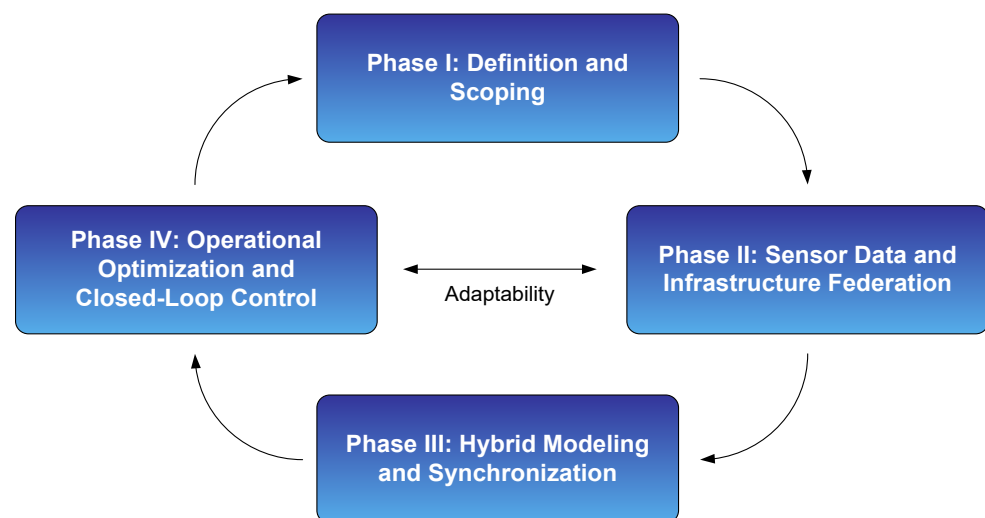


Figure 9. Proposed Digital Twin lifecycle framework covering definition, data infrastructure, hybrid modeling, and operational optimization, emphasizing closed-loop feedback between cyber and physical layers.

The four core phases are:

1. Definition and Scoping.
2. Data and Infrastructure Federation.
3. Hybrid Modeling and Synchronization.
4. Operational Optimization and Control.

This structure reflects the transformation from conceptual design to operational intelligence while maintaining continuous alignment with physical system evolution. The

bidirectional feedback loop shown in Figure 9 emphasizes adaptability as a key property of the DT lifecycle.

6.3. Phase Priorities and Typical Iteration Paths

While we present the F-DTLM as a structured, four-phase lifecycle, it is critical to understand that this is not a rigid, linear waterfall model. In practical industrial deployments, the relative priority—and even the starting point—of each phase dynamically shifts based on the physical asset’s maturity, the existing sensor infrastructure, the specific modeling objectives, and the level of operational risk tied to the system’s feedback decisions.

For completely new (greenfield) DT deployments, the logical iteration path must begin with rigorous Definition and Scoping. Without clearly delineated system boundaries and precisely defined Key Performance Indicators (KPIs), developers risk introducing unnecessary model complexity and weak validation criteria. This foundational phase becomes essential when the DT is tasked with supporting safety-critical decisions, predictive maintenance schedules, or high-stakes operational optimization.

Conversely, when dealing with legacy or brownfield industrial systems, the lifecycle practically kicks off at the Data and Infrastructure Federation phase. In these environments, the physical assets are usually already outfitted with PLCs, SCADA networks, historical databases, maintenance logs, and fragmented sensor arrays. Here, the primary hurdle is rarely a lack of data; rather, it is the profound heterogeneity, temporal inconsistency, and lack of semantic interoperability across these siloed data sources.

For analytically mature systems where the infrastructure is already well-established, the dominant iterative loop typically cycles between Hybrid Modeling and Synchronization and Operational Optimization and Control. In this continuous feedback loop, the system rigorously evaluates model outputs against empirical operational outcomes. Any statistically significant deviations between predicted and observed behaviors immediately trigger corrective actions—whether that means recalibrating the core model, updating residual estimators, executing sensor-quality diagnostics, or revising operational control constraints.

Consequently, the F-DTLM explicitly accommodates at least three distinct, empirically grounded iteration paths:

1. Design-driven iteration. Definition and Scoping → Data Federation → Hybrid Modeling → revised scope or KPIs;
2. Data-driven iteration. Data Federation → Hybrid Modeling → data-quality assessment → revised sensor/data integration;
3. Operation-driven iteration. Operational Optimization → observed outcome → model recalibration → updated decision logic.

Ultimately, framing the framework this way dramatically improves its practical reproducibility. It clarifies that while our four phases define essential lifecycle functions, real-world industrial deployments will fluidly and iteratively navigate among them, dynamically adapting to the realities of data availability, model maturity, and shifting operational imperatives.

6.4. Phase I: Definition and Scoping

The objective of this phase is to establish the functional, physical, and economic boundaries of the Digital Twin. Inputs include physical asset specifications, operational constraints, performance indicators (KPIs), and regulatory requirements.

Key activities include:

- Granularity selection, determining whether the DT models a component, system, or system-of-systems;
- Objective formalization, translating business goals into measurable KPIs;

- Boundary definition, specifying included phenomena and time scales;
- Stakeholder alignment, ensuring consistency across engineering and operational domains.

The outputs of this phase include a Digital Twin blueprint, KPI–sensor mappings, and clearly defined system boundaries. This phase is critical to prevent scope creep and to ensure that subsequent stages remain goal-oriented rather than technology-driven [32,34].

6.5. Phase II: Data and Infrastructure Federation

This phase establishes a federated, interoperable data backbone connecting Operational Technology (OT) and Information Technology (IT) layers. Inputs include real-time sensor data, control system data (PLC/SCADA), and enterprise systems such as ERP and PLM.

Unlike centralized architectures, F-DTLM adopts a federated data approach, where data sources remain distributed but are unified through semantic abstraction. Key activities include:

1. Protocol integration, combining protocols such as OPC UA and MQTT for interoperability and scalability [54,59].
2. Edge computing deployment, enabling low-latency preprocessing and filtering.
3. Semantic wrapping, using standardized representations such as Asset Administration Shell (AAS) for machine-readable data exchange [61].

Outputs include a unified data abstraction layer, context-aware data streams, and secure communication channels. This phase ensures scalability, reduces latency, and supports integration across heterogeneous systems.

6.6. Phase III: Hybrid Modeling and Synchronization

This phase represents the analytical core of the Digital Twin, where system behavior is modeled and synchronized with real-world observations.

Inputs include design models (CAD, physics-based equations), historical and real-time data, and domain knowledge.

F-DTLM explicitly promotes hybrid (grey-box) modeling, integrating:

- Physics-based models for interpretability and extrapolation;
- Data-driven models for capturing nonlinear relationships;
- Model fusion, where machine learning compensates for model uncertainties;
- State synchronization, using recursive estimation methods (e.g., Kalman filtering, Bayesian inference) [37,45].

Outputs include a synchronized DT state, predictive behavior with uncertainty bounds, and a real-time virtual representation of the physical system. This phase enables predictive and prescriptive capabilities beyond descriptive analytics.

6.7. Phase IV: Operational Optimization and Control

The objective of this phase is to generate operational value by closing the cyber–physical loop.

Inputs include DT predictions, operational constraints, and optimization objectives.

