JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

AI-Driven Digital Twin Framework for Smart Manufacturing: Integrating IoT, Edge AI, and Cloud Analytics

Shyasha Gupta
Department of Computer Science and Engineering,
University Institue of Engineering,
Chandigarh University
Mohali,India
shyashagupta@gmail.com

Vasul Raj
Department of Computer Science and Engineering,
University Institue of Engineering,
Chandigarh University
Mohali,India
vasuthk1969@gmail.com

Fida Khan
Department of Computer Science and Engineering,
University Institue of Engineering,
Chandigarh University
Mohali,India
fidakhan0208@gmail.com

Amandeep Kaur
Department of Computer Science and Engineering,
University Institue of Engineering,
Chandigarh University
Mohali,India
agilldeep20@gmail.com

Abstract-Digital Twin (DT) technology is steadily merging with Industrial IoT, edge intelligence, cloud computing, and generative AI, changing how manufacturing assets are monitored and maintained. This paper presents an AI-driven Digital Twin platform for smart manufacturing that connects real-time sensing, edge-based machine learning, a stateful virtual representation of the asset, cloud analytics, and maintenance alerts grounded in evidence. The asset of interest is an electric motor with a rolling-bearing assembly. The predictive-maintenance module is validated experimentally on vibration signals from twelve physical bearings taken from the Paderborn University Bearing DataCenter. Since repeated windows from one physical unit can inflate the reported accuracy, evaluation uses bearing-wise StratifiedGroupKFold cross-validation. Random Forest, Support Vector Machine, and a compact Multilayer Perceptron are compared at four sampling rates - 3.2, 12.8, 25.6, and 64 kHz - on matched 0.5-s vibration windows. The best configuration, a Random Forest model at 12.8 kHz, reaches 90.97% mean balanced accuracy, 90.93% mean Macro F1, and 93.30% mean AUROC, and it cuts raw vibration data volume by 80% compared with the 64-kHz baseline. Building on this validated Edge AI layer, the paper defines a full IoT-edge-Digital-Twin-cloud architecture along with a grounded Generative AI layer that converts sensor readings, predictive outputs, and vetted maintenance knowledge into alerts for operators. The resulting contribution pairs an experimentally validated edge diagnostic module with a reproducible architecture for AI-driven, maintenance-focused Digital Twins in smart manufacturing.

Index Terms- *Cloud analytics, Digital Twin, Edge AI, Generative AI, Industrial Internet of Things, machine learning, predictive maintenance, smart manufacturing.*

I. INTRODUCTION

Modern smart manufacturing relies on production systems that are able to track their own condition, pass information between cyber-physical layers, and convert raw shop-floor data into decisions people can act on. The groundwork for this comes from Industry 4.0 technologies - IIoT sensing, cloud computing, AI, data analytics, and cyber-physical integration. Among them, the Digital Twin has emerged as a key point of integration, since it keeps a continuously updated digital copy of a physical asset, process, or system and ties together monitoring, simulation, analytics, and decision-making [1]-[4].

Maintenance ranks among the most important uses of Digital Twins. Reactive maintenance steps in only once something has already broken, while fixed preventive schedules may swap out components too early or too late. Predictive maintenance takes a different route: it estimates the health of equipment from operating data, so servicing choices rest on evidence instead of the calendar. A Digital Twin extends this idea by placing the condition estimates inside a persistent virtual model of the asset, one that combines sensor readings, past trends, model outputs, and maintenance context [5], [6].

The main engineering challenge is that condition-monitoring data in industry is usually generated at rates well beyond those of ordinary telemetry. Vibration monitoring of rotating machines is a good example. High-rate signals hold a lot of fault information, yet sending the raw stream nonstop to a remote cloud service pushes up bandwidth consumption, latency, storage demand, and cost. Edge AI provides a complementary approach: time-critical preprocessing and inference happen close to the asset, and the

cloud is kept for long-term storage, fleet-wide analytics, visualization, model governance, and collaboration [8], [9], [12], [20].

Another emerging direction is using Generative AI as the layer through which people reason about and receive machine information. Recent studies have looked at generative models within AI-enabled Digital Twins for creating synthetic data, building virtual environments, diagnosing faults, and supporting decisions [14], [15], [17]. Unconstrained generation, though, may yield explanations or actions that have no basis. For maintenance this calls for grounded generation, in which the language model receives structured machine state, predictive output, and vetted maintenance knowledge, and is restricted to producing operator guidance that can be traced back, rather than open-ended advice.

This paper combines these strands into one integrated design. It develops a Digital Twin architecture for an industrial motor-bearing asset, specifies a real-time IoT and edge-processing pipeline, evaluates the Edge AI predictive-maintenance component through experiments, defines cloud analytics and Digital Twin synchronization, and proposes a grounded Generative AI alert layer. The aim is not to claim that any single technology here is new. Instead, the contribution is in joining them into a coherent platform, together with an experimental look at one important edge-design trade-off: how the vibration sampling rate influences diagnostic performance and data volume when validation is done bearing-wise to resist leakage.

