AI-Driven Digital Twin for Smart Manufacturing: A Dataset-Driven Predictive Maintenance Framework Using a Sample EV Car Model

Sahil Sanjay Chavan[1], Aditya Sandip Jagtap[2], Prajwal Balasaheb Kamble[3], Vinod Maruti Karche[4]

[1,2,3,4] Department of Computer Engineering, SVPM's College of Engineering, Malegaon Bk., Pune, India

Guide: **Prof. A. M. Bhunje**

Abstract - A Digital Twin keeps a virtual copy of a physical asset and uses data to mirror, analyse and anticipate the asset's condition. In smart manufacturing it is mostly discussed alongside predictive maintenance, where the goal is to act on a developing fault before it turns into downtime. Machine Learning (ML) is the usual tool for turning operating data into fault or health predictions, and electric vehicles (EVs) are an attractive test asset because the battery and motor produce rich thermal and electrical signals. In practice the work is scattered. Many studies build a predictor and stop at a score, many Digital Twin papers describe architectures without a learning component, and EV studies tend to concentrate on the battery alone. Most also assume sensors on a physical asset. This paper reviews the literature on Digital Twins, predictive maintenance, ML-based fault detection, EV battery and motor monitoring, and explainable AI, and compares the studies by asset, data source, method, twin type, visualization, real-time support, fault prediction and explainability. Three trends stand out: dependence on physical or laboratory data, loose coupling between prediction and visualization, and uneven explanation of outputs. Based on this, we outline a dataset-driven prototype in which CSV operating records of a sample EV model pass through validation, preprocessing, an ML classifier and a risk estimator, and the resulting health state is shown on a 3D twin and a dashboard. The prototype uses no physical EV or IoT sensors, and no experimental results are reported here; the evaluation measures are proposed for future validation.

Key Words: Digital Twin, Predictive Maintenance, Electric Vehicle, Machine Learning, Smart Manufacturing, Fault Detection, Industry 4.0, Dataset Replay.

1. INTRODUCTION

Factories collect far more data from their machines than they did a decade ago, and Industry 4.0 is largely the idea of putting that data to work through connected, cyber-physical production systems [11]. Maintenance is one of the first places where this pays off. A breakdown on a production line stops output, damages parts and sometimes endangers people, so knowing the state of a machine before it fails has obvious value.

Traditional maintenance policies handle this poorly. Fixing a machine after it breaks is cheap to organise but disruptive, while servicing on a fixed calendar replaces parts that still have life left and still misses faults that appear early [14]. Condition-based and predictive policies try to avoid both problems by using measured data to decide when maintenance is really needed. Surveys of the field describe a steady move towards data-driven methods, mainly ML, because they can learn fault patterns from history without a hand-built physical model [15], [16].

The Digital Twin idea adds a second layer. A twin is more than a log of sensor values: it is a virtual counterpart of an asset that is kept in step with it and can be queried, visualised and used for what-if reasoning [1], [3]. In manufacturing the term is applied to machines, production lines and whole shop floors [6], [9]. Combining a twin with ML is attractive because the twin organises and presents the asset state, while the learning model supplies the prediction that makes the state useful for maintenance decisions [4], [12].

EVs make a convenient subject. The traction battery and the drive motor are the costliest and most safety-relevant subsystems, and both are monitored through a small set of measurable quantities such as temperature, voltage, current, state of charge (SOC) and rotational speed [30], [31]. Abnormal behaviour in these quantities is a standard starting point for fault detection and for battery health work [32], [33].

Research problem. Reading across the literature, three difficulties recur. First, prediction models and twin interfaces are often developed separately, so a classifier outputs a number that nobody sees in context. Second, a large share of twin work presumes instrumented hardware, which a student laboratory or a small team rarely has. Third, predictions are frequently delivered without reasons, which limits how far a maintenance engineer can trust them [49], [50]. The question this survey asks is how far these difficulties have been addressed, and what a lightweight, dataset-driven prototype can reasonably claim given its limits.

This paper makes the following contributions:

- A structured review of maintenance methods, ML techniques, EV monitoring research and Digital Twin approaches relevant to predictive maintenance.
- A side-by-side comparison of representative studies on asset, data source, method, twin and visualization support, real-time capability, fault prediction and explainability, together with a chronological summary of 28 studies.
- A set of research gaps that we kept only where the reviewed literature supports them.
- A proposed dataset-driven framework with an EV model as the virtual asset, presented as a design proposal with no reported experimental results.

The paper is organised as follows. Sec. 2 gives background and requirements, Sec. 3 reviews existing maintenance and detection methods, Sec. 4 covers Digital Twin and AI combinations, and Sec. 5 reviews EV-specific work. Secs. 6 and 7 present the comparative analysis and literature summary, and Sec. 8 discusses research gaps. Sec. 9 describes the proposed framework, Sec. 10 its interpretation of evidence, and Sec. 11 evaluation guidelines. Secs. 12 and 13 cover open challenges and future directions, and Sec. 14 concludes.

2. BACKGROUND AND REQUIREMENTS

2.1 Digital Twins and Smart Manufacturing

The term Digital Twin is usually traced to Grieves' work on product lifecycle management, which describes a physical product, its virtual counterpart and the data link between them [1]. Aerospace researchers later used the same idea for vehicles whose health must be tracked over long service lives [2]. Definitions differ between papers, and reviews note that

the label is applied to everything from a simulation model to a fully synchronised system [7], [8].

Most descriptions share five elements. A *physical asset* exists in the real world. A *virtual representation* models its geometry, behaviour or state. A *data connection* carries measurements from asset to model. An *analysis layer* interprets the data, for example through simulation or learning. Finally, a *feedback* path returns decisions to the asset or its operators. Kritzinger et al. separate three levels of integration according to whether data flow between asset and model manually or automatically, and in one or both directions: a digital model, a digital shadow and a digital twin [6]. This distinction matters later in this paper, because a system fed from stored files sits at the weaker end of that scale.

In manufacturing, twin research has been reviewed for production management, equipment health and shop-floor control [3], [9], [10]. Tao et al. link twins with cyber-physical systems and argue that the two ideas overlap but are not the same [5].

2.2 Predictive Maintenance

Three maintenance strategies are normally contrasted. *Corrective* maintenance acts after a failure. *Preventive* maintenance acts on a schedule or usage count, regardless of condition. *Predictive* maintenance estimates the condition or remaining life of an asset from data and schedules work when the estimate says it is needed [14], [16]. The benefits usually claimed are fewer unplanned stops, better use of component life and a more informed maintenance plan. These benefits are reported in the literature for specific deployments, and our prototype makes no such claim.

A typical predictive pipeline has four stages: data acquisition, processing and feature extraction, diagnosis or prognosis, and a maintenance decision [14], [20]. Diagnosis asks whether something is wrong now, and prognosis asks how long until it is. The prototype in this paper sits at the first of these, classifying the current record rather than predicting remaining life.

2.3 Machine Learning for Predictive Maintenance

ML tasks in this field fall into a few groups. *Classification* assigns a state such as normal or faulty. *Regression* estimates a continuous quantity such as remaining useful life. *Anomaly detection* flags records that differ from a learned normal pattern, which helps when faults are rare or unlabelled [23].