Key activities include:

1. Anomaly detection and diagnostics.
2. Predictive maintenance planning.
3. What-if simulation and optimization.
4. Closed-loop control, where decisions are either recommended or automatically executed [34,42].

Outputs include control actions, maintenance schedules, and updated performance metrics. The explicit feedback loop distinguishes a true Digital Twin from a Digital Shadow.

6.8. Cross-Cutting Layers

6.8.1. Semantic Interoperability Layer

The semantic interoperability layer serves as the universal translator of the DT lifecycle, ensuring that data, models, and services maintain a consistent, unambiguous meaning from end to end. In real-world industrial deployments, sensor telemetry is typically generated by a highly heterogeneous mix of devices. This data is then scattered across disparate systems with wildly inconsistent naming conventions, units, sampling rates, and metadata structures. Without strict semantic harmonization, a single variable could easily be misinterpreted by the control layer, the analytics engine, the maintenance database, and the operator's dashboard [61].

To solve this, the F-DTLM operationalizes semantic interoperability through three core mechanisms:

1. **Explicit Linking.** Every monitored variable is directly tied to a specific asset, physical meaning, unit of measurement, sampling frequency, and raw data source.
2. **Standardized Descriptions.** Leveraging standardized asset descriptions, such as the Asset Administration Shell (AAS), to expose highly structured, machine-readable sub-models for individual sensors, physical components, operating states, and historical maintenance records.
3. **Semantic Mapping.** Utilizing sophisticated semantic mappings or knowledge graphs to tightly interconnect the physical assets with their corresponding data streams, model variables, known failure modes, and operational decisions.

Consider a DT for an electrical motor as a practical example. The generic variable “temperature” is practically useless on its own; it must be semantically distinguished as winding temperature, bearing temperature, ambient temperature, or inverter temperature. Similarly, any extracted vibration features must be intrinsically linked to the exact sensor placement, axis orientation, sampling frequency, and specific spectral bands. Ultimately, this robust semantic layer empowers the hybrid model to ingest data consistently and ensures that any resulting maintenance decisions remain fully traceable back to hard, physical evidence.

6.8.2. Governance and Trust Layer

The governance and trust layer tackles the critical imperatives of reliability, security, traceability, and auditability across every phase of the DT lifecycle. Its primary mission is simple but essential: to guarantee that the DT's outputs can be implicitly trusted when they are used to drive maintenance schedules, optimize processes, or execute direct control decisions.

Within the F-DTLM, we operationalize this governance through several strict mechanisms:

- **Data ownership and access control**—to clearly define exactly which users, digital services, and external systems are permitted to access raw sensor data, model outputs, and control recommendations;
- **Cybersecurity and secure communication**—to enforce robust encryption, authentication, and authorization protocols, securing API access, and continuously monitoring for anomalous data streams or potentially compromised edge nodes;
- **Model validation and versioning**—to maintain a rigorous registry of model versions, including their specific training datasets, calibration dates, validation metrics, and current deployment statuses;

- Data lineage and audit trails—to provide unbroken traceability, from the initial raw measurements all the way through to feature extraction, model predictions, automated decisions, and subsequent operator or control-system actions;
- Decision accountability—to explicitly classify every DT decision output as advisory, human-approved, semi-automated, or fully autonomous;
- Regulatory and safety compliance—to continuously verify that any control recommendations strictly respect physical operational constraints, safety thresholds, and all applicable industrial standards.

Embedding governance within the lifecycle ensures trustworthy and auditable DT systems, which is critical for industrial adoption [32,54].

To illustrate, suppose our DT recommends reducing a motor's operational load because it predicts imminent bearing degradation. In this scenario, the governance layer instantly is applied to preserve a permanent record of the specific model version used, the exact input features analyzed, the statistical confidence level of the prediction, the underlying decision rule, the operator's confirmation, and the final physical action taken. In safety-critical or highly regulated industrial environments, this level of exhaustive explainability and auditability is not just good practice, it is an essential requirement.

6.9. Comparative Analysis of Digital Twin Lifecycle and Reference Models

Existing DT lifecycle and reference models provide important contributions but differ in their objectives. Some frameworks originate from product lifecycle management, some focus on generic DT stages, while others define architectural entities or interoperability principles. Therefore, the comparison in this section does not aim to claim empirical superiority of F-DTLM. Instead, it evaluates how explicitly each model operationalizes lifecycle requirements identified in the structured review.

To reduce subjective self-positioning, the comparison uses the following scoring rubric:

- 0—Not addressed;
- 1—Mentioned indirectly or only implied;
- 2—Explicitly defined;
- 3—Operationalized as a lifecycle mechanism.

The scoring evaluates conceptual and methodological coverage, not experimental performance. Therefore, a high score does not imply that a framework has been empirically validated; it indicates only that the corresponding requirement is explicitly represented in the framework (Table 10).

Table 10. Neutral comparison of Digital Twin lifecycle and reference models using a requirement-coverage scoring rubric.

Evaluation Criterion	Grieves PLM Model	Generic DT Lifecycle (Survey-Based)	ISO 23247 Reference Model	RAMI 4.0 Lifecycle View	F-DTLM (This Work)
Explicit lifecycle phases	1	2	1	1	3
DT scoping and boundary definition	1	2	1	1	3
Data acquisition and integration	1	2	3	2	3
Federated/distributed data support	0	1	2	2	3

Table 10. *Cont.*

Evaluation Criterion	Grieves PLM Model	Generic DT Lifecycle (Survey-Based)	ISO 23247 Reference Model	RAMI 4.0 Lifecycle View	F-DTLM (This Work)
Edge–cloud deployment consideration	0	1	2	2	3
Hybrid physics/data-driven modeling	0	1	1	1	3
State synchronization and updating	1	2	2	1	3
Closed-loop operational feedback	1	1	2	1	3
Semantic interoperability	0	1	2	3	3
Governance, trust, and auditability	0	1	1	1	3
Practical iteration/refinement logic	1	2	1	1	3
Cross-domain methodological transferability	1	2	1	1	2

The table indicates that Grieves' PLM model is foundational for linking physical and virtual product representations, but it does not operationalize modern DT requirements such as federation, hybrid modeling, semantic interoperability, and governance. Generic DT lifecycle models usually provide lifecycle stages but often remain high-level. ISO 23247 provides strong manufacturing-oriented architectural guidance and data-entity definitions, while RAMI 4.0 provides a broader Industry 4.0 interoperability and lifecycle context. F-DTLM differs from these approaches by explicitly organizing the reviewed requirements into lifecycle phases and cross-cutting layers.

This comparison should be interpreted as a requirement-coverage analysis, not as proof of empirical superiority. The practical performance of F-DTLM still requires experimental validation and benchmarking in future work. The value of F-DTLM lies in its methodological integration: it connects lifecycle scoping, federated infrastructure, hybrid modeling, synchronization, operational feedback, semantic interoperability, and governance within one reproducible framework.

The comparison clarifies the specific added value of F-DTLM relative to existing frameworks. ISO 23247 and RAMI 4.0 are essential for interoperability and architectural alignment, but they do not prescribe a full implementation lifecycle from scoping through hybrid modeling and operational control. Generic lifecycle models define broad development stages but often provide limited guidance on federation, governance, and hybrid synchronization. F-DTLM addresses this methodological gap by structuring these requirements into lifecycle phases and by treating semantic interoperability and governance as continuous layers rather than optional implementation details.