The remainder of the paper is organized as follows. Section II reviews the technical background and earlier work. Section III sets out the research gap, objectives, and research questions, and Section IV describes the proposed platform. The experimental methodology and setup appear in Sections V and VI, with results in Section VII. Sections VIII and IX deal with Digital Twin/cloud integration and the Generative AI layer, respectively, while Section X covers security, reliability, and interoperability. Sections XI-XIII close with the discussion, limitations, conclusions, and future directions.

II. TECHNICAL BACKGROUND AND RELATED WORK

A. Digital Twins in Smart Manufacturing

Work on Digital Twins has moved on from static virtual models toward continuously synchronized cyber-physical representations that combine live data, historical state, analytics, and increasingly AI-driven behavior. Tao et al. pointed out how Digital Twins relate to cyber-physical systems within smart manufacturing [1], and later studies set up formal architectures and approaches for integrating the lifecycle of industrial Digital Twins [2]. Qi and Tao connected Digital Twin ideas to big-data-driven manufacturing [3], and later surveys name AI, IoT, communications, simulation, and data management as the core enabling technologies [4], [16].

Recent studies have paid more attention to practical architecture. Awouda et al. proposed an IoT-based Digital Twin framework organized around Industry 5.0 priorities such as human-centricity, sustainability, and resilience [19]. Crespo-Aguado et al. developed a distributed platform that covers IoT, edge, and cloud layers for real-time Digital Twin use cases [9]. Kabir and Ray surveyed the available Digital Twin tools and underlined the need for interoperability, AI-enabled analytics, and lifecycle governance in smart manufacturing [21]. Taken together, these works support viewing a manufacturing Digital Twin as an integrated data-and-service architecture, not just a visualization layer.

B. Predictive Maintenance and Edge AI

Digital Twin-based predictive maintenance shows up in both conceptual and applied studies. Aivaliotis et al. looked at how Digital Twin concepts could support maintenance operations in manufacturing [5], and van Dinter et al. cataloged obstacles that keep recurring, including computational load, data heterogeneity, model complexity, and system integration [6]. Abdullahi et al. proposed a distributed Digital Twin framework for IIoT predictive maintenance [8], while Khan et al. built a data-driven Digital Twin framework that compares several machine-learning approaches for smart manufacturing [11]. Somu and Dasappa also described an edge-cloud IIoT framework that covers data capture, model development, and deployment [12].

Edge AI matters most when high-frequency signals have to be processed right next to the machine. Montes-Sanchez et al. demonstrated predictive-maintenance inference on edge hardware for monitoring pump degradation, which supports the feasibility of running AI locally with limited compute [20]. The real design question, then, is not only whether a model can run at the edge, but which mix of signal representation, sampling rate, feature set, and classifier gives the best balance between diagnostic quality and resource use.

C. Generative AI and Maintenance Intelligence

Generative AI has begun to take Digital Twins past prediction toward interpretation for humans and scenario generation. Dai et al. introduced an AI-DT framework for manufacturing that pairs predictive and generative components for tasks such as data augmentation, inspection, and building virtual environments [14]. Mikolajewska et al. reviewed the ways Generative AI assists fault diagnosis and predictive maintenance in Digital Twins, noting opportunities in synthetic data, anomaly analysis, and human-centered decision support [15]. More recently, Williams et al. presented a cross-domain predictive-maintenance Digital Twin that brings together MQTT messaging, machine learning, and LLM-based maintenance guidance [17]. These results suggest that merely adding an LLM no longer counts as novel; what counts is how the model is grounded, constrained, evaluated, and connected to the physical-digital state pipeline.

D. Evaluation Integrity in Bearing Fault Diagnosis

Bearing datasets are widely used in condition-monitoring research, yet the evaluation protocol can strongly affect the accuracy that gets reported. If windows from one physical bearing or recording land in both the training and test sets, the model may pick up artifacts specific to that unit rather than fault signatures that generalize. Wheat et al. showed that this kind of vibration-data leakage can considerably inflate reported bearing fault-diagnosis performance [18]. For this reason, the present study keeps the identity of each physical bearing intact through preprocessing and uses grouped cross-validation, so within any fold no bearing supplies windows to both the training and test partitions.

TABLE I. Summary of representative literature relevant to the proposed platform.

Study	Relevance to this work
Aivaliotis et al. [5]	Manufacturing DT predictive-maintenance use case.
van Dinter et al. [6]	Review identifies computational, data, and model-complexity challenges.
Abdullahi et al. [8]	Distributed IIoT Digital Twin for predictive maintenance.
Crespo-Aguado et al. [9]	Real-time IoT-edge-cloud Digital Twin platform.
Khan et al. [11]	Data-driven Digital Twin with ML for smart manufacturing.
Somu and Dasappa [12]	Integrated edge-cloud IIoT predictive-maintenance framework.
Dai et al. [14]	Predictive and Generative AI integrated in manufacturing DT.
Williams et al. [17]	MQTT, ML, Digital Twin and LLM maintenance guidance.
Wheat et al. [18]	Shows how leakage can inflate bearing-fault results.

III. RESEARCH GAP, OBJECTIVES, AND RESEARCH QUESTIONS

The literature has advanced considerably in each of Digital Twins, IIoT monitoring, edge-cloud predictive maintenance, and Generative AI taken separately. The gap that matters, therefore, is not that these technologies are missing, but that there is no integrated, experimentally grounded platform that turns high-rate industrial sensing into low-latency Edge AI state estimates, reflects those estimates in a Digital Twin, stores them for cloud analytics, and converts them into evidence-grounded maintenance communication. A second gap lies in evaluation practice: predictive-maintenance performance ought to be judged with separation at the physical-unit level instead of random window splits, and edge-design decisions should be assessed on diagnostic quality and data reduction together.

