International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(19s) 2026 pp. 1368-1388
https://doi.org/10.70917/ijcisim-2026-5135
Article

Integrating Artificial Intelligence and IoT for Predictive Maintenance in Smart Manufacturing Systems

Richa Sharma[1], Lokesh Kumar[2]

[1]Uttaranchal Institute of Management (UIM), Uttaranchal University, Dehradun, Uttarakhand, India *
[2]Uttaranchal Institute of Management (UIM), Uttaranchal University, Dehradun, Uttarakhand, India
Corresponding authors: Richa Sharma — Email: 1richa.kaushik.kni@gmail.com
Email: 2lokeshindiaone@gmail.com

Abstract: Downtime is one of the most expensive events in today's manufacturing and downtime on a critical line can run the organization a ten of thousands of dollars per hour in lost production, lost product and expedited repair. The traditional maintenance strategies (reactive maintenance after failure and preventive maintenance on fixed calendar schedule) are not effective enough to satisfy the reliability, cost and sustainability requirements of production environments in Industry 4.0. This paper introduces a single Artificial Intelligence and Internet of Things (AI-IoT) framework for predictive maintenance (PdM) in smart manufacturing systems, which is capable of continuously monitoring the health of equipment, diagnosing incipient failure, predicting Remaining Useful Life (RUL), and generating timely, actionable maintenance decisions before failure. The proposed framework consists of a multi-modal IoT sensing layer, an edge-cloud hybrid computing architecture, a feature-engineering pipeline optimized for rotating and reciprocating machinery, and a hybrid deep-learning core, which features the Convolutional Neural Network (CNN) for spatial feature extraction, the Long Short-Term Memory (LSTM) network for temporal degradation modeling, and the attention-weighted fusion layer for joint fault classification and RUL regression. A rule and model hybrid decision support layer converts model results into prioritized maintenance work orders with the consideration of resources. The framework is tested against two widely used public test datasets: the NASA C-MAPSS turbofan degradation dataset and the Case Western University University (CWRU) bearing fault dataset, as well as on a simulated multi-machine factory-floor testbed that simulates a representative discrete-manufacturing production line. Experimental results demonstrate that the proposed hybrid CNN-LSTM-Attention model can achieve 98.1% fault-classification accuracy and reduce the error of RUL prediction (RMSE) by 17.6% compared with the baseline model (LSTM) and 24.3% compared with the baseline model (Random Forest), while maintaining the inference latency under 45 mS per prediction window at the edge. A cost-benefit analysis performed on the factory floor case study simulated results of an estimated 32–38% decrease in unplanned downtime and 18–25% decrease in overall factory maintenance cost compared to a preventive-maintenance baseline, which will need to be verified through field deployment. The paper also explores the architectural, organizational and data-governance considerations that must be taken into account for scaling such a framework and outlines open challenges, such as cross-machine generalization, label scarcity for rare failure modes and integrating with existing Manufacturing Execution Systems (MES) and Computerized Maintenance Management Systems (CMMS), which will motivate future work.

Keywords: Predictive Maintenance, Internet of Things (IoT), Artificial Intelligence (AI), Smart Manufacturing, Industry 4.0, Remaining Useful Life (RUL), Fault Diagnosis, Deep Learning, Edge Computing, Digital Twin, Condition Based Monitoring (CBM), and Industrial IoT (IIoT).

1. Introduction

1.1 Background and Motivation

Industry 4.0 (or the fourth industrial revolution) refers to the merging of cyber-physical systems, the Industrial Internet of Things (IIoT), big-data analytics, and artificial intelligence in manufacturing contexts. One of the core principles of this convergence is condition based and predictive maintenance, that is: having an accurate picture of the actual condition of equipment in real-time, and making maintenance decisions only as needed and not on a predetermined schedule, or after an actual failure. Reactive maintenance (repairing equipment when it fails) is also most expensive because unplanned downtime makes production schedules impossible, damages surrounding equipment, and usually involves parts and labor being rushed into service. Preventive maintenance involves servicing or replacement of parts every fixed period or every predetermined unit of use, even though they may be fine, and can cause unnecessary maintenance of parts, which wastes labour and spare parts inventory, and may lead to unnecessary replacement of parts when they are not yet ready for replacement.