7. Illustrative Application Scenario: F-DTLM for Electrical Motor Monitoring and RUL Prediction

7.1. Purpose and Scope of the Application Scenario

Electrical motors, particularly induction motors, are among the most widely used assets in industrial automation systems. Their failures may result in unplanned downtime, reduced energy efficiency, production losses, and increased maintenance costs. For this

reason, motor health monitoring and remaining useful life (RUL) prediction represent a relevant application domain for demonstrating how a lifecycle-oriented Digital Twin methodology can be instantiated in practice [62–65]. Digital Twin technology has emerged as a promising solution for this purpose, enabling real-time monitoring, fault diagnosis, and prognostics through the integration of sensor data, physical models, and machine learning algorithms [66,67].

Recent studies demonstrate that DT-based frameworks for electric motors can integrate multi-sensor measurements, such as current, voltage, vibration, and temperature—with high-fidelity models to detect early-stage faults, including bearing wear, rotor bar defects, and stator winding degradation [66]. Furthermore, DT-enabled predictive maintenance has been shown to improve reliability, reduce downtime, and optimize maintenance scheduling by shifting from reactive to proactive strategies [68].

This section presents an illustrative application scenario in which the proposed F-DTLM framework is applied to electrical motor monitoring and RUL-oriented predictive maintenance. The purpose of the scenario is not to claim empirical validation of F-DTLM, since no original experimental dataset, baseline comparison, or statistical benchmarking is introduced in this paper. Instead, the scenario demonstrates how the framework can structure the development of a motor-oriented Digital Twin by specifying the asset scope, data sources, modeling strategy, synchronization mechanism, evaluation metrics, and operational decision pathway.

Accordingly, the scenario should be interpreted as a methodological instantiation rather than a full industrial deployment. Its role is to show how the F-DTLM phases can be translated into a concrete engineering workflow and to define the evaluation criteria required for future experimental validation.

7.2. System Context and Monitoring Objectives

The considered physical asset is a three-phase induction motor operating in an industrial environment. The motor is treated as a component-level Digital Twin target, although the same workflow can be extended to a drive system, pump unit, conveyor, compressor, or larger production subsystem.

The primary monitoring objectives are as follows:

1. Early detection of abnormal operating conditions;
2. Identification of degradation patterns associated with electrical, mechanical, and thermal faults;
3. Estimation of RUL for critical components such as bearings, windings, and rotor elements;
4. Support for condition-based and predictive maintenance decisions;
5. Provision of operator recommendations or control constraints when degradation risk increases.

The application scenario focuses on health monitoring and prognostics, not on full autonomous control. Therefore, the expected feedback mode is primarily human-in-the-loop or human-on-the-loop, where the DT generates warnings, RUL estimates, maintenance recommendations, and possible operating constraints, while final intervention decisions remain under operator or maintenance-engineer supervision, which is consistent with recent DT implementations for electrical machines, which combine edge computing, cloud analytics, and hybrid modeling techniques [66,69].

7.3. Phase I: Definition and Scoping

In the first F-DTLM phase, the physical and functional boundaries of the motor Digital Twin are defined. The monitored object is scoped as a single motor unit, including its electrical input, electromagnetic conversion process, mechanical response, thermal

behavior, and selected degradation mechanisms. External production-level factors, such as scheduling and supply-chain constraints, are outside the immediate boundary of this scenario, although they may be integrated in future system-level extensions.

The main KPIs are selected according to the predictive-maintenance objective:

- fault-detection accuracy;
- false-alarm rate;
- RUL prediction error;
- synchronization latency;
- edge-processing time;
- maintenance lead time;
- model-update frequency;
- energy-efficiency deviation;
- downtime reduction potential.

The scoping phase also defines which fault modes are relevant. For induction motors, common fault categories include bearing defects, stator winding faults, rotor bar faults, eccentricity, insulation degradation, and thermal overload. The inclusion of these categories determines the required sensor configuration, feature-extraction methods, and modeling strategy.

The output of this phase is a motor DT blueprint containing asset boundaries, fault categories, KPIs, sensor requirements, data processing logic, and intended feedback mode.

7.4. Phase II: Data and Infrastructure Federation

The second phase establishes the federated data infrastructure required to connect the physical motor, edge-processing unit, industrial control systems, and higher-level analytics services. The scenario assumes that raw and processed data may originate from heterogeneous sources, including local sensors, PLCs, SCADA systems, maintenance logs, drive controllers, and enterprise databases.

The proposed sensor set includes electrical, mechanical, thermal, and speed-related measurements. The rationale for each measurement channel is summarized in Table 11.

Table 11. Sensor and data source rationale for the motor-monitoring application scenario.

Data Source	Typical Measurement	Diagnostic Relevance	Rationale for Inclusion
Current sensors	Stator phase currents	Rotor faults, stator faults, load variation, current harmonics	Current signals are non-invasive and widely used for motor current signature analysis.
Voltage sensors	Phase or line voltages	Supply imbalance, power quality, electrical stress	Voltage data help distinguish motor faults from supply-side disturbances.
Accelerometers	Vibration in radial/axial directions	Bearing defects, misalignment, imbalance, eccentricity	Vibration is highly sensitive to mechanical degradation and bearing-related faults.
Temperature sensors	Winding, bearing, housing, or ambient temperature	Thermal overload, insulation degradation, cooling problems	Temperature trends support degradation and thermal-stress assessment.

Table 11. Cont.

Data Source	Typical Measurement	Diagnostic Relevance	Rationale for Inclusion
Speed or encoder signal	Rotor speed or estimated speed	Slip, load condition, dynamic operating regime	Speed information improves state estimation and separates load variation from faults.
Drive/PLC/SCADA data	Load, operating mode, alarms, control commands	Contextual operating state	Contextual data reduce false alarms and improve model interpretation.
Maintenance records	Failure events, repair actions, component replacement	RUL labeling, degradation history	Historical maintenance information supports supervised prognostic modeling.

In this phase, OPC UA can be used for structured industrial communication with PLC/SCADA systems, while MQTT can support lightweight streaming of processed telemetry from edge nodes to higher-level services. Edge computing is used for low-latency preprocessing, including filtering, signal-windowing, FFT-based spectral analysis, statistical feature extraction, and anomaly pre-screening. This architecture reduces bandwidth load and avoids unnecessary centralization of high-frequency raw signals.

Semantic abstraction is required to ensure that each feature is traceable to its physical meaning, unit, sensor location, sampling frequency, and preprocessing method. For example, a vibration feature should specify the sensor axis, mounting position, window length, sampling rate, and spectral band. Similarly, a temperature variable should distinguish winding temperature, bearing temperature, housing temperature, and ambient temperature. This semantic consistency is essential for reliable hybrid modeling and later auditability.

7.5. Phase III: Hybrid Modeling and Synchronization

The third phase instantiates the analytical core of the motor Digital Twin. In accordance with F-DTLM, the scenario adopts a hybrid modeling strategy because motor health behavior is influenced by both known physical mechanisms and data-driven degradation patterns.