A. Research Objectives

- Develop a Digital Twin representation of industrial equipment.
- Collect real-time sensor data from connected IoT devices.
- Deploy Edge AI models in support of predictive maintenance.
- Bring in cloud services for analytics and visualization.
- Generate intelligent maintenance alerts using Generative AI.

B. Research Questions

- RQ1: How can the live state of sensors be captured and kept in sync inside a Digital Twin of industrial rotating equipment?
- RQ2: How far can IoT sensing paired with edge preprocessing reduce the amount of raw industrial data that is sent to the cloud?
- RQ3: Of Random Forest, SVM, and a compact MLP, which classifier gives the best bearing-condition classification under bearing-wise grouped validation, and at what sampling rate?
- RQ4: In what way should cloud analytics complement time-sensitive edge processing instead of duplicating it?
- RQ5: How can Generative AI produce maintenance alerts that remain consistent with sensor evidence, predictive outputs, Digital Twin state, and approved maintenance knowledge?

C. Contributions

The paper makes five contributions: (1) a layered IoT-edge-Digital-Twin-cloud architecture centered on a motor-bearing asset; (2) a reproducible, leakage-resistant assessment of bearing condition on twelve physical bearings; (3) a comparison across four sampling rates that quantifies the trade-off between vibration bandwidth and classification performance; (4) a compact state model that sends Edge AI outputs to the cloud instead of streaming raw vibration continuously; and (5) a design for grounded Generative AI maintenance alerts, in which natural-language output depends on structured industrial evidence and approved maintenance rules.

IV. PROPOSED AI-POWERED DIGITAL TWIN PLATFORM

A. Physical Equipment and IoT Sensing Layer

The asset under study is an electric motor fitted with a rolling-bearing assembly, which is referred to as Motor-01. The sensing layer is built around three complementary measurements: vibration for bearing condition, temperature for thermal state, and rotational speed for operating context. For a laboratory prototype, a high-bandwidth industrial vibration sensor is required, because the experiments identify 12.8 kHz as the preferred diagnostic sampling rate; the earlier low-cost ADXL345 concept therefore cannot meet the needs of the final edge configuration. A sensor like the IIS3DWB10IS, which is purpose-built for industrial vibration monitoring, would be a technically suitable choice for a future prototype [24].

Device management and telemetry acquisition can be handled by an ESP32-class controller, and MQTT offers a lightweight publish/subscribe channel linking the sensor node to the edge computer. The raw high-rate vibration stream is intended to remain inside the local edge network. What goes to the cloud is a compact state message made up of temperature, RPM, derived vibration descriptors, model probability, health state, and timestamps.

B. Edge AI Layer

The edge node performs windowing, signal conditioning, feature extraction, and classification. Experiments led to the choice of 12.8 kHz with 0.5-s windows, which gives 6,400 samples per inference window. The feature vector merges time-domain statistics with frequency-domain descriptors. A Random Forest classifier outputs both a discrete condition label and a fault probability, and the edge node maps the probability to a health state - NORMAL, WARNING, or CRITICAL - using configured thresholds. Those thresholds are deployment parameters and are not derived purely from the benchmark labels.

C. Digital Twin State Model

The Digital Twin is modeled as a stateful virtual entity linked to Motor-01. It stores asset identity, temperature, RPM, vibration RMS, selected spectral descriptors, fault probability, health state, edge-processing timestamp, twin-update timestamp, and alert state. Seen this way, the Twin is more than a graphical model; it acts as a synchronized information object connecting the physical asset, the analytical models, and the operator interface.

D. Cloud Analytics and Visualization

The proposed cloud path uses AWS IoT Core for secure device messaging, AWS IoT SiteWise to organize industrial time-series asset properties, AWS IoT TwinMaker to assemble the Digital Twin, and Grafana for dashboards [22], [23]. Similar industrial cloud services could be swapped in without changing the overall architecture. The cloud takes on historical storage, trend analysis, visualization across time periods, model/version metadata, fleet-level comparison, and reporting, whereas time-sensitive vibration classification remains at the edge.

E. Grounded Generative AI Layer

The Generative AI layer is placed on purpose after deterministic sensor processing and predictive inference. The model does not get open access to raw telemetry; it receives a structured evidence package containing the machine ID, current measurements, abnormal features, Edge AI prediction, fault probability, Digital Twin state, timestamps, and retrieved maintenance rules. Through the prompt, the model is told to rely only on the evidence supplied, to flag uncertainty, and not to recommend replacing a

component or shutting down unless maintenance policy clearly supports it. In this design the LLM serves as a layer for communication and summarization, and not as an autonomous maintenance authority.