Predictive maintenance (PdM) overcomes these problems by applying machine learning algorithms to machine sensor data (from vibration, temperature, acoustic emission, electrical current, and lubricant condition, among others) to determine the current condition of the equipment and predict when a part will fail, which is referred to as the remaining useful life (RUL). Industry surveys and academic research have both found that such a program can save from 10 to 40% of maintenance costs over a period of time and that unplanned downtime can be cut in half between 35 and 45% over the same period compared to the traditional reactive and/or preventive approach; again, actual savings rates differ significantly depending on the industry, equipment type, and level of implementation maturity of the program. While this is all desirable, it comes with a set of interrelated technical challenges: sensing at scale across large and heterogeneous equipment fleets reliably and cost-effectively, with low power consumption; robust feature extraction, given the presence of noisy, non-stationary industrial time-series data; machine-learning models that generalise across operating conditions, load levels, and indeed across different machines of the same or similar type; an architecture that can deliver timely predictions subject to the latency, connectivity and reliability constraints of a working factory floor.

1.2 Problem Statement

Although there is significant academic and industrial research interest in predictive maintenance using artificial intelligence, there are still some practical challenges between the research literature and the practical application in the factory floor. First, many published fault-diagnosis and RUL-estimation models have been developed and evaluated using only one benchmark dataset under a few benchmark operating conditions, and there are only few evidences of robustness to the sensor noise, missing data, and varying sensor load characteristics that will inevitably be present in actual production settings. Second, most published architectures focus on either fault classification or RUL regression, but not both, which is unfortunate as a maintenance planner must have both a correct diagnosis of issues and an estimation of remaining useful life before maintenance is required. Third, the computational architecture question — what should be implemented at the sensor node, what should be implemented at the edge gateway, and what to implement in the cloud — is often not considered, despite its direct implication on the ability of a proposed system to satisfy real constraints of factory floor latency, connectivity, etc. Fourth, there are relatively few studies that translate the output of the models to the type of maintenance work orders that are prioritized and based on the limited resources available, and integrate with the existing Computerized Maintenance Management Systems (CMMS) and Manufacturing Execution Systems (MES), which is what matters for the applicability of a predictive-maintenance system in the operations.

The paper fills these gaps with a proposed framework, AIoT-PdM, which:

It (a) classifies the faults and estimates the RU from the same multi-modal sensor representation; (b) explicitly partitions computing between the device–edge–cloud hierarchy to address factory floor constraints; (c) is tested not only on public benchmark datasets but also on a simulated multi-machine factory-floor testbed designed to reveal robustness, scalability and computational efficiency under conditions that better simulate real deployment; and (d)

explicitly includes a decision-support layer that translates raw model outputs into prioritized maintenance recommendations.

1.3 Research Objectives

The specific goals of this research are: (1) design a multipodal IoT sensing and edge-cloud computing architecture suitable for predictive maintenance in discrete and process manufacturing environments; (2) develop a hybrid deep-learning model that is able to perform the task of fault classification and RUL estimation jointly with a higher accuracy than a single model baseline; (3) evaluate the designed framework on established multiday public benchmarks as well as a representative simulation of a multi-machine factory floor, assessing its accuracy, latency, scalability and robustness to sensor noise and missing data; (4) quantify via a case-study cost-benefit analysis the potential operational and financial impact of the designed framework compared to the preventive maintenance baseline; and (5) identify practical deployment considerations and open research challenges that lie before a framework such as this one can be used in industrial settings at scale.

1.4 Contributions

This paper has the following contributions:

A single AI-IoT predictive-maintenance architecture (AIoT-PdM) covering the sensing, edge and cloud layers, featuring an explicit computational-partitioning method that meets the latency and connectivity requirements of the factory floor.

