A Digital Twin-Driven Deep Learning Framework for Real-Time Fault Prediction and Predictive Maintenance in Smart Manufacturing

Dr. Maithili S. Deshmukh[1], Dr. A. S. Alvi[2]

[1]Associate Professor & Department of Information Technology, Prof. Ram Meghe Institute of Technology & Research, Badnera, India

[2]Professor & Department of Information Technology, Prof. Ram Meghe Institute of Technology & Research, Badnera, India

DOI: https://doi.org/10.51584/IJRIAS.2026.11080061

Received: 06 August 2026; Accepted: 27 August 2026; Published: 05 September 2026

ABSTRACT

Predictive maintenance is a central requirement of smart manufacturing because unexpected equipment failures can interrupt production, increase maintenance cost, and reduce asset availability. Digital Twin technology provides a mechanism for maintaining a synchronized digital representation of a physical asset, while deep learning can learn complex temporal patterns from multivariate industrial sensor streams. This paper proposes a Digital Twin-driven deep learning framework for real-time fault prediction and predictive maintenance. The framework combines Industrial Internet of Things sensing, an edge-to-cloud data layer, a continuously updated Digital Twin state, temporal deep learning, Remaining Useful Life estimation, and explainable artificial intelligence. A Transformer-based temporal model is proposed as the principal predictor, with an LSTM model retained as a baseline for comparative evaluation. The Digital Twin acts as the integration layer between physical equipment and analytical models, enabling the prediction model to use current operating conditions and historical degradation patterns. SHAP-based explanations are incorporated to identify sensor variables contributing to individual predictions and to support maintenance engineers in interpreting model outputs. NASA C-MAPSS is proposed as a reproducible benchmark for the initial experimental evaluation, with later extension to industrial bearing or machine datasets. The manuscript defines the research gap, system architecture, data pipeline, mathematical formulation, experimental protocol, evaluation metrics, and ablation strategy. Because this document is a research manuscript draft, numerical performance results are intentionally not fabricated; they must be populated after implementation. The proposed framework provides a rigorous basis for evaluating whether Digital Twin synchronization and explainability can improve fault prediction and RUL estimation over standalone deep-learning models.

Keywords: Digital Twin, Predictive Maintenance, Deep Learning, Transformer, Remaining Useful Life, Industry 5.0, Explainable AI

INTRODUCTION

Manufacturing systems are increasingly instrumented with sensors, controllers, communication networks, and analytics platforms. This transformation creates an opportunity to shift maintenance from reactive or calendar-based practices toward condition-based and predictive strategies. Predictive maintenance uses observations of equipment condition to identify abnormal behavior and estimate future degradation.

Digital Twin technology provides a complementary mechanism by maintaining a digital representation of a physical asset and linking models, data, connections, and services. In manufacturing, Digital Twins have been proposed for monitoring, prognostics, health management, optimization, and cyber-physical integration [1,2].

Deep learning is attractive for predictive maintenance because sensor signals are multivariate, noisy, nonlinear, and temporal. LSTM networks can model sequential dependencies, whereas Transformer architectures use attention mechanisms to capture relationships across a sequence [3]. Recent surveys identify data quality, interpretability, computational cost, and generalization as important challenges [4,5].

The research problem addressed here is the design of an integrated analytical loop in which the Digital Twin provides an updated asset state, the deep-learning model predicts degradation or failure, and an explainability layer communicates the factors behind the prediction. The main contributions are a layered architecture, a temporal deep-learning formulation, SHAP-based explanation, and a reproducible evaluation protocol.

Related Work

Digital Twins have progressed from static virtual representations toward adaptive systems connected to live operational data. Recent reviews of Digital Twin-enabled predictive maintenance identify opportunities for real-time monitoring and prediction while highlighting interoperability, validation, uncertainty, data governance, and model maturity as continuing challenges [5,6].