Several model families are used. Decision trees give readable rules but can overfit. Random Forest averages many randomised trees and is generally more stable than a single tree [24]. Gradient-boosted trees such as XGBoost build trees one after another and are common on tabular data [25]. Support Vector Machines cope well with moderate-sized feature spaces but need careful kernel and scaling choices. Neural networks, including recurrent and convolutional forms, can learn directly from raw signals and sequences, at the price of more data and less transparency [22]. Ensemble approaches combine several learners to reduce variance. Reviews of predictive maintenance report all of these in use and do not name a single winner [15], [16], [18]. The best model depends on the dataset, the label quality and the metric, so the algorithm chosen in a project is a baseline to be tested and not a conclusion.

2.4 EV Monitoring

An EV battery management system typically measures cell or pack voltage, current and temperature, and from these estimates SOC and sometimes state of health (SOH) [30], [31], [32]. Temperature is central: abnormal heating is linked to faster ageing and to safety hazards in lithium-ion packs [31], [33]. For the drive train, motor temperature, phase current and rotational speed are the common observables, and the electrical-machine diagnosis literature treats thermal and electrical deviations as signs of developing winding, bearing or drive faults [28], [29].

In this paper we group the monitored quantities as thermal (battery and motor temperature), electrical (voltage, current, SOC) and mechanical (motor speed, vehicle speed). Which of these exist in a given dataset varies, and some may be derived or simulated rather than measured.

2.5 Dataset-Driven Digital Twins

When no physical asset or sensor network is available, a recorded dataset can stand in for the data stream. Records are read one after another and passed through the same pipeline a live feed would use. This is useful for developing and demonstrating the analysis and interface layers, and for repeatable testing, since the same sequence can be replayed.

It is not equivalent to a physical, real-time twin. The data are historical, there is no feedback path to a real asset, and the behaviour of the asset cannot be influenced by the system. In the terms of Kritzinger et al. such a setup is closer to a digital model than to a fully integrated twin [6]. We therefore write **dataset replay ≠ real-time physical Digital Twin**, and use the word "twin" for the prototype only in the sense of a virtual asset whose displayed state follows the data.

2.6 3D Digital Twin Visualization

Three things are easy to confuse. A *static 3D model* is just geometry. An *interactive 3D model* can be rotated, zoomed or annotated but does not respond to data. A *Digital Twin* in the full sense is tied to data about a specific asset and changes as that data changes [3], [7]. Visualization alone does not make something a twin, and a twin without any view of its state is hard to use.

A 3D view can help present model output. Colouring or labelling a part of the model according to the predicted state lets a user see a prediction in the context of the asset, which a raw probability does not. Whether this actually improves decisions is a usability question that the papers we reviewed rarely test, and we flag it for future evaluation.

2.7 Requirements for the Prototype

From the above we consider the following requirements reasonable for a dataset-driven prototype:

- **Data validation first:** invalid or incomplete records must not reach the model silently.
- **Transparent baseline:** a well-understood model whose behaviour can be inspected.
- **Readable output:** a small set of health states with a percentage and short reasons, rather than a bare score.
- **Honest scope:** the interface must not suggest that a physical vehicle is being monitored.
- **Repeatability:** the same input sequence must produce the same displayed sequence.
- **Low cost:** open-source tools and ordinary hardware.

3. EXISTING DETECTION AND PREDICTIVE-MAINTENANCE METHODS

Table -1 summarises the method families discussed below. We judge each family on the data it needs, whether it needs physical sensors, and whether it normally includes a twin view or an explanation.

Table -1: Families of maintenance and fault-detection methods

Family	Typical data	Physical sensors needed	Twin / 3D view	Explanation	Main limitation
Traditional maintenance	Failure and service logs	No	No	Not applicable	Reactive or wasteful
Rule / threshold	Single signals	Usually yes	Rarely	High (rules are readable)	Rigid; misses combined effects
Statistical	Time series	Usually yes	Rarely	Moderate	Assumes distributions and stationarity
Classical ML	Labelled tabular features	Data come from sensors, may be stored	Rarely	Varies by model	Needs good labels
Deep learning	Large signal or sequence data	Usually yes	Rarely	Low	Data-hungry and opaque
Battery-specific	Voltage, current, temperature, cycles	Yes, or laboratory data	Sometimes	Varies	Often cell-level and laboratory-based
Motor-specific	Current, vibration, temperature	Yes	Rarely	Varies	Needs specialised signals
IoT-based PdM	Streaming sensor data	Yes	Sometimes	Varies	Infrastructure cost
Twin-based PdM	Sensor data plus models	Usually yes	Yes	Varies	Fidelity and integration effort
Hybrid twin + ML	Sensor and/or simulated data	Often yes	Yes	Varies	Few end-to-end examples

3.1 Traditional Maintenance

Run-to-failure and calendar-based approaches need little data and little infrastructure, which explains their continued use. Their weakness is that neither tells the planner anything about the actual condition of the machine [14]. Condition-based maintenance emerged to close that gap by monitoring selected indicators, and the review by Jardine et al. remains a standard account of how diagnostics and prognostics feed such a policy [14].

3.2 Rule-Based Fault Detection

Threshold rules, such as a temperature limit or a voltage window, are simple to implement and easy to explain. They sit inside most battery management systems as protective limits [30]. A rule looks at one variable at a time, though, and fixed limits do not adapt to operating conditions, so they can trigger late or raise false alarms when the load changes. We treat rules as a useful safety net and a source of intuition for features, not as a full prognostic method.

3.3 Statistical Methods

Control charts, residual analysis and distribution-based outlier tests belong here. Surveys of anomaly detection describe statistical techniques as a mature family, useful when a normal-behaviour model can be assumed and when labelled faults are scarce [23]. The assumptions about distribution and stationarity are the usual constraint, and operating regimes that vary widely, as in a vehicle, can violate them.

3.4 Machine-Learning-Based Prediction

Systematic reviews report that ML, especially supervised classification, is the most common approach in recent predictive-maintenance work, with applications covering failure classification, remaining-life regression and anomaly detection [15], [16], [18]. Dalzochio et al. also discuss reasoning-based methods that use domain knowledge alongside learning, pointing out that purely data-driven models struggle with knowledge that is not in the data [17]. For tabular operating data, tree ensembles are frequent choices [24], [25]. The recurring cautions are label quality, class imbalance (failures are rare) and the gap between benchmark data and field data [15], [19].

3.5 Deep-Learning-Based Prediction

Deep models can learn features from raw signals instead of relying on hand-crafted ones. Zhao et al. review convolutional, recurrent and autoencoder-type models for machine health monitoring [22], and Lei et al. survey ML-based fault diagnosis and call out transfer learning and limited labelled data as open topics [21]. The gains come with costs: more data, more compute and less interpretability. Ucar et al. highlight trustworthiness as a key component of AI-based predictive maintenance for exactly this reason [19].

3.6 EV Battery Fault Detection

Battery work splits into state estimation (SOC, SOH), lifetime prognosis and fault or anomaly detection. Hannan et al. review SOC estimation for EV applications [31], Xiong et al. review SOH monitoring methods [32] and Hu et al. review lifetime prognostics [33]. For data-driven prognosis, Severson et al. predicted cell cycle life from early-cycle laboratory measurements on a set of 124 cells, using features built from the voltage curves with regularised regression [34]. For fault detection on vehicles, Hong et al. applied an LSTM network to forecast battery voltage and detect abnormal behaviour in operating data collected from real EVs [35]. The common limitation is that many studies work on single cells under controlled conditions, which differs from pack-level behaviour in a vehicle [33], [40].