2. A multi-class fault classification and continuous RUL regression model based on a learned representation, which is jointly performed by a hybrid CNN-LSTM-Attention deep learning model instead of separately tackling the two tasks.

3. Rule and model hybrid layer of decisions support for prioritizing maintenance tasks based on the classification and RUL results for integration into existing CMMS/MES systems; and integrating the prioritized maintenance tasks with the existing systems.

4. An extensive empirical study comprising of two public benchmark datasets (NASA C-MAPSS and CWRU bearing data) and a simulated multi-machine factory-floor testbed, with testing performed on classification and RUL accuracy, ablation analysis, computational latency, scalability in the number of machines, and robustness to sensor noise and missing data.

5. A case study and cost-benefit analysis of the case study setup, which places the framework in the context of a representative discrete-manufacturing production line, as well as a discussion about deployment considerations, organizational considerations and data-governance considerations for industrial deployment.

1.5 Paper Organization

The rest of this paper is organized as follows: Section 2 summarizes the existing research on predictive maintenance, condition monitoring using IoT, and fault diagnosis and RUL estimation using AI/ML methods and pinpoints the research gap. Proposed AIoT-PdM framework and its architectural layers are presented in Section 3. The experimental materials and methods are described in Section 4, including the datasets, preprocessing, model configuration and evaluation metrics. Results and discussion, including comparative performance, ablation analysis, latency, scalability and robustness are presented in Section 5. Section 6 contains a case study and cost benefit analysis of deployment on a representative production line of smart manufacturing. Some deployment issues are covered in Section 7 and limitations are covered in Section 8. The paper ends in section 9 which gives directions for future work.

2. Related Work

2.2 Predictive Maintenance Approaches: Technical Details

There are generally three types of maintenance strategies for a manufacturing process. Reactive (or corrective) maintenance – Equipment is only repaired or replaced once it has failed, helps to reduce maintenance labour in the short term but increases unplanned downtime, and secondary damage risks. Preventative Maintenance is a maintenance activity that is performed on a regular basis at a fixed time or usage interval based on statistical failure rate information (Mean Time to Failure), which tends to make fewer failures than predictive maintenance, but may involve changing parts much sooner than they are likely to actually fail. Predictive maintenance or even condition-based maintenance (CBM) as it is sometimes called, if it does not involve a predictive prognosis, is an ongoing process where equipment condition is monitored for the condition and only takes action when the data suggests that the failure risk has become critical, with sufficient lead time to plan for maintenance without impacting on production. The literature also distinguishes between the diagnosis (categorising the current fault type/s and fault severity) and the prognostics (forecasting the future degradation trajectory, and estimating Remaining Useful Life). These are both discussed together in the framework outlined in this paper.

2.2 IoT & Industrial IoT (IIoT) for Condition Monitoring (CM)

The spread of low-cost accelerometers, acoustic-emission sensors, infrared thermal sensors and current-clamp sensors and low-power wireless communication standards (such as IEEE 802.15.4 based mesh networks, Bluetooth Low Energy and industrial versions of Wi-Fi) has made the dense, continuous instrumentation of manufacturing equipment economically viable at a scale once confined to only the most critical assets. Industrial IoT (IIoT) platforms take the consumer IoT ideas and add the reliability, determinism and interoperability characteristics of a manufacturing environment and are typically based on industrial communication protocols like OPC-UA and MQTT for transferring data from the shop floor to higher level information systems. Most of the practical IIoT architectures for condition monitoring, such as the one suggested in this paper, are a hybrid edge-cloud design instead of being purely edge or cloud because both sides have advantages and drawbacks.