Remaining Useful Life estimation is a core prognostics task. NASA's C-MAPSS turbofan degradation simulation dataset is a widely used benchmark in which multiple sensor channels characterize simulated degradation trajectories [7]. Deep-learning studies use CNN, RNN, LSTM, autoencoder, attention, and Transformer architectures for learning health indicators and RUL mappings [4].

The Transformer was introduced as an attention-based architecture that removes recurrence and convolution from its core sequence mechanism [3]. For industrial time series, attention can help the model assign different importance to observations and sensor variables.

Explainability is important because predictive-maintenance decisions affect physical assets and maintenance schedules. SHAP provides a unified additive feature-attribution framework for interpreting individual predictions [8]. Explainable predictive-maintenance research emphasizes trust and operational adoption as important considerations [9].

Research Gap

Many predictive-maintenance studies evaluate a deep-learning model on sensor data without explicitly modeling the Digital Twin as an active synchronization and state-management layer. Digital Twin reviews identify AI integration as promising but also report gaps in standardization, functional requirements, validation, and real-time deployment [5,6]. In addition, accuracy alone does not establish operational usefulness; maintenance personnel need explanations and confidence information. This work therefore treats the Digital Twin state as an explicit model input and the explanation as an inference-time output.

Problem Statement

Let a physical asset generate a multivariate sensor vector x_t at time t. The objective is to estimate a fault state y_t and/or remaining useful life r_t from a temporal window $X_t=\{x_{(t-w+1)},...,x_t\}$, while incorporating a Digital Twin state vector d_t representing synchronized operating and health information. The prediction function is $\hat{y}_t, \hat{r}_t = f_\theta(X_t,d_t)$. The research question is whether explicit Digital Twin state information and temporal attention improve predictive performance and interpretability compared with standalone deep-learning models.

Objectives

• Develop a layered Digital Twin architecture for equipment health monitoring.

• Construct a multivariate temporal data pipeline from sensor observations to model-ready sequences.

• Develop a Transformer-based model for fault prediction and RUL estimation.

• Compare the proposed model with LSTM and classical machine-learning baselines.

• Integrate SHAP explanations for feature-level interpretation.

• Define an ablation study to quantify the contribution of Digital Twin state information.

• Establish a reproducible evaluation protocol using a public prognostics dataset.

Proposed Framework

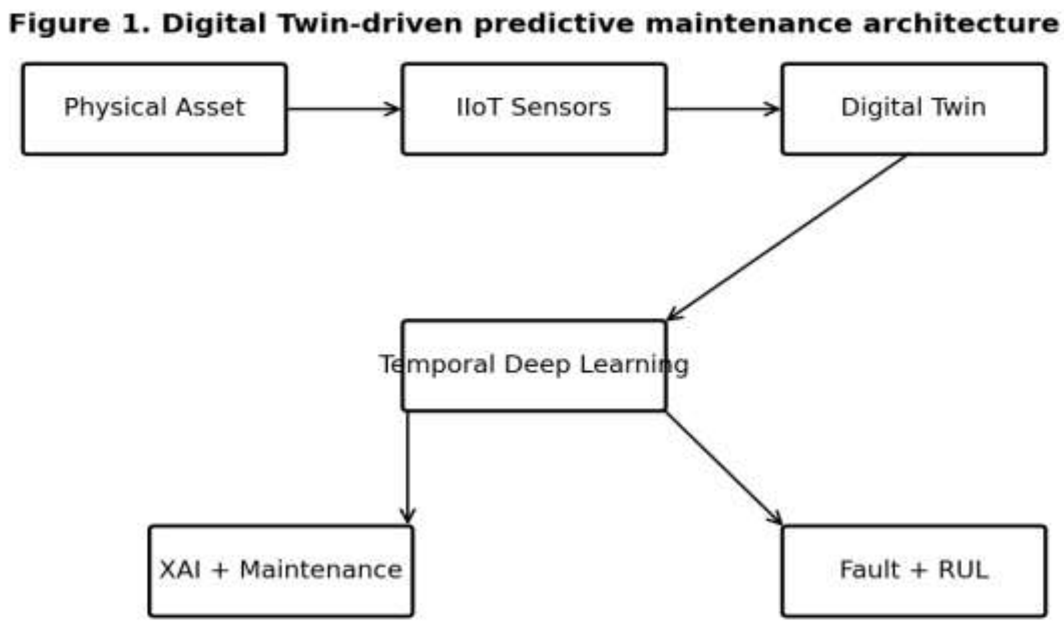

Figure 1. Proposed Digital Twin-driven predictive maintenance architecture.