Physics-Based Component. Provides interpretable structure for the Digital Twin. At a simplified level, the electromagnetic behavior of an induction motor can be represented using torque-related relationships, while thermal behavior can be approximated using lumped-parameter thermal dynamics. For example, electromagnetic torque may be represented conceptually as:

$$T_e = \frac{3}{2} p \frac{L_m}{L_r} \phi_r i_s,$$

where T_e is electromagnetic torque, p is the number of pole pairs, L_m and L_r are inductances, ϕ_r is rotor flux, i_s is stator current.

Thermal behavior may be approximated as:

$$C \frac{dT}{dt} = P_{loss} - h(T - T_{amb}),$$

where C is thermal capacitance, T is temperature, P_{loss} represents total motor losses, and h is the heat transfer coefficient.

These equations are not intended as a complete high-fidelity motor model. Their purpose in this scenario is to define physically meaningful state variables, constrain inter-

pretation, and support extrapolation under operating regimes where purely data-driven models may become unreliable.

Data-Driven Component. Captures degradation patterns that are difficult to model analytically. Depending on the data availability and task definition, suitable models may include:

- support vector machines or random forests for fault classification;
- convolutional neural networks for spectral or time–frequency feature learning;
- LSTM or transformer-based models for temporal degradation modeling;
- probabilistic models or ensembles for uncertainty-aware prediction.

The RUL prediction problem is formulated as:

$$\widehat{RUL} = f_{\varphi}(x_1, x_2, \dots, x_n),$$

where x_i are extracted features (like vibration spectra, temperature trends, motor current signatures, etc.).

RUL is selected as a central output because it directly links diagnostic information to maintenance planning. Fault classification identifies the presence or type of abnormality, whereas RUL estimation supports decisions about when maintenance should be scheduled. Studies show that such models can effectively classify motor health states and predict degradation trajectories with high accuracy [64,65,70].

Model Fusion and Synchronization. The hybrid model combines physics-based predictions with data-driven corrections:

$$x_{k+1} = f_{phys}(x_k, u_k) + r_{\varphi}(x_k),$$

where r_{φ} compensates for modeling inaccuracies.

State synchronization can be implemented using recursive estimation methods. The Extended Kalman Filter (EKF) is appropriate when a state-space model is available and nonlinearities are moderate enough to be locally linearized. Its advantage is computational efficiency and interpretability, which are important for edge-compatible industrial applications. However, EKF is not the only possible choice. Unscented Kalman Filters may be preferable for stronger nonlinearities, particle filters for non-Gaussian uncertainty, and purely data-driven recurrent models when physical structure is weak or unavailable.

Thus, the choice of EKF in this scenario is methodological rather than universal. It is selected because it provides a transparent mechanism for aligning predicted motor states with real-time sensor observations while preserving compatibility with physics-based and hybrid models.

7.6. Phase IV: Operational Optimization and Control

The fourth phase translates synchronized DT outputs into maintenance and operational decisions. In this scenario, the main outputs include fault indicators, anomaly scores, RUL estimates, uncertainty bounds, and recommended maintenance actions.

The operational logic can be structured as follows:

1. If anomaly score exceeds a warning threshold, increase monitoring frequency;
2. If fault probability exceeds a diagnostic threshold, request inspection or confirmatory measurement;
3. If predicted RUL falls below a maintenance threshold, schedule service before expected failure;
4. If thermal or vibration risk exceeds a safety threshold, recommend load reduction or controlled shutdown;

5. If uncertainty is high, defer autonomous action and request human review.

This structure supports a graded response rather than a binary alarm. It is also compatible with industrial safety requirements because high-impact actions can remain human-approved even when lower-risk monitoring or recommendation tasks are automated.

The feedback mode can therefore be classified into three levels:

- advisory feedback. Dashboard warning, diagnostic report, or RUL estimate;
- semi-automated feedback. Recommended maintenance order, operator-approved set-point change;
- control feedback. Automatic load reduction or shutdown under predefined safety constraints.

Previous studies indicate that DT-based predictive maintenance can improve reliability and maintenance planning in specific implementations; however, such findings should be interpreted as background evidence rather than direct validation of F-DTLM [68,71].

For the present manuscript, the application scenario is positioned mainly at the advisory or semi-automated level. Full autonomous control would require additional safety validation, regulatory analysis, and industrial testing.

7.7. Evaluation Metrics for Future Empirical Validation

As the original experimental validation lies beyond the aims of this paper, the evaluation framework is defined as a set of metrics that should be used in future benchmarking (Table 12). This directly addresses the need for quantitative assessment without overstating the evidence provided by the present study.

Table 12. Proposed evaluation metrics for future empirical validation of the motor-monitoring F-DTLM scenario.

Evaluation Dimension	Metric	Purpose
Fault detection	Accuracy, precision, recall, F1-score	Measures ability to distinguish healthy and faulty states.
False alarms	False-positive rate, false-negative rate	Assesses reliability of alarms in industrial operation.
Fault classification	Confusion matrix, class-wise F1-score	Evaluates discrimination among fault types.
RUL prediction	MAE, RMSE, MAPE, prognostic horizon	Quantifies error and usefulness of RUL estimates.
State synchronization	State-estimation error, innovation residual, synchronization delay	Measures alignment between physical and digital states.
Uncertainty quality	Prediction interval coverage probability, calibration error	Evaluates whether uncertainty bounds are reliable.
Edge performance	Processing latency, CPU/memory load, feature-extraction time	Assesses feasibility of edge deployment.
Communication efficiency	Bandwidth reduction, packet loss tolerance, message latency	Evaluates federated data infrastructure.
Scalability	Number of monitored motors, data-stream throughput, model-update time	Assesses deployment across multiple assets.
Maintenance impact	Lead time before failure, downtime reduction, maintenance-cost reduction	Measures operational value.

Table 12. *Cont.*

Evaluation Dimension	Metric	Purpose
Robustness	Performance under missing data, noise, load variation, sensor drift	Evaluates adaptability to real industrial conditions.
Trust and governance	Model version traceability, audit completeness, explainability score	Supports accountable industrial use.

The metrics presented in Table 12 also clarify how F-DTLM should be compared with alternative approaches in future work. For example, a centralized DT implementation may be compared with a federated implementation in terms of latency, bandwidth reduction, scalability, and data-governance performance. Similarly, physics-based, data-driven, and hybrid models may be compared using prediction accuracy, explainability, computational cost, and robustness to operating-regime changes.

Prior experimental studies have reported high fault-classification performance in specific motor-monitoring settings, but these values are not directly transferable to the proposed F-DTLM scenario without dedicated benchmarking [66,70].

7.8. Scalability and Adaptability Considerations

The motor-monitoring scenario can be scaled in several directions. At the asset level, we can expand the framework from a single motor to an entire fleet, with each machine operating under vastly different loads, duty cycles, environmental conditions, and maintenance histories. Zooming out to the system level, we can seamlessly connect this motor DT to twins representing upstream and downstream equipment, such as pumps, compressors, conveyors, or entire production lines. Moving up to the enterprise level, multiple asset-level twins can pool their insights to drive holistic maintenance planning, spare-parts management, and facility-wide energy optimization.

Of course, scaling up introduces significant technical hurdles. As asset counts multiply, the DT infrastructure must juggle heterogeneous sensor configurations, fluctuating sampling rates, varying motor types, inconsistent historical maintenance records, and site-specific operational quirks. While our federated architecture mitigates this by enabling local preprocessing and distributed data ownership, it inherently demands rigorous synchronization, strict model-version control, semantic consistency, and robust governance.