2.3 The student will learn how to apply machine learning and deep learning to fault diagnosis.

The machine-learning methods used in classical fault diagnosis, such as Support Vector Machines (SVM), Random Forests and Gradient Boosted Trees, are efficient when working with hand-engineered, statistical and frequency-domain features (e.g., root-mean-square vibration amplitude, kurtosis, spectral energy in characteristic fault-frequency bands) derived from raw sensor signals, and are performing well on problems where the fault signature is well-known, physically interpretable, and can be extracted from the raw sensor signals. The deep-learning methods, on the other hand, extract the feature representation directly from raw or slightly pre-processed sensor data, although they require larger amounts of labeled data and more computational power to learn. CNNs, which were designed for image data, have been successful in extracting the local spatial and spectral patterns from vibration and acoustic waveforms as well as time-frequency representations like spectrograms. Recurrent architectures, especially Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks, are well suited for modeling the temporal degradation trajectories. For most components, the degradation process typically does not happen in one fell swoop over a brief operating history, but rather is a gradual process that spans across a much longer operating history. The hybrid model proposed in Section 3.4 is based on hybrid CNN-LSTM architectures, which combine convolutional layers to capture local spatial/spectral features with recurrent layers to capture temporal evolution of features. Hybrid CNN-LSTM architectures are successful on fault-classification and RUL-estimation benchmarks and serve as the architectural building blocks of the hybrid model proposed in Section 3.4. Recently, attention mechanisms and transformer-based architectures were implemented as a solution for RUL estimation, in which the attention mechanism is able to focus more on the degradation-relevant parts of the long input sequence than on the less informative parts, which inspires the attention weighted fusion layer in the proposed framework.

2.4 Remaining Useful Life (RUL) Estimation

RUL estimation is typically formulated as a regression problem: If a window of sensor data is given up to the current operating cycle, how many cycles (or operating cycles remaining) does the component last until it fails or

exceeds acceptable operating tolerances? The NASA Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) dataset consists of simulated turbofan engine degradation trajectories under various operating conditions and fault modes, and is a commonly adopted benchmark for RUL-estimation research, for which the various models in the literature can then be compared with each other and evaluated using consistent metrics like Root Mean Squared Error (RMSE) and the asymmetric scoring function originally proposed with the dataset that gives larger penalties for incorrect late predictions (which could potentially miss maintenance windows) than for early predictions (which simply result in a conservative estimate). One of the constant results from the research into RUL is that it is extremely difficult to accurately predict RUL early in the life of a component, when changes associated with degradation are far more likely to be confused with other normal operating phenomena or sensor noise, thus the accuracy of most existing published models has been primarily evaluated in the later stages of the operating life, where the value of a good estimate of RUL is highest for the purposes of maintenance scheduling.

2.5 Bearing and Rotating-Machinery Fault Diagnosis

One of the most frequently occurring types of failures in rotating machinery devices is due to rolling-element bearings, and the inner-race, outer-race, ball, and cage fault frequencies are well documented in the mechanical-engineering literature. The Case Western Reserve University (CWRU) bearing dataset is a common dataset widely used in bearing fault-classification research and is employed in this paper to test the performance of the proposed model for fault-classification. It contains vibration data from bearings with seeded faults of various type, diameter, and load conditions. The hybrid design used in this paper represents one of the highest classification accuracy reported in the literature for this benchmark, which can be achieved by published approaches that can be classified into two categories: envelope-spectrum analysis approaches based on classic concepts of bearing fault-frequency theory, and deep-learning classifiers that operate on raw vibration waveforms, time-frequency spectrograms, or engineered statistical features.The highest classification accuracy reported on this benchmark is generally achieved by envelope-spectrum analysis approaches, based on the classic concepts of bearing fault-frequency theory, and deep-learning classifiers that operate on raw vibration waveforms, time-frequency spectrograms, or engineered statistical features, which are the two categories of approaches mentioned above and which are used in this paper.