Digital Twin state representation

The Digital Twin state vector is defined as d_t=[s_t,h_t,o_t,m_t], where s_t represents current sensor-derived state variables, h_t represents estimated health indicators, o_t represents operating context, and m_t represents model or metadata attributes. The synchronization process preserves timestamps, asset identity, sensor identity, sampling frequency, and data-quality indicators.

Data acquisition and preprocessing

Sensor streams are aligned to a common time base. Missing values are handled using a documented imputation strategy, while communication anomalies are flagged. Continuous variables are normalized using training-set statistics. A sliding window of length w is constructed. For RUL experiments, targets are derived from the degradation trajectory and end-of-life condition.

Temporal deep-learning model

The principal model uses a compact Transformer encoder. Each sensor observation is projected into an embedding space, augmented with temporal positional information, and passed through multi-head self-attention and feed-forward layers. The final representation is passed to two heads: classification for fault state and regression for RUL. An LSTM receives the same windows as a baseline.

Explainability layer

SHAP values are computed for sensor features or aggregated sensor-time features. Positive contributions indicate evidence pushing the prediction toward higher fault probability or shorter RUL, while negative contributions

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue VIII August 2026

push in the opposite direction. Explanations should be validated by maintenance experts rather than interpreted as causal proof.

WORKFLOW

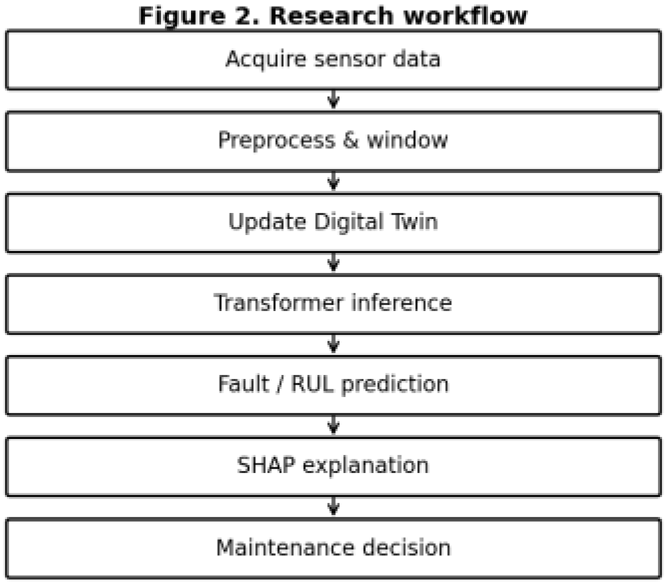

Figure 2. End-to-end research workflow.

Algorithm

1. Initialize the Digital Twin for the selected asset.

2. Acquire timestamped sensor observations.

3. Validate, synchronize, normalize, and window observations.

4. Update the Digital Twin state.

5. Concatenate sensor representations with Digital Twin state features.

6. Run Transformer inference to obtain fault probability and RUL estimate.

7. Generate SHAP feature contributions.

8. Trigger a maintenance alert when predicted risk exceeds the predefined threshold.

9. Store prediction, explanation, and twin state for subsequent evaluation.

Mathematical Formulation

Let $X_t \in R^{(w\times p)}$ denote a sequence of w observations with p sensor variables and $D_t \in R^q$ denote the Digital Twin state. The model input is $Z_t = \mathrm{Embed}(X_t) \oplus \mathrm{Project}(D_t)$, where $\oplus$ denotes concatenation. The Transformer encoder produces $H_t = \mathrm{Transformer}(Z_t)$. A pooled representation z_t is mapped to a fault probability $P_t=\sigma(W_c z_t+b_c)$ and RUL estimate $\hat{r}_t=W_r z_t+b_r$.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue VIII August 2026