Operational variability presents another major challenge. A predictive model trained perfectly under one specific load regime might fail completely when applied to another. Similarly, vibration signatures can drastically shift depending on how a machine is mounted, its coupling, foundation stiffness, or underlying process dynamics. Because of this, the DT must incorporate built-in mechanisms to actively detect model drift, recalibrate operational thresholds, update residual models, and validate RUL predictions whenever operating conditions shift.

Ultimately, the F-DTLM does not eliminate the need for rigorous, domain-specific validation. Rather, it offers a structured methodology for managing these adaptations throughout the asset's lifecycle. For example, poor data quality can systematically trigger infrastructure upgrades; shifting model residuals can prompt recalibration; and unexpected operational outcomes can force a reevaluation of KPIs or safety constraints.

7.9. Interpretation of Literature-Based Evidence

Previous study on DT-enabled predictive maintenance for electrical machines confirms that multi-sensor data fusion, hybrid modeling, and RUL-oriented analytics can drastically improve diagnostic and prognostic capabilities [66–71]. These findings firmly support the

technical plausibility of our proposed application scenario. However, these literature-based insights should not be misconstrued as direct experimental validation of the F-DTLM itself.

Therefore, the specific contribution of this section is purposely limited to the following key objectives. First, we demonstrate precisely how the F-DTLM can be instantiated for a highly relevant industrial asset. Then, we clarify the rationale behind selecting specific variables (current, voltage, vibration, temperature, speed) and techniques (RUL estimation, Extended Kalman Filter-based synchronization). Finally, we lay out the exact evaluation metrics required to conduct future empirical validation.

In this paper, the expected advantages of this instantiation remain methodological rather than experimentally proven. These advantages include a sharper scoping of motor-monitoring objectives, a more structured approach to fusing heterogeneous sensor data, the deliberate blending of physics-based and data-driven models, and an unbroken, traceable path from algorithmic prediction to real-world maintenance decisions. Confirming these advantages quantitatively remains an exciting avenue for future implementation using live experimental or industrial datasets.

7.10. Summary of the Application Scenario

This illustrative application scenario demonstrates how F-DTLM can guide the design of a Digital Twin for electrical motor monitoring and RUL-oriented predictive maintenance. The scenario maps the four F-DTLM phases onto a practical workflow: defining the motor asset and KPIs, federating sensor and control-system data, constructing a hybrid model with state synchronization, and translating predictions into maintenance recommendations or constrained control actions.

The scenario also identifies the evaluation metrics necessary for future validation, including fault-detection performance, RUL prediction error, synchronization latency, scalability, robustness, and governance traceability. Therefore, the section supports the methodological plausibility of F-DTLM while maintaining a clear distinction between illustrative instantiation and empirical validation.

8. Conclusions and Future Work

This study presented a comprehensive and structured investigation of Digital Twin technologies, covering their methodological foundations, modeling paradigms, architectural frameworks, and practical implementation aspects. Building upon this analysis, the Federated Digital Twin Lifecycle Model was proposed as a unified methodology for designing, deploying, and operating Digital Twin systems in a scalable and interoperable manner.

A key contribution of this work is the integration of three traditionally fragmented dimensions of DT research: data processing workflows, modeling paradigms, and technical architectures. The analysis demonstrated that effective DT systems cannot be reduced to a single component, such as a simulation model or a machine learning algorithm, but must instead be understood as closed-loop cyber-physical systems that continuously integrate sensing, modeling, prediction, and control [32,33,37]. The workflow described in Section 2 highlights this transformation from raw sensor data to prescriptive actuation, emphasizing the importance of feedback-driven system evolution.

The taxonomy developed in this study further clarified the landscape of DT approaches by distinguishing between levels of integration, algorithmic paradigms, and application domains. In particular, the comparative analysis confirmed that physics-based, data-driven, and hybrid models represent complementary approaches, each with distinct advantages and limitations. Physics-based models provide interpretability and physical consistency, data-driven models enable scalability and real-time inference, and hybrid approaches, such as Physics-Informed Neural Networks, offer a promising balance between these

properties [44,45,47,62]. This insight forms a critical foundation for the design of advanced DT systems.

The proposed F-DTLM framework addresses key limitations of existing DT methodologies by introducing a lifecycle-oriented approach that integrates federated data architectures, hybrid modeling, and closed-loop control mechanisms. The four-phase structure—Definition and Scoping, Data and Infrastructure Federation, Hybrid Modeling and Synchronization, and Operational Optimization and Control—provides a systematic pathway for DT implementation. The inclusion of cross-cutting layers for semantic interoperability and governance ensures that the framework aligns with emerging standards and supports trustworthy, scalable deployments.

The illustrative application scenario for electrical motor monitoring showed how F-DTLM can be instantiated for RUL-oriented predictive maintenance. Rather than constituting empirical validation, the scenario clarifies the framework's methodological applicability and identifies metrics required for future benchmarking. The application scenario indicates that combining federated data integration with hybrid modeling techniques may support improved fault detection, subject to empirical validation; enhance predictive capabilities; and enable proactive maintenance strategies. These findings are consistent with recent research emphasizing the role of DTs in predictive maintenance and asset lifecycle management [33,62]. The application scenario suggests that federated data integration combined with hybrid modeling may support fault detection, predictive analytics, and proactive maintenance planning, subject to future empirical validation.

From an architectural perspective, this work highlighted the critical role of standardization and interoperability. Reference frameworks such as ISO 23247 and RAMI 4.0, along with technologies such as the Asset Administration Shell (AAS), OPC UA, and MQTT, provide essential building blocks for integrating heterogeneous systems into a unified DT ecosystem [53–55]. The analysis also confirmed the growing importance of cloud–edge hybrid architectures, which enable low-latency processing while maintaining scalability for large-scale analytics.

Overall, this study contributes to the field by providing a holistic and reproducible framework for Digital Twin development. The F-DTLM model serves both as a methodological guide for practitioners and as a conceptual foundation for future research, bridging the gap between theoretical models, system architectures, and real-world applications.

Despite the significant advancements presented in this study, several research directions remain open and warrant further investigation.

First, experimental validation and benchmarking of the F-DTLM framework across multiple domains are necessary to assess its generalizability and performance under different operational conditions. While the electrical motor case study demonstrates its feasibility, future work should include large-scale industrial deployments, cross-domain applications (e.g., energy systems, smart cities, healthcare), and quantitative comparisons with existing DT methodologies.

Second, further research is needed to enhance hybrid modeling techniques, particularly in the integration of physics-based and data-driven components. While approaches such as PINNs show considerable promise, challenges remain in terms of training stability, computational cost, and scalability to high-dimensional systems. Developing more efficient algorithms and model reduction techniques will be essential for real-time DT applications [44,45].

Third, uncertainty quantification and explainability remain critical challenges. As DT systems increasingly rely on AI-driven models, ensuring transparency and trustworthiness becomes essential, especially in safety-critical domains. Future work should focus on

integrating explainable AI techniques with DT frameworks and improving probabilistic modeling approaches to better capture uncertainty in both data and models [37,72].

Fourth, the scalability of federated architectures requires further investigation. While federated data approaches improve interoperability and reduce latency, they introduce challenges related to data consistency, synchronization, and distributed optimization. Advanced methods for federated learning, distributed data management, and edge–cloud orchestration should be explored to address these issues.