3.7 EV Motor Fault Detection

The electrical-machine literature predates the EV boom. Nandi et al. reviewed condition monitoring of motors, covering bearing, stator, rotor and eccentricity faults [28], and Riera-Guasp et al. extended this to power-electronic and drive monitoring [29]. These approaches rely on specialised signals, typically current spectra or vibration, sampled at rates well above those found in a general operating-condition table. In the reviewed set we found little that combines motor diagnosis with an EV twin interface, which is why motor evidence in

our prototype can only act as a general abnormality indicator and not as a fault type.

3.8 IoT-Based Predictive Maintenance

IoT systems stream sensor data to a processing layer where ML models score it. Ayvaz and Alpay describe a predictive maintenance system for a production line that applies ML models to real-time IoT data [27]. Such systems show that live prediction is practical, but they need the sensor network, connectivity and data engineering that a small project does not have. A Digital Twin was not their focus.

3.9 Digital Twin-Based Predictive Maintenance

Errandonea et al. reviewed twins for maintenance and found that applications vary a great deal in maturity and in how much of the twin concept they truly implement [12]. Aivaliotis et al. used physics-based simulation models inside a twin to estimate the remaining useful life of equipment, with the aim of improving maintenance planning in manufacturing [13]. Tao et al. proposed a twin-driven prognostics and health-management framework for complex equipment, which is conceptual in nature [4]. A shared feature is that the twin supplies context and models, while the prediction logic differs from study to study.

3.10 Hybrid Digital Twin and Machine Learning Approaches

Hybrid approaches use twin-based simulation to complement data. Xu et al. used a twin to generate simulated data for deep transfer learning in fault diagnosis, addressing the scarcity of labelled fault data [43]. Wang et al. proposed a twin for rotating-machinery fault diagnosis in smart manufacturing [42]. Luo et al. presented a hybrid predictive maintenance approach for CNC machine tools that combines model-driven and data-driven parts within a twin [44]. Booyse et al. proposed "deep digital twins" for detection, diagnostics and prognostics [45]. These studies are the closest relatives of our work, but none of them, as far as we could establish, is a dataset-replay prototype with an EV asset and a 3D dashboard.

4. DIGITAL TWIN + AI FOR PREDICTIVE MAINTENANCE

4.1 Typical Data Flow

In the studies above the pipeline follows a similar pattern. Data from an asset (or a simulation of it) are collected and cleaned, the twin model is updated, an ML or physics-based analysis produces a condition estimate, and the result is shown to a user or used to plan maintenance [3], [4], [10]. The differences lie in what feeds the twin, how tightly the twin is synchronised with the asset, and what the analysis layer actually computes.

4.2 Synchronisation and Real-Time Operation

Synchronisation is a defining issue. Tao et al. and Qi et al. describe enabling technologies such as sensing, communication, modelling and data processing on which a synchronised twin depends [3], [10]. Fuller et al. discuss the same dependencies and add latency, security and standardisation as open problems [8]. A twin whose data arrive in batches or from stored files is a weaker form of synchronisation, and several authors advise that this be stated openly [6], [7].

4.3 Fault Prediction, Risk and Recommendations

In the ML-based twin studies, prediction usually means a fault class, an anomaly flag or a remaining-life estimate [43]–[45]. Converting a classifier probability into a risk percentage and a health band is a design choice that we found discussed more as general practice than as a studied topic, and we did not find a standard for the thresholds. Maintenance recommendations, where present, are generally framed at the level of "inspect" or "schedule service" and not as specific repair instructions. Our prototype follows the same general level of advice.

4.4 How This Differs from a Plain ML Predictor

A plain predictive-maintenance model takes features in and returns a label or a number. A twin-based system adds a persistent state representation of a particular asset, a way to show that state, and often models that can be run beyond the data (simulation, what-if). The reviewed work suggests that the added value is clearest where physical models or simulated data supplement the learning [13], [43], [44], and less clear where the twin is mainly a front end for a classifier. We think the second case is still worth building as a teaching and demonstration platform, as long as it is described that way.

4.5 Dataset-Driven Systems

We found few papers that explicitly frame a twin as a dataset-replay system. Several of the hybrid works use simulated or recorded data during development [43], [45], but they target other assets and are not packaged around a replay-driven dashboard. We therefore treat this as an area with limited published coverage, and do not conclude that no such work exists.

5. EXISTING EV DIGITAL TWIN AND PREDICTIVE-MAINTENANCE SYSTEMS

5.1 EV Digital Twin Surveys

Bhatti et al. reviewed twin technology for smart EVs, covering applications across design, manufacturing, battery management and driving [41]. Naseri et al. surveyed use cases, requirements and platforms for twins of EV battery systems [40]. Both are reviews, so they identify opportunities and challenges without reporting a new implemented detector. Their shared message is that battery twins are promising but depend on reliable data, models and computing platforms.

5.2 Battery Twins and State Estimation

Li et al. built a twin for battery systems with a cloud battery management system that carries out online SOC and SOH estimation [36]. Merkle et al. estimated the state of an e-Golf battery by combining diagnostic data from the vehicle with a twin [37]. Wu et al. discuss the fusion of models, data and AI for smart battery management and set out a twin-based perspective [38], and Wang et al. review twin applications in smart BMS [39]. These works centre on state estimation and management. Fault-risk classification and 3D presentation are not their main subject.

5.3 Battery Health and Fault Prediction

The study of Severson et al. [34] and the LSTM voltage-forecasting study [35] are examples of data-driven battery prognosis and fault detection without a twin. They use different data (laboratory cells versus real vehicle operation) and different targets (cycle life versus abnormal voltage). Both show what a dataset-based approach can do, and both show a limitation: results depend closely on the data source and may not transfer to other cells, vehicles or conditions [33].

5.4 Thermal and Motor Monitoring

Thermal behaviour is covered in battery reviews as a safety and ageing factor [31], [33], and motor temperature and

current appear in motor-monitoring surveys [28], [29]. From the sources we reviewed we could not identify a study that combines battery temperature, motor temperature and speed signals into a single classified vehicle-level health state with a 3D twin view. Existing studies tend to look at one subsystem.

5.5 Summary of the EV Landscape

The EV literature offers strong foundations in battery state estimation and some real-vehicle data studies. Twin work on EVs is dominated by reviews and battery-centred architectures. Vehicle-level, ML-driven health classification with visual twin feedback was not prominent in what we found, though we cannot rule out studies outside our reading.

6. COMPARATIVE ANALYSIS OF EXISTING WORK

6.1 Comparison of Representative Approaches

Table -2 sets representative studies side by side. The comparison is qualitative and is based on how each source describes itself. Where a source did not report a property in a form we could confirm, the cell reads "Not specified". Accuracy figures are left out on purpose: they come from different datasets and metrics and are not comparable, and in this paper they are "Not reported in the reviewed study" unless stated in the text.