For classification, cross-entropy is $L\_CE = -\Sigma\_k\ y\_k \log(P\_k)$. For RUL regression, $MAE=(1/N)\Sigma\_i\ |r\_i-r\_hat\_i|$ and $RMSE=\mathrm{sqrt}((1/N)\Sigma\_i(r\_i-r\_hat\_i)^2)$. The joint loss can be written as $L = \lambda\_c\ L\_CE + \lambda\_r\ L\_RUL + \lambda\_reg\ ||\theta||^2$.

Experimental Design

Dataset

NASA C-MAPSS is proposed as the primary benchmark. NASA's Prognostics Center of Excellence repository identifies the Turbofan Engine Degradation Simulation Data Set and cites Saxena and Goebel (2008) [7]. The dataset contains simulated degradation trajectories and multiple sensor channels.

Data split

Training, validation, and test sets should be separated at the engine/unit level rather than randomly splitting individual windows, preventing leakage between overlapping windows.

Baselines

Compare Random Forest or Gradient Boosting, LSTM, Transformer without Digital Twin state, and the proposed Transformer with Digital Twin state.

Metrics

Fault classification: accuracy, precision, recall, F1-score and ROC-AUC. RUL: MAE and RMSE. Operational metrics: early-warning lead time, false-alarm rate, missed-failure rate, and inference latency.

Ablation Study

Configuration A uses raw sensor windows with LSTM. Configuration B uses Transformer without Digital Twin state. Configuration C adds Digital Twin state features. Configuration D adds SHAP explanation. Configuration E evaluates reduced sensor sets. Results should be reported as mean ± standard deviation over repeated runs where feasible.

Model	Twin State	Temporal Model	Fault Metrics	RUL Metrics
ML baseline	No	RF/GB	Accuracy, F1	MAE, RMSE
LSTM	No	LSTM	Accuracy, F1	MAE, RMSE
Transformer	No	Transformer	Accuracy, F1	MAE, RMSE
Proposed	Yes	Transformer	Accuracy, F1, AUC	MAE, RMSE

Table 1. Planned experimental comparison.

Expected Results And Result Reporting Protocol

No numerical experimental result is claimed in this draft. After implementation, the results section should report measured performance for every baseline and the proposed model, including mean and standard deviation, training time, and inference latency. The analysis should determine whether Digital Twin state information provides meaningful improvement rather than assuming that model complexity guarantees better performance.

Model	Accuracy	Precision	Recall	F1	RUL RMSE
Random Forest	To be measured	To be measured	To be measured	To be measured	To be measured
LSTM	To be measured	To be measured	To be measured	To be measured	To be measured
Transformer	To be measured	To be measured	To be measured	To be measured	To be measured
Proposed DT-Transformer	To be measured	To be measured	To be measured	To be measured	To be measured

Table 2. Experimental results table to be populated after implementation.

DISCUSSION

The expected value of the framework lies in integration rather than in selecting a single neural architecture. A Digital Twin can provide operating context, asset identity, health indicators, and synchronization metadata that are difficult to represent using raw sensor sequences alone.

The twin introduces additional engineering requirements including state consistency, model validation, interoperability, data governance, and computational overhead. These issues are repeatedly identified in Digital Twin predictive-maintenance literature [5,6].

SHAP explanations can identify influential variables but should not automatically be interpreted as physical causality. Maintenance engineers should validate explanations against known fault mechanisms and sensor quality. Generalization is a major concern. A model trained on simulated degradation may not transfer directly to a physical production line because of differences in sensor characteristics, loads, sampling frequencies, maintenance policies, and failure modes.

Limitations

• The initial evaluation relies on public benchmark data.

• Digital Twin synchronization is computationally represented and does not by itself establish a physical industrial twin.

• Transformer performance depends on window size, hyperparameters, and data volume.