Fifth, cybersecurity and data governance are increasingly important as DT systems become more interconnected. Ensuring secure data exchange, protecting intellectual property, and complying with regulatory requirements are essential for industrial adoption. Future research should focus on integrating security-by-design principles into DT architectures and developing robust governance frameworks.

Finally, the evolution of DT systems toward autonomous and cognitive Digital Twins represents a promising long-term direction. By incorporating advanced AI techniques such as reinforcement learning, multi-agent systems, and self-adaptive control, future DTs may achieve higher levels of autonomy and decision-making capability. This transition will require not only technological advancements but also new theoretical frameworks for understanding and managing complex, adaptive cyber–physical systems.

Digital Twin technology is rapidly evolving into a cornerstone of modern cyber–physical systems, with applications spanning industrial manufacturing, energy systems, healthcare, and smart environments. The proposed F-DTLM framework provides a structured pathway for harnessing this potential, enabling the development of scalable, interoperable, and intelligent DT systems. As research continues to advance, the integration of hybrid modeling, federated architectures, and autonomous control mechanisms will play a central role in shaping the next generation of Digital Twin technologies.

Author Contributions: Conceptualization, J.P. and D.M.; methodology, J.P. and D.M.; formal analysis, J.P. and D.M.; investigation, J.P. and D.M.; resources, J.P.; data curation, J.P. and D.M.; writing—original draft preparation, J.P. and D.M.; writing—review and editing, J.P. and D.M.; visualization, J.P. and D.M.; supervision, J.P. and D.M.; project administration, J.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Acknowledgments: The authors used ChatGPT (GPT-5; OpenAI, San Francisco, CA, USA) during manuscript preparation to assist with minor language editing and translation. All AI-generated content was thoroughly reviewed and revised by the authors, who take full responsibility for the accuracy and integrity of the final work.

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

References

1. Glaessgen, E.; Stargel, D. The digital twin paradigm for future NASA and U.S. Air Force vehicles. In Proceedings of the 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference 20th AIAA/ASME/AHS Adaptive Structures Conference 14th AIAA, Honolulu, HI, USA, 23–26 April 2012; p. 1818.
2. Lehner, C.; Padovano, A.; Zehetner, C.; Hackenberg, G. Digital Twin and Digital Thread within the Product Lifecycle Management. *Procedia Comput. Sci.* **2024**, *232*, 2875–2886. [[CrossRef](#)]

3. Bergs, T.; Gierlings, S.; Auerbach, T.; Klink, A.; Schraknepper, D.; Augspurger, T. The Concept of Digital Twin and Digital Shadow in Manufacturing. *Procedia CIRP* **2021**, *101*, 81–84. [\[CrossRef\]](#)
4. Sepasgozar, S.M.E. Differentiating Digital Twin from Digital Shadow: Elucidating a Paradigm Shift to Expedite a Smart, Sustainable Built Environment. *Buildings* **2021**, *11*, 151. [\[CrossRef\]](#)
5. Ladj, A.; Wang, Z.; Meski, O.; Belkadi, F.; Ritou, M.; Da Cunha, C. A Knowledge-Based Digital Shadow for Machining Industry in a Digital Twin Perspective. *J. Manuf. Syst.* **2021**, *58*, 168–179. [\[CrossRef\]](#)
6. Brarda, P.G.; Fernandez, G.; Ayala, N.F. Digital Twin, Digital Shadow or Digital Model? A Systematic Literature Review. In *Advances in Production Management Systems*; Springer: Cham, Switzerland, 2025; pp. 306–319.
7. Xu, H.; Wu, J.; Pan, Q.; Guan, X.; Guizani, M. A Survey on Digital Twin for Industrial Internet of Things: Applications, Technologies and Tools. *IEEE Commun. Surv. Tutor.* **2023**, *25*, 2569–2598. [\[CrossRef\]](#)
8. He, Y.; Guo, J.; Zheng, X. From Surveillance to Digital Twin: Challenges and Recent Advances of Signal Processing for Industrial Internet of Things. *IEEE Signal Process. Mag.* **2018**, *35*, 120–129. [\[CrossRef\]](#)
9. Attaran, S.; Attaran, M.; Celik, B.G. Digital Twins and Industrial Internet of Things: Uncovering Operational Intelligence in Industry 4.0. *Decis. Anal. J.* **2024**, *10*, 100398. [\[CrossRef\]](#)
10. Xiong, H.; Chu, C.; Fan, L.; Song, M.; Zhang, J.; Ma, Y.; Zheng, R.; Zhang, J.; Yang, Z.; Jiang, T. The Digital Twin Brain: A Bridge between Biological and Artificial Intelligence. *Intell. Comput.* **2023**, *2*, 0055. [\[CrossRef\]](#)
11. DIN SPEC 91345:2016-04; Reference Architecture Model Industrie 4.0 (RAMI4.0). DIN Deutsches Institut für Normung e.V.: Berlin, Germany, 2016.
12. ISO 23247-1:2021; Automation Systems and Integration—Digital Twin Framework for Manufacturing—Part 1: Overview and General Principles. International Organization for Standardization: Geneva, Switzerland, 2021.
13. IEC 63278-1:2023; Asset Administration Shell for Industrial Applications—Part 1: Asset Administration Shell Structure. International Electrotechnical Commission: Geneva, Switzerland, 2023.
14. Liu, W.; Wu, M.; Wan, G.; Xu, M. Digital Twin of Space Environment: Development, Challenges, Applications, and Future Outlook. *Remote Sens.* **2024**, *16*, 3023. [\[CrossRef\]](#)
15. Zhang, Y.; Shi, K.; Li, B. A Brief History of Digital Twin Technology. *PET Clin.* **2026**, *21*, 143–151. [\[CrossRef\]](#)
16. Maharjan, R.; Kim, N.A.; Kim, K.H.; Jeong, S.H. Transformative roles of digital twins from drug discovery to continuous manufacturing: Pharmaceutical and biopharmaceutical perspectives. *Int. J. Pharm. X* **2005**, *10*, 100409. [\[CrossRef\]](#)
17. Grieves, M.W. Digital Twins: Past, Present, and Future. In *The Digital Twin*; Springer: Cham, Switzerland, 2023; pp. 97–121.
18. Pires, F.; Cachada, A.; Barbosa, J.; Moreira, A.P.; Leitão, P. Digital Twin in Industry 4.0: Technologies, Applications and Challenges. In *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*; IEEE: Piscataway, NJ, USA, 2019; pp. 721–726. [\[CrossRef\]](#)
19. Cinar, Z.M.; Nuhu, A.A.; Zeeshan, Q.; Korhan, O. Digital Twins for Industry 4.0: A Review. In *Proceedings of the Global Joint Conf. Industrial Engineering*; Springer: Cham, Switzerland, 2019; pp. 193–203. [\[CrossRef\]](#)
20. Sadeghi, A.; Bellavista, P.; Song, W.; Yazdani-Asrami, M. Digital Twins for Condition and Fleet Monitoring of Aircraft: Toward More-Intelligent Electrified Aviation Systems. *IEEE Access* **2024**, *12*, 99806–99832. [\[CrossRef\]](#)
21. Zheng, X.; Lu, J.; Kiritsis, D. The Emergence of Cognitive Digital Twin: Vision, Challenges and Opportunities. *Int. J. Prod. Res.* **2022**, *60*, 7610–7632. [\[CrossRef\]](#)
22. Liu, Y.; Ji, T.; Guo, X.; Xu, X.; Polzer, J. A Survey of Cognitive Digital Twin and the Potential Use of LLMs. *Manuf. Lett.* **2025**, *44*, 1242–1253. [\[CrossRef\]](#)
23. Nguyen, H.; Hussein, A.; Garratt, M.A.; Abbass, H.A. Swarm Metaverse for Multi-Level Autonomy Using Digital Twins. *Sensors* **2023**, *23*, 4892. [\[CrossRef\]](#) [\[PubMed\]](#)
24. Souza, D.; Webber, T.; Wanner, E. Towards Federated, Autonomous and Cognitive Digital Twins with DAR-LING. In *2025 20th Annual System of Systems Engineering Conference (SoSE)*; IEEE: Piscataway, NJ, USA, 2025; pp. 1–6. [\[CrossRef\]](#)
25. Pronost, G.; Mayer, F.; Marche, B.; Camargo, M.; Dupont, L. Towards a Framework for the Classification of Digital Twins and Their Applications. In *2021 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC)*; IEEE: Piscataway, NJ, USA, 2021; pp. 1–7. [\[CrossRef\]](#)
26. Lindner, M.; Bank, L.; Schilp, J.; Weigold, M. Digital Twins in Manufacturing: A RAMI 4.0 Compliant Concept. *Sci* **2023**, *5*, 40. [\[CrossRef\]](#)
27. Guo, M.; Shang, X.; Zhao, J.; Shi, Y.; Sun, M.; Huang, M.; Wang, G.; Liao, L. Digital Twin Models for Architectural Heritage Conservation. *J. Build. Eng.* **2025**, *112*, 113792. [\[CrossRef\]](#)
28. Shao, G.; Frechette, S.; Srinivasan, V. An Analysis of the New ISO 23247 Series of Standards on Digital Twin Framework for Manufacturing. In *Proceedings of the ASME 2023 18th International Manufacturing Science and Engineering Conference*; ASME: New York, NY, USA, 2023; p. V002T07A001. [\[CrossRef\]](#)
29. Yi, C.; Chen, R.; Chen, J.; Li, X.; Cai, J. Introduction to Digital Twin over Wireless Networks. In *Self-Evolving Digital Twin over Wireless Networks*; Springer: Cham, Switzerland, 2026; pp. 1–21.