Table -2: Representative approaches (qualitative)

Approach	Asset	Data Source	ML/AI Method	Digital Twin	3D Visualization	Real-Time	Fault Prediction	Explainability	Main Limitation
Tao et al. [4]	Complex equipment (generic)	Not specified	Not specified	Yes (conceptual PHM framework)	Not specified	Not specified	Framework level	Not specified	Conceptual contribution; implementation left to later work
Aivaliotis et al. [13]	Manufacturing equipment	Simulation models, sensor data	Physics-based, not primarily ML	Yes	Not specified	Not specified	Remaining useful life estimation	Model-based, inspectable in principle	Depends on accuracy of the physical models
Wang et al. [42]	Rotating machinery	Not specified	Learning-based diagnosis	Yes	Not specified	Not specified	Fault diagnosis	Not specified	Not an EV; no EV-specific signals
Xu et al. [43]	Mechanical equipment	Twin-simulated plus measured data	Deep transfer learning	Yes (twin-assisted)	Not specified	Not specified	Fault diagnosis	Low (deep model)	Relies on fidelity of simulated data
Luo et al. [44]	CNC machine tool	Not specified	Hybrid model-driven and data-driven	Yes	Not specified	Not specified	Predictive maintenance	Not specified	Machine-tool specific
Booyse et al. [45]	Not specified (generic framework)	Simulated and measured	Deep learning	Yes ("deep digital twin")	Not specified	Not specified	Detection, diagnostics, prognostics	Not specified	Opaque model; not EV-specific
Ayvaz and Alpay [27]	Production line	Real-time IoT data	ML models	No	Not specified	Yes	Failure prediction	Not specified	Needs sensor network; no twin layer
Li et al. [36]	Battery system	Measured battery data via cloud BMS	Model-based online estimation	Yes	Not specified	Online estimation	SOC/SOH estimation, not fault classes	Model-based	Estimation, not fault-risk classification
Merkle et al. [37]	EV battery (e-Golf)	Vehicle diagnostic data	Not specified	Yes	Not specified	Not specified	Battery state estimation	Not specified	Single vehicle and battery type
Hong et al. [35]	EV battery	Operating data from real EVs	LSTM network	No	No	Not specified	Voltage abnormity prognosis	Not specified	Battery only; no twin or visual layer
Severson et al. [34]	Li-ion cells	Laboratory cycling data (124 cells)	Regularised regression on engineered features	No	No	No	Cycle-life prediction	Moderate (feature-based)	Laboratory cells, not vehicle operation
Naseri et al. [40]; Bhatti et al. [41]	EV battery systems; smart EVs	Review	Review	Yes (surveyed)	Not specified	Discussed	Not applicable	Not applicable	Reviews; no new detector
Wu et al. [38]; Wang et al. [39]	Smart battery management	Review / perspective	Fusion of models, data and AI	Yes (discussed)	Not specified	Discussed	Not applicable	Not applicable	Perspective papers; implementation left open
Dalzochio et al. [17]; Ucar et al. [19]	Industrial assets (general)	Review	ML and reasoning; AI trustworthiness	Not the focus	Not specified	Discussed	Not applicable	Discussed as a challenge	Not EV-specific; reviews
Proposed prototype (this paper)	Sample EV model (virtual asset)	CSV dataset replay	Random Forest baseline	Yes, in the dataset-driven sense only	Yes (planned 3D EV view)	No (replay, not live)	Failure / abnormality risk and health class	Feature importance planned; SHAP/LIME future	Not validated; no physical asset; component-level diagnosis depends on dataset labels

6.2 Analysis

Several patterns can be read from Table -2, and we mention each only to the extent the reviewed sources support it.

First, almost every implemented study works from data that originate at a physical asset or a laboratory test, whether that is a production line [27], a vehicle battery [35], [37] or cells on a cycler [34]. The twin studies on machines add simulation to this [13], [43], but they still begin from, or are compared against, measured behaviour. None of the implemented studies we reviewed started from a stored operating-data table as the primary basis of the twin.

Second, the studies tend to specialise. Some focus on prediction without a twin [27], [34], [35]. Others build a twin around estimation or physics-based reasoning with comparatively little emphasis on a learned fault classifier [13], [36]. The hybrid studies [43]–[45] combine both, but on machine tools and rotating equipment and not on EVs.

Third, 3D visualization is the least documented dimension. In the sources we read it is not a central, evaluated feature, so we marked it "Not specified" throughout. This may reflect our reading of abstracts and method sections and should be checked against full texts, but it is consistent with the view that twin papers concentrate on modelling and data handling [7], [8].

Fourth, real-time operation appears mostly in IoT and cloud-based systems [27], [36]. This is also where infrastructure demands are highest, which fits the concern raised in the twin reviews about connectivity, latency and standardisation [8], [10].

Fifth, explanation of predictions is reported unevenly. Feature-based and physics-based methods are easier to interpret than deep networks [34], [45], and reviews of AI maintenance single out trust and transparency as unfinished topics [19], [49], [50]. We did not find a study in this set that attached feature-level explanations to a twin interface.

Finally, evidence from the EV side is mostly battery-centred and comes either from laboratory cells [34] or from limited vehicle data [35], [37]. Vehicle-level health classification that combines battery and motor quantities was not prominent. Component-level diagnosis in particular needs labels that identify which component failed, and we return to this in Sec. 8.

7. SUMMARY OF RELATED LITERATURE

Table -3 lists 28 studies in reverse chronological order. The entries are kept brief and no numerical results are copied. Studies from 2025 and 2026 are not included because we could not verify them reliably while preparing this draft; a final version should add recent work after checking it against the publisher's record.