• SHAP is attribution-based and does not prove causality.

• Simulated RUL trajectories may differ from real maintenance records.

• Industrial deployment additionally requires cybersecurity, interoperability, latency, and safety validation.

Future Work

Future work should extend the evaluation to real industrial datasets involving bearings, gearboxes, CNC tools, pumps, or robotic cells. Physics-informed Digital Twins can combine degradation constraints with data-driven learning. Edge deployment should be evaluated for low-latency inference, while federated learning may support privacy-preserving collaboration across factories. Uncertainty estimation should also be incorporated so that maintenance decisions consider prediction confidence rather than only point estimates.

CONCLUSION

This paper proposed a Digital Twin-driven deep-learning framework for real-time fault prediction and predictive maintenance in smart manufacturing. The framework integrates IIoT sensing, Digital Twin synchronization, temporal Transformer modeling, RUL estimation, and SHAP-based explanation. Numerical performance claims are intentionally excluded until implementation and experimental validation are completed. The proposed ablation design provides a means to determine whether Digital Twin state information offers measurable benefit over standalone temporal deep learning. The manuscript therefore provides a structured and reproducible foundation for subsequent experimental research.

REFERENCES

1. Tao, F., Zhang, M., Nee, A.Y.C. (2019). Digital Twin Driven Smart Manufacturing. Academic Press. DOI: 10.1016/C2018-0-02206-9.
2. Rasheed, A., San, O., Kvamsdal, T. (2020). Digital Twin: Values, Challenges and Enablers. IEEE Access, 8, 214000–214019.

3. Vaswani, A., Shazeer, N., Parmar, N., et al. (2017). Attention Is All You Need. Advances in Neural Information Processing Systems 30.
4. Li, X., Ding, Q., Sun, J.-Q. (2024). Remaining Useful Life Prediction Based on Deep Learning: A Survey. Sensors, 24(11), 3454. DOI: 10.3390/s24113454.
5. Ma, S., Flanigan, K.A., Bergés, M. (2024). State-of-the-Art Review: The Use of Digital Twins to Support Artificial Intelligence-Guided Predictive Maintenance.
6. Nagrani, S., Narwane, V.S. (2025). Systematic Literature Review on Digital Twins in Predictive Maintenance. Industrial Engineering Journal, XVIII(02), 19–25.
7. Saxena, A., Goebel, K. (2008). Turbofan Engine Degradation Simulation Data Set. NASA Prognostics Data Repository, NASA Ames Research Center.
8. Lundberg, S.M., Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. Advances in Neural Information Processing Systems 30.
9. Cummins, L., Sommers, A., Bakhtiari Ramezani, S., et al. (2024). Explainable Predictive Maintenance: A Survey of Current Methods, Challenges and Opportunities.
10. Pal Pandian, P., Sasianandham, K., Katherrine, C.J. (2025). Digital Twins for Predictive Maintenance of Production and Machines: A Comprehensive Review. IEEE ICCIST 2025. DOI: 10.1109/ICCIST67338.2025.11438591.
11. Laxkar, P. (2025). The Role of Machine Learning-Driven Digital Twins for Smart Manufacturing Predictive Maintenance: A Review. Journal of Global Research in Mathematical Archives.
12. Hamel, C., Ahsan, M.M., Raman, S. (2024). PMI-DT: Leveraging Digital Twins and Machine Learning for Predictive Modeling and Inspection in Manufacturing.
13. Goodfellow, I., Bengio, Y., Courville, A. (2016). Deep Learning. MIT Press.
14. LeCun, Y., Bengio, Y., Hinton, G. (2015). Deep learning. Nature, 521, 436–444. DOI: 10.1038/nature14539.
15. Schmidhuber, J. (2015). Deep learning in neural networks: An overview. Neural Networks, 61, 85–117. DOI: 10.1016/j.neunet.2014.09.003.
16. Ismail, L., Abdelmoti, A., Basu, A., et al. (2025). A Systematic Review of Digital Twin-Driven Predictive Maintenance in Industrial Engineering.