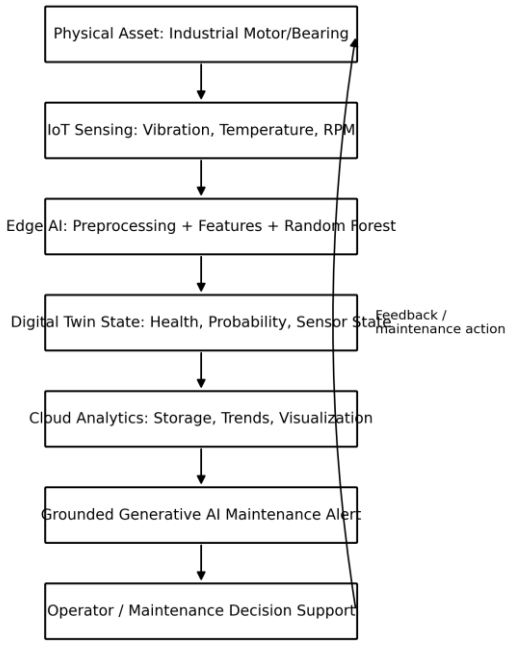

Fig. 1. Proposed end-to-end architecture linking IoT, edge processing, the Digital Twin, the cloud, and a grounded Generative AI maintenance layer.

V. METHODOLOGY

A. Dataset and Asset Selection

The predictive-maintenance experiments use the Paderborn University Bearing DataCenter benchmark [7]. The chosen subset has six healthy bearing identities (K001-K006) and six damaged ones (KA01, KA03, KA04, KI01, KI03, and KI04), where each archive holds recordings made under several operating conditions. Vibration is used as the main diagnostic signal, since it directly supports analysis of bearing condition and fits naturally with the proposed motor-bearing sensor architecture.

The original recordings are cut into matched 0.5-s windows so that all sampling-rate experiments share the same underlying time intervals. Once incomplete segments at the end are dropped, 7,669 common windows are left: 3,838 healthy and 3,831 damaged. The resulting class distribution is almost balanced, and the physical bearing ID is still kept as the grouping variable for validation.

TABLE II. Summary of the experimental dataset.

Item	Value
Physical bearing identities	12 (6 healthy, 6 damaged)
Matched windows	7,669
Healthy windows	3,838
Damaged windows	3,831
Window duration	0.5 s
Sampling rates	3.2, 12.8, 25.6, 64 kHz
Primary task	Healthy vs. damaged condition assessment

B. Resampling and Signal Conditioning

The original 64-kHz vibration recordings are analyzed at 64 kHz and are also resampled to 25.6, 12.8, and 3.2 kHz with anti-alias filtering. Before features are extracted, each 0.5-s window is mean-centered. Consequently, a window in the 12.8-kHz setting holds 6,400 samples versus 32,000 at 64 kHz, which equals 25.6 kB against 128 kB per window when stored as 32-bit floating-point values.

C. Feature Extraction

The time-domain features are RMS, standard deviation, variance, peak-to-peak amplitude, skewness, kurtosis, crest factor, impulse factor, shape factor, and clearance factor, and the frequency-domain features are dominant frequency, spectral centroid, and normalized spectral-band energy. For a mean-centered signal x with N samples, RMS is calculated as

$$RMS = \sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2} \quad (1)$$

and the crest factor is given by

$$CF = \max(|x_i|) / RMS. \quad (2)$$

Frequency-domain features come from the one-sided spectrum once the DC component has been removed. A control experiment checks whether the sampling-rate result is just an artifact of the different bandwidths on offer; it repeats the analysis with identical features limited to the shared 0-6.4-kHz band across the 12.8, 25.6, and 64 kHz rates.

D. Machine-Learning Models

Three classifiers are compared: Random Forest (RF), a Support Vector Machine (SVM) with a radial-basis-function kernel, and a compact Multilayer Perceptron (MLP). RF gives an interpretable tree-ensemble baseline that suits tabular features, SVM gives a nonlinear margin-based point of comparison, and MLP adds a lightweight neural baseline. For SVM and MLP, feature scaling is fitted inside the cross-validation pipeline, which prevents test-fold statistics from leaking into training.

E. Leakage-Resistant Validation

Validation uses five-fold StratifiedGroupKFold with bearing identity as the grouping variable. If K001, for example, lands in a test fold, none of its windows appear in the matching training set. The protocol is stricter than a random window split and is meant to estimate how well the model generalizes to bearing identities it has not seen [18]. Macro F1 is the primary metric, and balanced accuracy, precision, recall, and AUROC are also reported as means and standard deviations across folds.

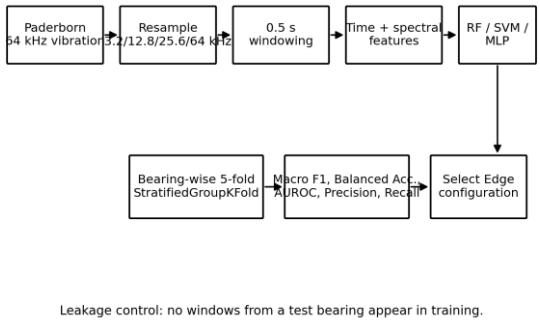

Fig. 2. Leakage-resistant preprocessing pipeline and Edge AI evaluation workflow.

F. Data-Reduction Metric

The reduction in raw data compared with the 64-kHz representation is calculated as

$$Reduction(\%) = [1 - Bytes_rate / Bytes_64k] \times 100. \quad (3)$$

This metric gives an estimate of how much high-rate vibration data could be avoided when inference runs locally and only compact state messages go to the cloud.