30. Mihai, S.; Yaqoob, M.; Hung, D.V.; Davis, W.; Towakel, P.; Raza, M.; Karamanoglu, M.; Barn, B.; Shetve, D.; Prasad, R.V.; et al. Digital Twins: A Survey on Enabling Technologies, Challenges, Trends and Future Prospects. *IEEE Commun. Surv. Tutor.* **2022**, *24*, 2255–2291. [\[CrossRef\]](#)
31. Ruah, C.; Simeone, O.; Al-Hashimi, B.M. A Bayesian Framework for Digital Twin-Based Control, Monitoring, and Data Collection in Wireless Systems. *IEEE J. Sel. Areas Commun.* **2023**, *41*, 3146–3160. [\[CrossRef\]](#)
32. Tao, F.; Zhang, H.; Liu, A.; Nee, A.Y.C. Digital Twin in Industry: State-of-the-Art. *IEEE Trans. Ind. Inform.* **2019**, *15*, 2405–2415. [\[CrossRef\]](#)
33. van Dinter, R.; Tekinerdogan, B.; Catal, C. Predictive Maintenance Using Digital Twins: A Systematic Literature Review. *Inf. Softw. Technol.* **2022**, *151*, 107008. [\[CrossRef\]](#)
34. Wu, H.; Ji, P.; Ma, H.; Xing, L. A Comprehensive Review of Digital Twin from the Perspective of Total Process: Data, Models, Networks and Applications. *Sensors* **2023**, *23*, 8306. [\[CrossRef\]](#)
35. Wang, Y.; Su, Z.; Guo, S.; Dai, M.; Luan, T.H.; Liu, Y. A Survey on Digital Twins: Architecture, Enabling Technologies, Security and Privacy, and Future Prospects. *IEEE Internet Things J.* **2023**, *10*, 14965–14987. [\[CrossRef\]](#)
36. Kessels, B.M.; Fey, R.H.B.; van de Wouw, N. Real-Time Parameter Updating for Nonlinear Digital Twins Using Inverse Mapping Models and Transient-Based Features. *Nonlinear Dyn.* **2023**, *111*, 10255–10285. [\[CrossRef\]](#)
37. Kapteyn, M.G.; Pretorius, J.V.R.; Willcox, K.E. A Probabilistic Graphical Model Foundation for Enabling Predictive Digital Twins at Scale. *Nat. Comput. Sci.* **2021**, *1*, 337–347. [\[CrossRef\]](#) [\[PubMed\]](#)
38. Selçuk, Ş.Y.; Ünal, P.; Albayrak, Ö.; Jomâa, M. A Workflow for Synthetic Data Generation and Predictive Maintenance for Vibration Data. *Information* **2021**, *12*, 386. [\[CrossRef\]](#)
39. Es-Haghi, M.S.; Anitescu, C.; Rabczuk, T. Methods for Enabling Real-Time Analysis in Digital Twins: A Literature Review. *Comput. Struct.* **2024**, *291*, 107342. [\[CrossRef\]](#)
40. Ates, C.; Bicat, D.; Yankov, R.; Arweiler, J.; Koch, R.; Bauer, H.-J. Model Predictive Evolutionary Temperature Control via Neural-Network-Based Digital Twins. *Algorithms* **2023**, *16*, 387. [\[CrossRef\]](#)
41. Kritzinger, W.; Karner, M.; Traar, G.; Henjes, J.; Sihm, W. Digital Twin in Manufacturing: A Categorical Literature Review and Classification. *IFAC-PapersOnLine* **2018**, *51*, 1016–1022. [\[CrossRef\]](#)
42. Fuller, A.; Fan, Z.; Day, C.; Barlow, C. Digital Twin: Enabling Technologies, Challenges and Open Research. *IEEE Access* **2020**, *8*, 108952–108971. [\[CrossRef\]](#)
43. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016. [\[CrossRef\]](#)
44. Karniadakis, G.E.; Kevrekidis, I.G.; Lu, L.; Perdikaris, P.; Wang, S.; Yang, L. Physics-Informed Machine Learning. *Nat. Rev. Phys.* **2021**, *3*, 422–440. [\[CrossRef\]](#)
45. Raissi, M.; Perdikaris, P.; Karniadakis, G.E. Physics-Informed Neural Networks: A Deep Learning Framework for Solving Forward and Inverse Problems Involving Nonlinear Partial Differential Equations. *J. Comput. Phys.* **2019**, *378*, 686–707. [\[CrossRef\]](#)
46. Batty, M. Digital Twins. *Environ. Plan. B Urban Anal. City Sci.* **2018**, *45*, 817–820. [\[CrossRef\]](#)
47. Thelen, A.; Zhang, X.; Fink, O.; Lu, Y.; Ghosh, S.; Youn, B.D.; Todd, M.D.; Mahadevan, S.; Hu, C.; Hu, Z. A Comprehensive Review of Digital Twin—Part 1: Modeling and Twinning Enabling Technologies. *Struct. Multidiscip. Optim.* **2022**, *65*, 331. [\[CrossRef\]](#)
48. Thelen, A.; Zhang, X.; Fink, O.; Lu, Y.; Ghosh, S.; Youn, B.D.; Todd, M.D.; Mahadevan, S.; Hu, C.; Hu, Z. A Comprehensive Review of Digital Twin—Part 2: Roles of Uncertainty Quantification and Optimization. *Struct. Multidiscip. Optim.* **2023**, *66*, 1. [\[CrossRef\]](#)
49. Cespedes-Cubides, A.S.; Gallego, L.; Sánchez, D.G.; Wagner, A.; Carlucci, S. A Review of Building Digital Twins to Improve Energy Efficiency. *Energy Inform.* **2024**, *7*, 58. [\[CrossRef\]](#)
50. Hartmann, D.; van Dinter, R.; Catal, C.; Tekinerdogan, B. Digital Twins—A Golden Age for Industrial Mathematics. *Adv. Model. Simul. Eng. Sci.* **2025**, *12*, 8. [\[CrossRef\]](#)
51. Perno, M.; Hvam, L.; Haug, A. Implementation of Digital Twins in the Process Industry: A Systematic Literature Review of Enablers and Barriers. *Comput. Ind.* **2022**, *134*, 103558. [\[CrossRef\]](#)
52. Melo, V.; de la Prieta, F.; Leitão, P. Alignment of Digital Twin Systems with the RAMI 4.0 Model Using Multi-Agent Systems. In *Service Orientation in Holonic and Multi-Agent Manufacturing*; Springer: Cham, Switzerland, 2022; pp. 23–35. [\[CrossRef\]](#)
53. Kang, M.-S.; Lee, D.-H.; Bajestani, M.S.; Kim, D.B.; Noh, S.D. Edge Computing-Based Digital Twin Framework Based on ISO 23247 for Enhancing Data Processing Capabilities. *Machines* **2025**, *13*, 19. [\[CrossRef\]](#)
54. Fett, M.; Kraft, M.; Wilking, F.; Goetz, S.; Wartzack, S.; Kirchner, E. Medium-Level Architectures for Digital Twins. *Electronics* **2024**, *13*, 1373. [\[CrossRef\]](#)
55. Abdel-Aty, T.A.; Negri, E.; Galparoli, S. Asset Administration Shell in Manufacturing. *IFAC-PapersOnLine* **2022**, *55*, 2533–2538. [\[CrossRef\]](#)
56. Zhang, J.; Ellwein, C.; Heithoff, M.; Michael, J.; Wortmann, A. Digital Twin and the Asset Administration Shell. *Softw. Syst. Model.* **2025**, *24*, 771–793. [\[CrossRef\]](#)
57. Pribiš, R.; Beño, L.; Drahoš, P. Asset Administration Shell Design Methodology Using OPC UA. *Electronics* **2021**, *10*, 2520. [\[CrossRef\]](#)