Table -3: Summary of related literature

Sr. No.	Paper	Authors	Year	Problem	Methodology	Main Contribution	Limitation / Research Gap
1	Digital twin of electric vehicle battery systems: Comprehensive review of the use cases, requirements, and platforms [40]	Naseri et al.	2023	Unclear requirements for EV battery twins	Literature review	Survey of use cases, requirements and platforms	Review only; no implemented detector
2	Artificial Intelligence for Predictive Maintenance Applications: Key Components, Trustworthiness, and Future Trends [19]	Ucar, Karakose, Kırımça	2024	Scattered knowledge on AI-based PdM	Review	Key components, trust issues and trends	Not EV-specific; no system built
3	Towards the future of smart electric vehicles: Digital twin technology [41]	Bhatti, Mohan, Singh	2021	Role of twins across the EV lifecycle	Review	Overview of twin applications for EVs	Review; no experimental validation
4	Predictive maintenance system for production lines in manufacturing: A machine learning approach using IoT data in real-time [27]	Ayvaz, Alpay	2021	Unplanned stops on production lines	ML on real-time IoT data	Predictive system working on live data	Needs sensors; no twin
5	Application of Digital Twin in Smart Battery Management Systems [39]	Wang et al.	2021	Limits of conventional BMS	Review / perspective	Twin-based view of smart BMS	Implementation left open
6	Estimate e-Golf Battery State Using Diagnostic Data and a Digital Twin [37]	Merkle, Pöthig, Schmid	2021	Battery state is hard to observe directly	Twin with vehicle diagnostic data	Battery state estimate for a real EV	One vehicle; estimation, not fault class
7	Enabling technologies and tools for digital twin [10]	Qi et al.	2021	Fragmented twin technology landscape	Review	Technology map for twin construction	General; not maintenance-specific
8	A hybrid predictive maintenance approach for CNC machine tool driven by Digital Twin [44]	Luo et al.	2020	Maintenance of machine tools	Hybrid model/data-driven approach in a twin	Twin-driven PdM approach	Machine-tool focus
9	Digital twin for battery systems: Cloud battery management system with online state-of-charge and state-of-health estimation [36]	Li et al.	2020	Limited on-board computing for BMS	Cloud BMS with twin	Online SOC/SOH estimation	Estimation, not fault classification
10	Battery digital twins: Perspectives on the fusion of models, data and artificial intelligence for smart battery management systems [38]	Wu et al.	2020	Combining models and data for BMS	Perspective	Framework for fusing models, data and AI	Conceptual
11	Deep digital twins for detection, diagnostics and prognostics [45]	Booyse, Wilke, Heyns	2020	Detection and prognosis with limited fault data	Deep-learning twin framework	Unified detection / diagnosis / prognosis idea	Opaque; not EV
12	Digital Twin: Enabling Technologies, Challenges and Open Research [8]	Fuller et al.	2020	Open challenges for twins	Review	Challenges incl. security and standards	Not maintenance-specific
13	Characterising the Digital Twin: A systematic literature review [7]	Jones et al.	2020	Inconsistent twin definitions	Systematic review	Characterisation of twin features	No detector
14	Digital Twin for maintenance: A literature review [12]	Errandonea et al.	2020	State of twins in maintenance	Literature review	Mapping of maintenance twins	Review only
15	Digital Twin-driven smart manufacturing: Connotation, reference model, applications and research issues [9]	Lu et al.	2020	Twin use in smart manufacturing	Review with reference model	Reference model and issues	Not specific to predictive maintenance
16	Machine learning in predictive maintenance towards sustainable smart manufacturing in Industry 4.0 [18]	Çınar et al.	2020	Role of ML in PdM	Review	Overview of ML for PdM	Review
17	Machine learning and reasoning for predictive maintenance in Industry 4.0: Current status and challenges [17]	Dalzochio et al.	2020	Combining learning with knowledge	Review	Status and challenges of ML and reasoning	Review
18	Applications of machine learning to machine fault diagnosis: A review and roadmap [21]	Lei et al.	2020	Fault diagnosis with ML	Review	Roadmap incl. transfer learning	Not twin- or EV-focused
19	Battery lifetime prognostics [33]	Hu et al.	2020	Lifetime prediction of batteries	Review	Overview of prognostic methods	Review
20	Data-driven prediction of battery cycle life before capacity degradation [34]	Severson et al.	2019	Slow battery lifetime testing	Early-cycle features, regularised regression	Early life prediction on 124 cells	Laboratory cells; no twin
21	The use of Digital Twin for predictive maintenance in manufacturing [13]	Aivaliotis, Georgoulias, Chryssolouris	2019	Planning maintenance with twins	Physics-based twin models for RUL	Twin-based RUL approach	Depends on model accuracy
22	Digital twin for rotating machinery fault diagnosis in smart manufacturing [42]	Wang et al.	2019	Fault diagnosis of rotating machines	Twin plus learning	Twin-based diagnosis	Not EV
23	A Digital-Twin-Assisted Fault Diagnosis Using Deep Transfer Learning [43]	Xu et al.	2019	Scarce labelled fault data	Twin data plus transfer learning	Diagnosis using simulated data	Depends on simulation fidelity
24	Fault prognosis of battery system based on accurate voltage abnormity prognosis using long short-term memory neural networks [35]	Hong, Wang, Yao	2019	Early battery fault prognosis	LSTM on EV operating data	Voltage abnormity prognosis	No twin or visual layer
25	A systematic literature review of machine learning methods applied to predictive maintenance [15]	Carvalho et al.	2019	Which ML methods are used in PdM	Systematic review	Overview of methods and trends	Review

26	Data-driven methods for predictive maintenance of industrial equipment: A survey [16]	Zhang, Yang, Wang	2019	Data-driven PdM approaches	Survey	Categorised methods	Review
Sr. No.	**Paper**	**Authors**	**Year**	**Problem**	**Methodology**	**Main Contribution**	**Limitation / Research Gap**
27	Digital Twin in Industry: State-of-the-Art [3]	Tao et al.	2019	Maturity of industrial twins	Review	Overview of twin research	Not EV-specific
28	Digital twin driven prognostics and health management for complex equipment [4]	Tao et al.	2018	PHM of complex equipment	Conceptual framework	Twin-driven PHM idea	Conceptual

7.1 Trends

Three trends are visible in Table -3. Surveys outnumber implementations: of the 28 entries, 17 are reviews or perspective papers, which suggests that the field is still organising its concepts. Next, the implemented twin systems concentrate on machines and batteries and prefer simulation or physics models as a complement to data [13], [36], [43], [44]. Finally, the ML-only studies deliver prediction without a twin layer [27], [34], [35]. The combination studied here, a replay-driven twin with an ML classifier and a visual interface for an EV asset, sits in the space between these groups.

8. RESEARCH GAPS

We started from eight candidate gaps and checked each against the reviewed literature. Table -4 records how well each is supported. Where the evidence was thin we say so, rather than keeping the gap as a fact.

Table -4: Candidate research gaps and their support

No.	Candidate gap	Evidence in reviewed literature	Status
1	Prediction and twin visualization are treated separately	Prediction-only studies [27], [34], [35]; twin studies centred on estimation or physics models [13], [36]	Supported
2	Twins depend on physical IoT infrastructure	Real-time and cloud systems need sensing and connectivity [8], [10], [27], [36]	Supported
3	Few accessible dataset-replay prototypes	Little coverage found, but absence in our reading is not proof of absence	Partly supported
4	Datasets may lack reliable component-level labels	Label scarcity is raised in fault-diagnosis and PdM reviews [15], [21], [43]; applies to any given dataset and must be checked	Partly supported
5	Predictions lack understandable explanations	Trust and transparency are named as open issues [19], [49], [50]	Supported
6	3D visualization is not integrated with predictive analytics	3D support is rarely documented in the sources we read	Partly supported
7	Real-world validation is hard because of limited access to physical assets	EV and battery twin reviews stress data, platform and validation difficulty [33], [40], [41]	Supported
8	Dataset quality, imbalance and domain differences hurt reliability	Raised in PdM and battery reviews [15], [19], [33]	Supported

Gaps 1, 2, 5, 7 and 8 are well supported by the literature. Gaps 3, 4 and 6 rest more on what we did not find than on explicit statements in the papers, so we present them as observations that need confirmation through a wider search. Our prototype is aimed at the supported gaps and tries to avoid overclaiming in the partly supported ones.

9. PROPOSED FRAMEWORK: AI-DRIVEN DIGITAL TWIN FOR SMART MANUFACTURING

This section describes our project. It is an academic software prototype and a design proposal. It does not use a physical EV, does not use physical IoT sensors, does not control any vehicle and does not claim live telemetry. A sample EV model is the virtual asset, and CSV operating data stand in for the data stream. No experimental results are reported in this paper.

9.1 Architecture

Fig -1 depicts the pipeline: Dataset → Validation → Preprocessing → ML Engine → Risk Analyzer → Digital Twin State Manager → Dashboard. The components are described below.