VI. EXPERIMENTAL SETUP

All sampling-rate experiments use the same twelve bearing identities and the same matched 0.5-s time windows, and the comparison of models is repeated at each of the four sampling rates. Hyperparameters are deliberately kept lightweight so that they reflect eventual edge deployment and not an unconstrained offline benchmark. Timing values measured on a desktop are treated only as development-stage figures and not as Raspberry Pi latency, because real edge latency can only be obtained by running on the target device.

A. Evaluation Metrics

Balanced accuracy gives every class equal weight in sensitivity. Macro F1 takes the average of the per-class F1 scores, so it holds up against mild class imbalance. Precision and recall are reported for the damaged-bearing class specifically, and AUROC reflects ranking performance over all decision thresholds. Standard deviations across folds are retained because they reveal whether performance stays steady across different unseen bearing identities.

B. Edge-Cloud State Message

The state message that goes to the cloud is designed to include the asset ID, temperature, RPM, vibration RMS, selected feature values, predicted condition, fault probability, health state, model version, inference timestamp, and edge-processing time. Raw vibration data remains local unless it is explicitly requested for diagnostics or model retraining, so the cloud can keep operational context without ingesting a continuous high-bandwidth waveform.

VII. RESULTS AND ANALYSIS

A. Best Model at Each Sampling Rate

TABLE III. Best-performing model observed at each sampling rate.

Rate	Model	Bal. Acc.	F1	AUROC	Data
3.2 kHz	MLP	60.81%	57.99%	67.22%	6.4 kB
12.8 kHz	Random Forest	90.97%	90.93%	93.30%	25.6 kB
25.6 kHz	MLP	79.24%	79.12%	80.71%	51.2 kB
64 kHz	Random Forest	78.75%	78.53%	79.86%	128.0 kB

The 12.8-kHz configuration gives the strongest overall operating point. Random Forest reaches 90.97% mean balanced accuracy, 90.93% mean Macro F1, and 93.30% mean AUROC, along with 94.21% mean precision and 87.57% mean recall. Variation between folds is considerable, about 10.16 percentage points for balanced accuracy and 10.18 for Macro F1, which indicates that some unseen bearings are harder to classify than others.

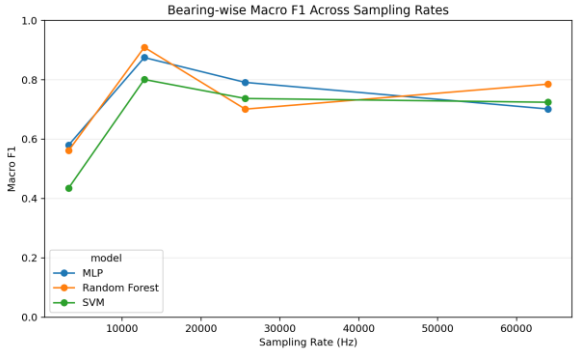

Fig. 3. Bearing-wise Macro F1 scores across the four vibration sampling rates.

B. Model Comparison at 12.8 kHz

TABLE IV. Model performance at 12.8 kHz.

Model	Bal. Acc.	F1	AUROC	Precision	Recall
MLP	87.81%	87.49%	90.17%	91.02%	84.06%
Random Forest	90.97%	90.93%	93.30%	94.21%	87.57%
SVM	82.72%	80.09%	90.30%	87.29%	76.00%

At 12.8 kHz, Random Forest does better than both SVM and MLP on the primary Macro F1 metric. MLP remains competitive, but SVM performs worse in thresholded classification even though its AUROC is above 90%, which shows why a single metric is not enough for choosing a model. RF is preferred for the proposed edge application because it combines the highest grouped-validation Macro F1 with a compact feature-based input and a simple probability output.

C. Sampling Rate and Data-Volume Trade-off

The findings show that setting the sampling rate to its maximum does not automatically give the best classifier. At 3.2 kHz, the best Macro F1 is only 57.99%, which suggests that cutting bandwidth too far removes useful diagnostic content. At 64 kHz, Random Forest reaches 78.53% Macro F1 but needs 128 kB of raw float32 data for each 0.5-s window. In contrast, the 12.8-kHz RF configuration needs just 25.6 kB per window - an 80% reduction - and still gives the best diagnostic performance overall, which makes it the preferred operating point for the proposed Edge AI design.

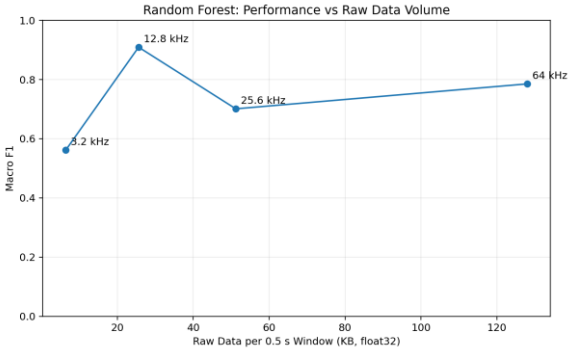

Fig. 4. Random Forest diagnostic performance plotted against raw vibration data volume.

D. Common-Band Control Experiment

TABLE V. Random Forest control results using identical 0-6.4-kHz spectral features.