2.6 Digital Twins and Physics-Informed Approaches

The other line of research is the combination of physics-based models of equipment degradation (such as fatigue and wear models based on mechanical-engineering first principles) with data-driven machine-learning models, frequently referred to as the “digital twin” concept — a continuously updated virtual image of a physical asset that incorporates simulation and real-time sensor data. Physics-informed and hybrid physics-ML modeling techniques have been shown to enhance the generalization capabilities to operating conditions and fault severity not accounted for during training, and to offer interpretable failure-mechanism explanations not available by purely data-driven models; however, accurate physical models of the specific equipment being monitored are not always available and are often not economically feasible for low-criticality assets. The framework outlined in this paper is purely data-driven, and in Section 9 we describe a potential avenue of future investigations that combines physics informed models and digital-twin approaches where appropriate—especially in high-value, well-defined equipment.

2.7 Real-time constraints and edge computing in manufacturing are explored.Edge Computing and Real-Time constraints in manufacturing is explored.

Actionable alerts from predictive-maintenance systems may need to come with a real-time or near real-time response, especially when the fault might develop into a functional failure in a short time (such as some bearing faults once they reach a certain stage). Edge-learning architectures for industrial monitoring execute local or near-term inference without relying on network connections or central-server latency: in several studies, edge-deployed models of low complexity (compared with models deployed to central servers) have been shown to produce a sub-100ms inference latency on embedded or single-board-computer-class devices for vibration-based fault classification. This

encourages the edge-cloud partitioning strategy that we use in the proposed AIoT-PdM architecture, as discussed in Section 3.

2.8 The other frameworks are compared here.The other frameworks are compared here 2.8 Comparison of Representative Related Frameworks

Table 1 presents a summary of the representative predictive-maintenance frameworks from the literature and the proposed AIoT-PdM framework, based upon the sensing modality, the model architecture, the parallel processing of diagnosis and prognosis, the processing location of the model, and the inclusion of decision-support.

Table 1. Comparing representative predictive-maintenance plan with the proposed AIoT-PdM framework.

Identifies the major sensing modalities used in the primary sensing system	**Specifies the primary sensing system's model architecture**	**Defines the joint diagnosis + RUL model architecture**	**Defines the computational placement model architecture**	**Defines the model architecture for decision support / CMMS integration**				
Yes	Yes	Yes	Yes – diagnosis only	No (local instrument / PLC)	Manual review			
Traditional methods for bearing fault classification (CNN-based)	Vibration	CNN	No (only diagnosis)	Not addressed. The use of traditional approaches to bearing fault classification (CNN-based)	Vibration	CNN	No (only diagnosis)	Not addressed.
LSTM-based RUL estimators (C-MAPSS)	Multivariate sensor stream	LSTM / GRU	Not addressed (RUL only)	Cloud	Addressed			
Hybrid CNN-LSTM RUL models	Multivariate sensor stream	CNN + LSTM	No (RUL only)	Cloud	Not addressed			
Digital-twin physics-ML hybrids	Multi-modal + simulation	Physics model + ML	Partial (asset-specific)	Cloud / server	Partial (asset dashboards)			

Application of lightweight fault classifiers on the edge	Vibration / acoustic	Lightweight CNN	No (diagnosis only)	Edge (embedded)	Not addressed			
Proposed AIoT-PdM	Vibration, acoustic, temperature, current, lubricant	CNN + LSTM + Attention fusion	Yes (joint)	Edge + fog + cloud	Yes (prioritized work orders, CMMS/MES-oriented)			

2.9 Research Gap

Based on the literature reviewed above, four gaps are motivated for the present work. First, fault classification and RUL estimation are normally modeled as separate tasks, despite the fact that the maintenance planner needs both fault diagnosis and fault prognosis information, and a common representation learnt jointly in the two modeling problems may achieve better overall performance than RUL and fault classification learnt separately. Third, there have been limited studies that directly tackle the computational-placement problem along the device–edge–cloud continuum, which directly impacts deployability under realistic factory-floor latency and connectivity requirements. Thirdly, most published evaluations are based on relatively clean benchmark data, and have not been tested for robustness to the sensor noise, missing data and changing operating conditions found in production. The final step to transform the model outputs into prioritized maintenance recommendations that incorporate resource information and are usable within existing maintenance management systems (CMMS/MES) is largely missing from the technical literature, as most studies only report the classification accuracy or RUL error from the model. This paper presents a framework of AIoT-PdM to solve all four gaps in one architecture under evaluation.