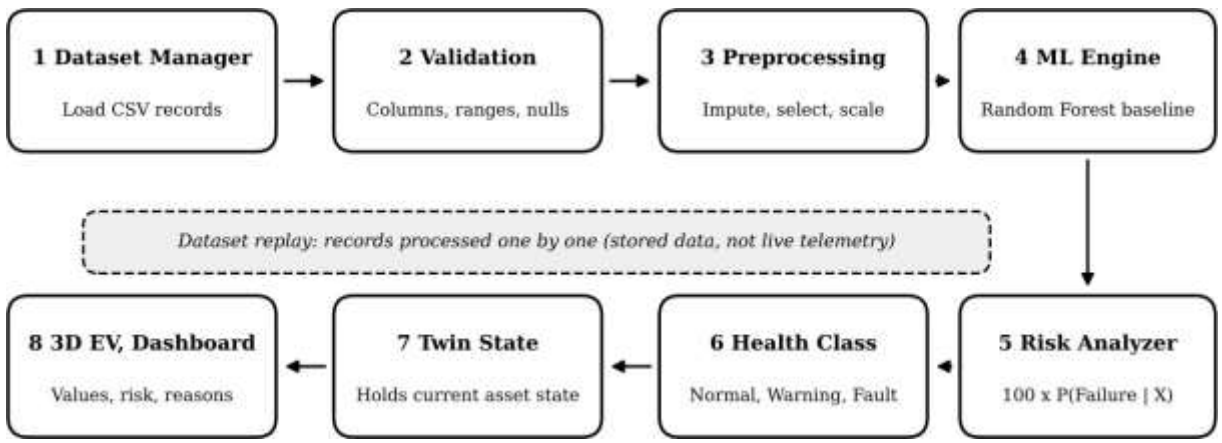


Fig -1: Proposed pipeline of the AI-Driven Digital Twin prototype, from CSV record to dashboard

- **Dataset Manager:** loads the CSV file, checks that the expected columns are present and exposes records one at a time or in batches. The dataset has not been fixed in this paper; the chosen one must be documented with its source, licence, feature definitions and label meaning.
- **Validation and Preprocessing:** handles missing values (imputation or removal, with the choice recorded), invalid records such as values outside physically plausible ranges, feature selection, scaling where a model requires it, and encoding of categorical fields. Tree-based models do not need scaling, but it stays in the pipeline so that other classifiers can be compared later.
- **ML Engine:** a Random Forest classifier serves as the baseline [24], implemented with a standard library [26]. It is a sensible starting point for tabular data because it handles nonlinear interactions and mixed feature scales, needs little tuning, outputs class probabilities and gives feature importances that can be inspected. It is a baseline and not a claim of superiority: gradient boosting [25] or other models may do better on a given dataset, and this must be tested.
- **Risk Analyzer:** converts the classifier output into a percentage (Sec. 9.2).
- **Digital Twin State Manager:** holds the current state of the virtual asset, namely the latest parameter values, risk, health class and reasons. It is the single place the dashboard and the 3D view read from.
- **Digital Twin (3D EV model):** a sample EV model whose displayed state follows the health class, for example through colour or an annotation.
- **Dashboard:** displays battery temperature, motor temperature, SOC, current, voltage, motor speed, vehicle speed, health status, failure-risk percentage, possible reasons and a general maintenance recommendation. Parameters that are absent from the chosen dataset are not displayed.

9.2 Risk Estimation and Health Classification

The failure risk of one record is computed as

$$Risk\ (\%) = 100 \times P(Failure \mid X)$$

where X is the preprocessed feature vector of that record (for instance battery temperature, motor temperature, SOC, current, voltage, motor speed and vehicle speed, to the extent the dataset provides them) and $P(Failure \mid X)$ is the probability that the classifier assigns to the failure or abnormal class. Random Forest probabilities are vote fractions and are not guaranteed to be calibrated, so the percentage should be read as a model score and not as a true likelihood unless calibration is checked.

Health classes use the prototype thresholds given in Table -5.

Table -5: Prototype health thresholds

Risk range	Health status
Risk < 40%	Normal
40% ≤ Risk < 70%	Warning
Risk ≥ 70%	Fault

These thresholds are prototype design rules. They are not automotive safety standards and have not been validated against any safety requirement. If the selected dataset has no component-level failure labels, the model predicts a general failure or abnormality state only, and the system must not be described as having verified physical component diagnosis.

9.3 Dataset Replay

To show changing conditions, records are processed in order. For each record the pipeline validates, scores and classifies it, the state manager is updated, and the dashboard and 3D model refresh. Replay controls (start, pause, speed) allow the same sequence to be repeated. This is a simulation of a changing operating situation using stored data. It is not a live feed, and the interface should say so.

9.4 Positioning

Compared with the studies in Table -2, the prototype does not introduce a new learning algorithm. Its aim is to combine a baseline ML classifier, a risk estimator, a 3D asset view and a dashboard in one dataset-driven tool that can be run without hardware. Whether this combination is useful is a question for the evaluation in Sec. 11.

10. INTERPRETATION OF EVIDENCE AND EXPLAINABILITY

The model sees all features at once, but a user wants to know which readings look unusual. The parameters shown on the dashboard can serve as supporting evidence for the predicted state:

- **Battery and motor temperature:** elevated values are recognised in the battery and motor literature as signs of stress or ageing [28], [31], [33].
- **Current and voltage:** unusually high current or voltage excursions may accompany abnormal operation or protective-limit conditions [30].
- **SOC:** extreme or rapidly changing SOC can add context to the other readings [31].
- **Motor speed and vehicle speed:** indicate the operating regime, which matters because the same temperature can be normal under heavy load and abnormal at rest.

These are indicators, not causes. A high battery temperature next to a high risk score does not show that the temperature caused the fault, so the prototype should word its reasons as "possible contributing factors". Causal claims would need support from domain studies or controlled experiments.

For explanation methods, tree-ensemble feature importance is available directly from the baseline model and gives a global view of which inputs matter most to the model. SHAP [46], [48] and LIME [47] provide per-prediction attributions and are commonly proposed for making maintenance models more transparent [49], [50]. **SHAP and LIME are not implemented in the current prototype; they are listed as future extensions.** If implemented, their outputs should be checked for stability and for agreement with domain knowledge before being shown to users as reasons.

11. EVALUATION GUIDELINES

No experimental results are presented in this paper. **The evaluation metrics are proposed for future experimental validation**, and any numbers must come from experiments on the chosen dataset.

- **Model level:** accuracy, precision, recall, F1-score, a confusion matrix and, where the task is binary, ROC-AUC, with precision-recall analysis if classes are imbalanced. Accuracy alone is misleading for rare failures, so recall and false-alarm rate on the failure class

should be reported. If records come from time sequences, splits should respect time order or group by run to avoid leakage between neighbouring records. Random Forest should be compared with at least one other model such as gradient boosting [25], and probability calibration should be examined because the risk percentage depends on it.

- **System level:** processing time per record, replay throughput, dashboard update delay and 3D rendering frame rate on typical student hardware. Behaviour on malformed input (missing columns, out-of-range values, empty files) should also be tested.
- **Explainability:** if feature importance or SHAP-style attributions are added, check consistency across runs and whether the highlighted features are plausible to a domain reader.
- **Usability:** a small task-based study in which participants interpret the dashboard with and without the 3D view, together with a standard usability questionnaire. User studies need appropriate ethical approval.
- **Datasets:** report the source, size, class balance, number of missing values and the meaning of the labels. Test on a held-out partition and, where possible, on a second dataset, since models trained on one source may not transfer [15], [33].