Rate	Bal. Acc.	F1	AUROC
12.8 kHz	90.97%	90.93%	93.30%
25.6 kHz	84.87%	84.86%	91.14%
64 kHz	69.88%	69.13%	73.53%

The control experiment is important because the 64-kHz representation naturally contains frequency content that lower rates cannot capture. Even when RF at 12.8, 25.6, and 64 kHz is limited to identical 0-6.4-kHz spectral features, 12.8 kHz is still the strongest setting at 90.93% Macro F1, against 84.86% at 25.6 kHz and 69.13% at 64 kHz. This should not be taken as a general claim that downsampling always improves bearing diagnosis. Rather, for this specific feature representation and set of bearings, anti-alias filtering together with the 12.8-kHz rate produces a cleaner and more discriminative signal.

VIII. DIGITAL TWIN, CLOUD ANALYTICS, AND VISUALIZATION DESIGN

A. Twin Synchronization

After every edge inference cycle, a compact state update is sent to the cloud, so the Digital Twin gets the latest sensor context and prediction instead of the whole high-rate waveform. Synchronization delay can be defined as T_sync = T_twin - T_edge, where T_edge is the moment the edge state is finalized and T_twin is the moment the updated property becomes available in the Digital Twin. A full implementation should report the median, mean, and 95th-percentile synchronization delay, along with the successful-update rate and the message-loss rate.

B. Historical Analytics

Cloud analytics are intended to answer questions that one edge window cannot: How has vibration RMS trended over several days? Does fault probability rise within a certain RPM band? How often does the machine enter the WARNING state? What maintenance action followed a repeated alert? A time-series data model can hold the compact state variables and event metadata, with the Digital Twin providing asset semantics and relationships. Dashboards in the style of Grafana can then display temperature, RPM, vibration descriptors, fault probability, health state, and alert history [22], [23].

C. Edge-Cloud Division of Responsibility

The design separates responsibilities on purpose. Edge processing deals with high-rate signal conditioning and immediate condition inference, and the cloud handles historical persistence, trend analysis, model/version governance, fleet-level comparison, visualization, and reporting. This division reduces dependence on the network for time-sensitive classification while keeping the scalability and centralized visibility that cloud analytics offer.

IX. GENERATIVE AI MAINTENANCE ALERT FRAMEWORK

A. Evidence Package

The Generative AI component works from an assembled machine-state package and not from an open-ended conversation. A typical package contains the asset ID, timestamp, temperature, RPM, vibration condition, Edge AI label, fault probability, Digital Twin health state, recent trend indicators, and a retrieved set of approved maintenance rules. The model works under explicit constraints: it may not invent measurements, may not infer a root cause the evidence does not support, may not recommend replacement unless a maintenance rule permits it, and must flag uncertainty whenever the evidence conflicts.

B. Example Alert Logic

Suppose Motor-01 reports abnormal vibration, raised temperature, and a high probability of a damaged-bearing condition. The generated alert should then summarize this evidence and propose a sequence of inspection steps - for instance, checking bearing condition, lubrication, and shaft alignment - and should not state that the bearing has to be replaced unless the maintenance policy specifically calls for replacement given the observed condition. The distinction matters because fluent language should never be confused with engineering certainty.

C. Evaluation Plan

A rigorous evaluation of the GenAI layer would rely on a predefined set of scenarios that cover NORMAL, WARNING, and CRITICAL conditions, and would compare grounded and ungrounded prompts on condition correctness, severity correctness, evidence consistency, completeness of required fields, action relevance, rate of unsupported recommendations, and generation latency. A maintenance-domain expert is best placed to judge action relevance by human review, whereas evidence consistency and completeness of required fields can partly be checked with deterministic rules.

X. SECURITY, RELIABILITY, AND INTEROPERABILITY

A. Security

A Digital Twin in smart manufacturing enlarges the attack surface, since sensing, edge devices, cloud services, dashboards, and AI components all exchange operational data. For this reason, device identity management, encrypted communication, least-privilege authorization, certificate rotation, audit logging, and separation of monitoring from control channels are all necessary. GenAI outputs must never be able to actuate equipment directly without passing through an independently authorized control path.

B. Reliability and Data Quality

How good a Digital Twin is depends on how fresh and valid its source data are. Sensor failure, packet loss, clock drift, changes in mounting, and noisy measurements can leave the virtual state looking precise while it is in fact misleading for operations. The Twin should therefore carry data-quality indicators, timestamps, missing-data flags, and confidence values, and edge models should fail conservatively whenever the incoming signal lies outside the range covered by their training data.

C. Interoperability

Industrial sites often combine equipment from various vendors and generations. MQTT makes message transport simpler, but real semantic interoperability also needs consistent asset naming, units, timestamps, property definitions, and model/version metadata. Standards-based Digital Twin architectures and explicit information models are thus important for scaling past a single motor [19], [21].

XI. DISCUSSION

The empirical findings support two wider lessons for AI-powered Digital Twins. First, the validation method is part of the model itself: an early pilot with two bearings gave seemingly perfect classification, but that result had little scientific value because the model could simply exploit characteristics specific to each unit. Grouping by physical bearing gave a more realistic view of generalization and exposed large variation between folds, which supports the broader caution that random window-level splits can overstate bearing fault-diagnosis performance [18].