58. Wang, K.; Wang, Y.; Li, Y.; Xiao, S.; Fan, X.; Hu, L. A Review of the Technology Standards for Enabling Digital Twin. *Digit. Twin* **2024**, *2*, 4. [\[CrossRef\]](#)
59. Cho, Y.; Kim, J.; Kim, J.; Kim, H.; Lee, J. Real-Time Digital Twin Using MQTT. *Machines* **2024**, *12*, 759. [\[CrossRef\]](#)
60. Parle, D.; Sharma, G.; Anand, N.; Padgaonkar, N.; Stoddart, D.; Malley, D.J. Comparative Analysis of Digital Twin Platforms. *IEEE Access* **2024**, *12*, 180175–180197. [\[CrossRef\]](#)
61. Cutrona, V.; Bonomi, N.; Montini, E.; Ruppert, T.; Delinavelli, G.; Pedrazzoli, P. Extending Factory Digital Twins through Human Characterisation in Asset Administration Shell. *Int. J. Comput. Integr. Manuf.* **2024**, *37*, 1214–1231. [\[CrossRef\]](#)
62. Falekas, G.; Karlis, A. Digital Twin in Electrical Machine Control and Predictive Maintenance: State-of-the-Art and Future Prospects. *Energies* **2021**, *14*, 5933. [\[CrossRef\]](#)
63. Abd Wahab, N.H.; Hasikin, K.; Lai, K.W.; Xia, K.; Bei, L.; Huang, K.; Wu, X. Systematic review of predictive maintenance and digital twin technologies challenges, opportunities, and best practices. *PeerJ Comput. Sci.* **2024**, *10*, e1943. [\[CrossRef\]](#)
64. Al-Mashakbeh, A.S.; Mamchur, D.; Kalinov, A.; Zagirnyak, M. A Diagnostic of Induction Motors Supplied Using Frequency Converter Based on Current and Power Signal Analysis. *Prz. Elektrotech.* **2016**, *1*, 7–10. [\[CrossRef\]](#)
65. Mamchur, D.; Husach, S. An Analysis on Induction Motor Reliability and Lifetime Estimation Methods. *Prz. Elektrotech.* **2020**, *96*, 218–221. [\[CrossRef\]](#)
66. Phutane, P.; Chavan, S.; Pawar, H.; Mane, S.; Barde, V. Digital twin innovations for induction motor monitoring and predictive maintenance: A 2020–2025 review. *Next Res.* **2025**, *2*, 100969. [\[CrossRef\]](#)
67. Singh, R.R.; Bhatti, G.; Kalel, D.; Vairavasundaram, I.; Alsaif, F. Building a Digital Twin Powered Intelligent Predictive Maintenance System for Industrial AC Machines. *Machines* **2023**, *11*, 796. [\[CrossRef\]](#)
68. Harries, T.; Hartnoll, M.; Hafezianrazavi, M.; Meek, H.; Nassehi, A. Digital Twins for Predictive Maintenance. *Procedia CIRP* **2023**, *118*, 306–311. [\[CrossRef\]](#)
69. Dos Santos, J.F.; Tshoombe, B.K.; Santos, L.H.; Araújo, R.C.; Manito, A.R.; Fonseca, W.S.; Silva, M.O. Digital Twin-Based Monitoring System of Induction Motors Using IoT Sensors and Thermo-Magnetic Finite Element Analysis. *IEEE Access* **2023**, *11*, 1682–1693. [\[CrossRef\]](#)
70. Khan, T.; Khan, U.; Khan, A.; Mollan, C.; Morkvenaite-Vilkonciene, I.; Pandey, V. Data-Driven Digital Twin Framework for Predictive Maintenance. *Machines* **2025**, *13*, 481. [\[CrossRef\]](#)
71. Hedayati Kia, S.; Dunai, L.; Antonino-Daviu, J.A.; Razik, H. Real-Time Digital Twins for Fault Diagnosis. *Energies* **2025**, *18*, 4637. [\[CrossRef\]](#)
72. Wang, L.; Deng, T.; Zheng, Z. Explainable Modeling in Digital Twin. In *Proceedings of the 2021 Winter Simulation Conference (WSC)*; IEEE: Piscataway, NJ, USA, 2021; pp. 1–12. [\[CrossRef\]](#)

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