12. CHALLENGES AND OPEN RESEARCH PROBLEMS

Data availability and quality. Public EV operating datasets are limited, and many that exist were collected for other purposes, such as drive-cycle studies or battery ageing, and are not aligned with a vehicle-level health classification task. Missing values, sensor noise and inconsistent units affect a stored dataset as much as a live one, and any imputation changes what the model learns [15], [19].

Labels, imbalance and component-level diagnosis. Failure records are usually rare, so models can score well by predicting the majority class. Many datasets give a single abnormality label with no indication of the failing component, which limits the system to a general statement about the vehicle. Claiming that the battery or the motor specifically failed would require labels that say so [21].

Domain shift, generalisation and drift. A model trained on one vehicle, one battery chemistry or one test condition may not transfer to another [33]. In a deployed setting, ageing and changing usage would shift the data distribution over time, calling for monitoring and retraining. A replay prototype cannot show this, and we avoid suggesting that it does.

Synchronisation and fidelity. A full twin needs a trustworthy link to its asset and a model accurate enough to be useful [6], [8], [10]. Replay can test neither property. How much model fidelity is needed for a maintenance decision is an open question across the twin literature [7], [12].

3D rendering and interface load. Detailed models cost memory and frame rate, especially in a browser or on modest hardware. The balance between visual detail and responsiveness is a practical design decision that few of the reviewed papers examine.

Explainability. Attribution methods can disagree with each other and with domain expectations, and a plausible-looking reason can still be misleading [49], [50].

Cybersecurity and privacy. Connected twins expose data and control paths, which Fuller et al. identify as a concern [8]. Vehicle data can reveal location and driving behaviour. Our prototype avoids both by using stored data, but any move to live data would raise them.

Physical validation. Without a real EV or test rig, claims about real-world behaviour cannot be verified. The prototype should be presented as an academic demonstration of an analysis-and-visualization pipeline.

13. FUTURE DIRECTIONS

- **Real-time telemetry and IoT integration:** replacing replay with streamed data from a test bench or vehicle, with attention to latency and security [8], [27].
- **EV battery twin and SOH prediction:** adding battery-specific models and state-of-health estimation as in [36], [37].
- **Motor fault diagnosis:** using richer motor signals so that faults can be assigned to a component [28], [29].
- **Explainable AI:** adding SHAP or LIME with checks on stability [46]–[48].
- **Deep learning:** testing sequence models on time-ordered data where enough data exist [22], [35], at the cost of interpretability.
- **Edge and cloud deployment:** on-board or edge scoring for latency, and cloud twins for fleet-level analysis [36].
- **Multi-EV twins:** comparing vehicles and learning across a fleet.
- **Hybrid physics-based and data-driven twins:** combining model-based and learned components, as in [13], [44].
- **Higher-fidelity 3D models:** component-level geometry linked to component-level evidence.
- **Real-world validation and standard datasets:** a shared benchmark for vehicle-level health classification would make results comparable [33], [40].
- **Continuous model updating:** retraining with drift monitoring.

14. CONCLUSION

The literature reviewed here treats Digital Twins as a way to keep a virtual copy of an asset in step with data, and predictive maintenance as the use of data to decide when work is needed. ML is the dominant analysis tool, with tree ensembles common on tabular data and deep models used where signals are rich and data plentiful [15], [16], [22]. In EVs, the strongest results are on battery state estimation and prognosis, often with laboratory or limited vehicle data [32], [34], [35]. Twin studies on batteries, and on machines such as CNC tools and rotating equipment, mix physics-based models with learning [13], [36], [43], [44].

Comparing the studies showed that implementations usually rely on physical or laboratory data, that prediction and visualization are often developed separately, that 3D visualization is rarely documented, and that explanations are uneven. Not all candidate gaps were equally supported, and we marked those that rest on absence of evidence as needing confirmation.

We proposed a dataset-driven prototype that uses a sample EV model as a virtual asset and passes CSV records through validation, preprocessing, a Random Forest baseline and a risk

estimator into a 3D view and a dashboard. It is not a physical Digital Twin, it uses no real vehicle or sensors, and we report no experimental results. The next steps are to select and document a dataset, run the evaluation described in Sec. 11, and then consider explainability, real-time data and component-level diagnosis. Until those are done, the contribution of this work is the structured review, the gap analysis and the design of the framework.

ACKNOWLEDGEMENT

The authors thank the Department of Computer Engineering, SVPM's College of Engineering, Malegaon Bk., Pune, and our guide Prof. S. P. Gawade for their support and guidance.

REFERENCES

[1] M. Grieves and J. Vickers, "Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems," in *Transdisciplinary Perspectives on Complex Systems*, F.-J. Kahlen, S. Flumerfelt, and A. Alves, Eds. Cham, Switzerland: Springer, 2017, pp. 85–113.

[2] E. Glaessgen and D. Stargel, "The digital twin paradigm for future NASA and U.S. Air Force vehicles," in *Proc. 53rd AIAA/ASME/ASCE/AHS/ASC Struct., Struct. Dyn. Mater. Conf.*, Honolulu, HI, USA, 2012.

[3] F. Tao, H. Zhang, A. Liu, and A. Y. C. Nee, "Digital twin in industry: State-of-the-art," *IEEE Trans. Ind. Informat.*, vol. 15, no. 4, pp. 2405–2415, Apr. 2019.

[4] F. Tao, M. Zhang, Y. Liu, and A. Y. C. Nee, "Digital twin driven prognostics and health management for complex equipment," *CIRP Ann.*, vol. 67, no. 1, pp. 169–172, 2018.

[5] F. Tao, Q. Qi, L. Wang, and A. Y. C. Nee, "Digital twins and cyber–physical systems toward smart manufacturing and Industry 4.0: Correlation and comparison," *Engineering*, vol. 5, no. 4, pp. 653–661, 2019.

[6] W. Kritzinger, M. Karner, G. Traar, J. Henjes, and W. Sihn, "Digital twin in manufacturing: A categorical literature review and classification," *IFAC-PapersOnLine*, vol. 51, no. 11, pp. 1016–1022, 2018.

[7] D. Jones, C. Snider, A. Nassehi, J. Yon, and B. Hicks, "Characterising the digital twin: A systematic literature review," *CIRP J. Manuf. Sci. Technol.*, vol. 29, pp. 36–52, 2020.

[8] A. Fuller, Z. Fan, C. Day, and C. Barlow, "Digital twin: Enabling technologies, challenges and open research," *IEEE Access*, vol. 8, pp. 108952–108971, 2020.

[9] Y. Lu, C. Liu, K. I.-K. Wang, H. Huang, and X. Xu, "Digital twin-driven smart manufacturing: Connotation, reference model, applications and research issues," *Robot. Comput.-Integr. Manuf.*, vol. 61, Art. no. 101837, 2020.

[10] Q. Qi *et al.*, "Enabling technologies and tools for digital twin," *J. Manuf. Syst.*, vol. 58, pp. 3–21, 2021.

[11] J. Lee, B. Bagheri, and H.-A. Kao, "A cyber-physical systems architecture for Industry 4.0-based manufacturing systems," *Manuf. Lett.*, vol. 3, pp. 18–23, 2015.

[12] I. Errandonea, S. Beltrán, and S. Arrizabalaga, "Digital twin for maintenance: A literature review," *Comput. Ind.*, vol. 123, Art. no. 103316, 2020.