Second, edge design cannot be boiled down to choosing the smallest model. Signal acquisition, preprocessing, bandwidth, feature engineering, and communication volume all influence one another. The best configuration found here is neither the minimal 3.2-kHz representation nor the full 64-kHz one, but the intermediate 12.8-kHz point. This result gives a practical basis for selecting sensor hardware and edge resources, and it backs the central architectural decision to keep raw high-rate vibration local and publish only compact analytical state to the cloud.

The wider Digital Twin platform adds value by keeping context around each inference. A fault probability alone is fleeting, but a Twin can link it to machine identity, operating condition, timestamps, temperature, RPM, historical trend, maintenance events, and model version. With this context, cloud analytics and human-facing AI work from a coherent asset state and not from isolated predictions.

Generative AI is best seen as the final communication layer and not as the source of ground truth; the system of record stays the

sensed and computed Digital Twin state. Preserving this order lowers the risk of a fluent model inventing a cause or action with no support. In real manufacturing settings, this human-in-the-loop arrangement is easier to defend than allowing an LLM to change maintenance schedules or control equipment on its own.

XII. LIMITATIONS AND THREATS TO VALIDITY

The study has several limitations. First, the benchmark experiments use twelve bearing identities and not the complete set of damaged bearings in the Paderborn repository; widening the evaluation would show whether the preferred sampling rate and model choice still hold across a broader range of fault types. Second, the present task is restricted to binary healthy-versus-damaged classification. It does not estimate remaining useful life and should not be read as RUL prediction, and multi-class fault-type and severity analysis are left as separate research questions.

Third, the reported timing values come from the development environment and not from a Raspberry Pi, and they are intentionally left out of any edge-latency claims; a complete hardware study should measure feature-extraction time, inference latency, CPU utilization, RAM use, model size, and end-to-end sensor-to-decision delay on the actual target device. Fourth, the Digital Twin, cloud, and GenAI modules are specified at the architecture level but have not yet been evaluated end-to-end in a physical deployment, so the paper intentionally avoids reporting made-up synchronization or LLM-quality figures.

Finally, a domain shift may arise between the Paderborn test rig and a live motor that has a different accelerometer, mounting configuration, operating speed, load, and environmental noise, so it would be premature to treat benchmark performance as equal to deployment performance. The Generative AI layer introduces further validity concerns related to changes in model version, sensitivity to prompts, and the quality of the underlying maintenance knowledge base.

XIII. CONCLUSION AND FUTURE WORK

This paper proposes a comprehensive AI-powered Digital Twin platform for smart manufacturing that unites IIoT sensing, Edge AI, Digital Twin state management, cloud analytics, visualization, and evidence-grounded Generative AI maintenance communication. Instead of staying purely conceptual, the study experimentally evaluates the Edge AI predictive-maintenance component on twelve physical bearing identities using leakage-resistant grouped validation.

Over matched 0.5-s vibration windows, the 12.8-kHz Random Forest configuration gives the strongest overall result: 90.97% mean balanced accuracy, 90.93% mean Macro F1, and 93.30% mean AUROC, while reducing raw vibration volume by 80% compared with 64 kHz. A control experiment on a common band keeps the same ranking, which strengthens 12.8 kHz as the preferred operating point for this feature-based design.

The next experimental stage is to validate the selected configuration on a physical motor-bearing prototype with a vibration sensor of sufficient bandwidth, deploy the model on a Raspberry Pi-class edge computer, measure actual inference time and resource use, and connect compact state messages to the Digital Twin and cloud dashboard. The GenAI layer should afterward be assessed with a controlled maintenance-scenario benchmark that compares grounded and ungrounded alerts. Future work could extend the system to multi-class fault diagnosis, remaining-useful-life estimation, cross-machine model adaptation, explainable AI, federated learning, and fleet-level Digital Twins.

REFERENCES

[1] 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, doi: 10.1016/j.eng.2019.01.014.

[2] 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, doi: 10.1109/TII.2018.2873186.

[3] Q. Qi and F. Tao, "Digital Twin and Big Data Towards Smart Manufacturing and Industry 4.0: 360 Degree Comparison," IEEE Access, vol. 6, pp. 3585-3593, 2018, doi: 10.1109/ACCESS.2018.2793265.

[4] Z. Huang, Y. Shen, J. Li, M. Fey, and C. Brecher, "A Survey on AI-Driven Digital Twins in Industry 4.0: Smart Manufacturing and Advanced Robotics," Sensors, vol. 21, no. 19, Art. no. 6340, 2021, doi: 10.3390/s21196340.

[5] 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, doi: 10.1080/0951192X.2019.1686173.

[6] R. van Dinter, B. Tekinerdogan, and C. Catal, "Predictive maintenance using digital twins: A systematic literature review," Inf. Softw. Technol., vol. 151, Art. no. 107008, 2022, doi: 10.1016/j.infsof.2022.107008.

[7] C. Lessmeier, J. K. Kimotho, D. Zimmer, and W. Sextro, "Condition Monitoring of Bearing Damage in Electromechanical Drive Systems by Using Motor Current Signals of Electric Motors: A Benchmark Data Set for Data-Driven Classification," PHM Society European Conf., vol. 3, no. 1, 2016.