[13] P. Aivaliotis, K. Georgoulias, and G. Chryssolouris, "The use of digital twin for predictive maintenance in manufacturing," *Int. J. Comput. Integr. Manuf.*, vol. 32, no. 11, pp. 1067–1080, 2019.

[14] A. K. S. Jardine, D. Lin, and D. Banjevic, "A review on machinery diagnostics and prognostics implementing condition-based maintenance," *Mech. Syst. Signal Process.*, vol. 20, no. 7, pp. 1483–1510, 2006.

[15] T. P. Carvalho *et al.*, "A systematic literature review of machine learning methods applied to predictive maintenance," *Comput. Ind. Eng.*, vol. 137, Art. no. 106024, 2019.

[16] W. Zhang, D. Yang, and H. Wang, "Data-driven methods for predictive maintenance of industrial equipment: A survey," *IEEE Syst. J.*, vol. 13, no. 3, pp. 2213–2227, Sep. 2019.

[17] J. Dalzochio *et al.*, "Machine learning and reasoning for predictive maintenance in Industry 4.0: Current status and challenges," *Comput. Ind.*, vol. 123, Art. no. 103298, 2020.

[18] Z. M. Çınar *et al.*, "Machine learning in predictive maintenance towards sustainable smart manufacturing in Industry 4.0," *Sustainability*, vol. 12, no. 19, Art. no. 8211, 2020.

[19] A. Ucar, M. Karakose, and N. Kırımça, "Artificial intelligence for predictive maintenance applications: Key components, trustworthiness, and future trends," *Appl. Sci.*, vol. 14, no. 2, Art. no. 898, 2024.

[20] Y. Lei, N. Li, L. Guo, N. Li, T. Yan, and J. Lin, "Machinery health prognostics: A systematic review from data acquisition to RUL prediction," *Mech. Syst. Signal Process.*, vol. 104, pp. 799–834, 2018.

[21] Y. Lei, B. Yang, X. Jiang, F. Jia, N. Li, and A. K. Nandi, "Applications of machine learning to machine fault diagnosis: A review and roadmap," *Mech. Syst. Signal Process.*, vol. 138, Art. no. 106587, 2020.

[22] R. Zhao *et al.*, "Deep learning and its applications to machine health monitoring," *Mech. Syst. Signal Process.*, vol. 115, pp. 213–237, 2019.

[23] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, Art. no. 15, 2009.

[24] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[25] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 785–794.

[26] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[27] S. Ayvaz and K. Alpay, "Predictive maintenance system for production lines in manufacturing: A machine learning approach using IoT data in real-time," *Expert Syst. Appl.*, vol. 173, Art. no. 114598, 2021.

[28] S. Nandi, H. A. Toliyat, and X. Li, "Condition monitoring and fault diagnosis of electrical motors—A review," *IEEE Trans. Energy Convers.*, vol. 20, no. 4, pp. 719–729, Dec. 2005.

[29] M. Riera-Guasp, J. A. Antonino-Daviu, and G.-A. Capolino, "Advances in electrical machine, power electronic, and drive condition monitoring and fault detection: State of the art," *IEEE Trans. Ind. Electron.*, vol. 62, no. 3, pp. 1746–1759, Mar. 2015.

[30] H. Rahimi-Eichi, U. Ojha, F. Baronti, and M.-Y. Chow, "Battery management system: An overview of its application in the smart grid and electric vehicles," *IEEE Ind. Electron. Mag.*, vol. 7, no. 2, pp. 4–16, Jun. 2013.

[31] M. A. Hannan, M. S. H. Lipu, A. Hussain, and A. Mohamed, "A review of lithium-ion battery state of charge estimation and management system in electric vehicle applications: Challenges and recommendations," *Renew. Sustain. Energy Rev.*, vol. 78, pp. 834–854, 2017.

[32] R. Xiong, L. Li, and J. Tian, "Towards a smarter battery management system: A critical review on battery state of health monitoring methods," *J. Power Sources*, vol. 405, pp. 18–29, 2018.

[33] X. Hu, L. Xu, X. Lin, and M. Pecht, "Battery lifetime prognostics," *Joule*, vol. 4, no. 2, pp. 310–346, 2020.

[34] K. A. Severson *et al.*, "Data-driven prediction of battery cycle life before capacity degradation," *Nature Energy*, vol. 4, pp. 383–391, 2019.

[35] J. Hong, Z. Wang, and Y. Yao, "Fault prognosis of battery system based on accurate voltage abnormity prognosis using long short-term memory neural networks," *Appl. Energy*, vol. 251, Art. no. 113381, 2019.

[36] W. Li *et al.*, "Digital twin for battery systems: Cloud battery management system with online state-of-charge and state-of-health estimation," *J. Energy Storage*, vol. 30, Art. no. 101557, 2020.

[37] L. Merkle, M. Pöthig, and F. Schmid, "Estimate e-Golf battery state using diagnostic data and a digital twin," *Batteries*, vol. 7, no. 1, Art. no. 15, 2021.

[38] B. Wu, W. D. Widanage, S. Yang, and X. Liu, "Battery digital twins: Perspectives on the fusion of models, data and artificial intelligence for smart battery management systems," *Energy AI*, vol. 1, Art. no. 100016, 2020.

[39] W. Wang, J. Wang, J. Tian, J. Lu, and R. Xiong, "Application of digital twin in smart battery management systems," *Chin. J. Mech. Eng.*, vol. 34, Art. no. 57, 2021.

[40] F. Naseri *et al.*, "Digital twin of electric vehicle battery systems: Comprehensive review of the use cases, requirements, and platforms," *Renew. Sustain. Energy Rev.*, vol. 179, Art. no. 113280, 2023.

[41] G. Bhatti, H. Mohan, and R. R. Singh, "Towards the future of smart electric vehicles: Digital twin technology," *Renew. Sustain. Energy Rev.*, vol. 141, Art. no. 110801, 2021.

[42] J. Wang, L. Ye, R. X. Gao, C. Li, and L. Zhang, "Digital twin for rotating machinery fault diagnosis in smart manufacturing," *Int. J. Prod. Res.*, vol. 57, no. 12, pp. 3920–3934, 2019.

[43] Y. Xu, Y. Sun, X. Liu, and Y. Zheng, "A digital-twin-assisted fault diagnosis using deep transfer learning," *IEEE Access*, vol. 7, pp. 19990–19999, 2019.

[44] W. Luo, T. Hu, Y. Ye, C. Zhang, and Y. Wei, "A hybrid predictive maintenance approach for CNC machine tool driven by digital twin," *Robot. Comput.-Integr. Manuf.*, vol. 65, Art. no. 101974, 2020.

[45] W. Booyse, D. N. Wilke, and S. Heyns, "Deep digital twins for detection, diagnostics and prognostics," *Mech. Syst. Signal Process.*, vol. 140, Art. no. 106612, 2020.

[46] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Adv. Neural Inf. Process. Syst. 30 (NIPS)*, 2017, pp. 4765–4774.

[47] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why should I trust you?': Explaining the predictions of any classifier," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 1135–1144.

[48] S. M. Lundberg *et al.*, "From local explanations to global understanding with explainable AI for trees," *Nature Mach. Intell.*, vol. 2, pp. 56–67, 2020.

[49] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (XAI)," *IEEE Access*, vol. 6, pp. 52138–52160, 2018.

[50] A. Barredo Arrieta *et al.*, "Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Inf. Fusion*, vol. 58, pp. 82–115, 2020.