[8] I. Abdullahi, S. Longo, and M. Samie, "Towards a Distributed Digital Twin Framework for Predictive Maintenance in Industrial Internet of Things (IIoT)," Sensors, vol. 24, no. 8, Art. no. 2663, 2024, doi: 10.3390/s24082663.

[9] M. Crespo-Aguado, R. Lozano, F. Hernandez-Gobertti, N. Molner, and D. Gomez-Barquero, "Flexible Hyper-Distributed IoT-Edge-Cloud Platform for Real-Time Digital Twin Applications on 6G-Intended Testbeds for Logistics and Industry," Future Internet, vol. 16, no. 11, Art. no. 431, 2024, doi: 10.3390/fi16110431.

[10] N. Mchirgui, N. Quadar, H. Kraiem, and A. Lakhssassi, "The Applications and Challenges of Digital Twin Technology in Predictive Maintenance," Appl. Sci., vol. 14, no. 23, Art. no. 10933, 2024, doi: 10.3390/app142310933.

[11] T. Khan, U. Khan, A. Khan, C. Mollan, I. Morkvenaite-Vilkonciene, and V. Pandey, "Data-Driven Digital Twin Framework for Predictive Maintenance of Smart Manufacturing Systems," Machines, vol. 13, no. 6, Art. no. 481, 2025, doi: 10.3390/machines13060481.

[12] N. Somu and N. S. Dasappa, "An edge-cloud IIoT framework for predictive maintenance in manufacturing systems," Adv. Eng. Informatics, vol. 65, Art. no. 103388, 2025, doi: 10.1016/j.aei.2025.103388.

[13] S. Chen, E. T. Bekar, J. Bokrantz, and A. Skoogh, "AI-enhanced digital twins in maintenance: Systematic review, industrial challenges, and bridging research-practice gaps," J. Manuf. Syst., vol. 82, pp. 678-699, 2025, doi: 10.1016/j.jmsy.2025.07.006.

[14] D. Dai, B. Zhao, Z. Yu, P. Franciosa, and D. Ceglarek, "Generative and Predictive AI for digital twin systems in manufacturing," Front. Artif. Intell., vol. 8, Art. no. 1655470, 2025, doi: 10.3389/frai.2025.1655470.

[15] E. Mikolajewska, D. Mikolajewski, T. Mikolajczyk, and T. Paczkowski, "Generative AI in AI-Based Digital Twins for Fault Diagnosis for Predictive Maintenance in Industry 4.0/5.0," Appl. Sci., vol. 15, no. 6, Art. no. 3166, 2025, doi: 10.3390/app15063166.

[16] M. B. Al Zami, S. Shaon, V. K. Quy, and D. C. Nguyen, "Digital Twin in Industries: A Comprehensive Survey," IEEE Access, vol. 13, pp. 47291-47336, 2025, doi: 10.1109/ACCESS.2025.3551532.

[17] J. Williams, G. Jones, T. Berg, L. Birt, A. Stowe, and X. Li, "Cross-domain digital twin architecture for predictive maintenance via machine learning and Large Language Models," Comput. Ind. Eng., vol. 215, Art. no. 111914, 2026, doi: 10.1016/j.cie.2026.111914.

[18] L. Wheat, M. V. Mohrenschildt, S. Habibi, and D. Al-Ani, "Impact of Data Leakage in Vibration Signals Used for Bearing Fault Diagnosis," IEEE Access, vol. 12, pp. 169879-169895, 2024, doi: 10.1109/ACCESS.2024.3497716.

[19] A. Awouda, E. Traini, G. Bruno, and P. Chiabert, "IoT-Based Framework for Digital Twins in the Industry 5.0 Era," Sensors, vol. 24, no. 2, Art. no. 594, 2024, doi: 10.3390/s24020594.

[20] J. M. Montes-Sanchez, Y. Uwate, Y. Nishio, S. Vicente-Diaz, and A. Jimenez-Fernandez, "Predictive Maintenance Edge Artificial Intelligence Application Study Using Recurrent Neural Networks for Early Aging Detection in Peristaltic Pumps," IEEE Trans. Rel., vol. 74, no. 3, pp. 3730-3744, Sep. 2025, doi: 10.1109/TR.2024.3488963.

[21] M. R. Kabir and S. Ray, "Digital Twin Tools for Smart Manufacturing: A Paradigm Shift for Industry 4.0," IEEE Open J. Ind. Electron. Soc., vol. 6, pp. 1756-1770, 2025, doi: 10.1109/OJIES.2025.3628531.

[22] Amazon Web Services, "AWS IoT TwinMaker Documentation," AWS Documentation, 2026. [Online]. Available: https://docs.aws.amazon.com/iot-twinmaker/. [Accessed: Aug. 17, 2026].

[23] Amazon Web Services, "AWS IoT SiteWise and Amazon Managed Grafana Integration," AWS Documentation, 2026. [Online]. Available: https://docs.aws.amazon.com/iot-sitewise/. [Accessed: Aug. 17, 2026].

[24] STMicroelectronics, "IIS3DWB10IS: Advanced Vibration Sensor for Predictive Maintenance," 2026. [Online]. Available: https://www.st.com/en/mems-and-sensors/iis3dwb10is.html. [Accessed: Aug. 17, 2026].