3. The proposed framework is the AIoT-PdM architecture.The proposed framework is AIoT-PdM Architecture.

The proposed AI-IoT Predictive Maintenance framework (AIoT-PdM) consists of five architectural layers: (1) IoT Sensing Layer, (2) Data Preprocessing and Feature Engineering Layer, (3) AI/ML Prediction Core, which includes a hybrid CNN-LSTM-Attention model to jointly classify faults and regress RUL values, (4) Edge-Cloud Deployment Layer, which partitions the computation between the device, edge, fog and cloud layer, and (5) Decision-Support and Work-Order Generation Layer, which converts the model output into prioritized maintenance actions.

3.1 System Architecture Overview

Data from the sensors is continuously collected at each monitored asset and forwarded to a local edge gateway that provides service to a group of machines in close proximity (usually a full production line and/or work cell). The edge gateway preprocesses, extracts features and does lightweight inference by a compressed version of the prediction model, thus allowing for low-latency fault detection and coarse RUL estimation without relying on the network to connect to the cloud. A fog-tier server, usually within the plant, aggregates data and predictions from multiple edge gateways; hosts full capacity prediction model for periodic re-scoring of higher accuracy; and conducts cross-machine analytics. The cloud tier is home to model training and retraining pipelines, fleet-wide analytics (across multiple plants), and the long-term data store for model improvement across time. Depending on the organization's preference, the decision-support layer can be deployed in the fog or cloud, and receives the model output from all the connected machines, generating prioritized maintenance work orders to be integrated with the plant's CMMS/MES systems.

3.2 IoT Sensing Layer

The sensing layer is composed of five complementary sensor modalities, each one selected to detect various types of faults which are most likely to occur in rotating and reciprocating industrial machinery: (i) tri-axial vibration accelerometers sampling at 10–25 kHz are used to detect bearing, gear and imbalance/misalignment faults through their signature vibration frequencies; (ii) acoustic-emission sensors sampling at higher frequencies (100 kHz to 1 MHz) detect early surface fatigue and crack propagation which may not be evident in the lower-frequency vibration spectrum; (iii) infrared or contact temperature sensors are used to measure the temperature of the machine during operation, and detect overheating associated with bearing lubrication failure, electrical faults or excessive friction; (iv) current-clamp sensors connected to motor supply lines detect electrical faults and load-related anomalies via the motor current signature analysis; and (v) periodic or continuous lubricant-condition sensors such as particle-count or dielectric-constant sensors detect lubricant degradation or contamination, a leading indicator of accelerated wear. The sensing modalities and their typical sampling properties and failure modes are summarized in Table 2.

Table 2. The modalities of the IoT sensors in the context of the AIoT-PdM framework.

Touch	**100 Hz - 1 KHz**	**Short circuit, shorts to adjacent wires, or voltage overloads**	**At the end of the cable or near the end of the subject's arm or leg**

A vibration accelerometer that measures vibration in three directions.Vibration accelerometer that measures vibration in three axes, used for detecting bearing wear, gear tooth defects, imbalance, misalignment, etc., in a bearing housing or gearbox casing.

Acoustic emission sensor	**100 kHz-1 MHz**	**Early stage surface fatigue, crack initiation/propagation**	**Near high stress structural members**
Infrared / contact temperature sensor	1 Hz–1 kHz	Lubrication failure, electrical faults, excessive friction	Bearing housing, motor casing, electrical cabinet
Current-clamp (motor current signature)	1 - 10 kHz	Electrical faults, rotor bar defects, load anomalies	Motor supply line

A lubricant-condition sensor measures the level of particles and/or dielectric in the lubricant and can be continuous or periodic (minutes) and indicates the level of degradation or contamination and the onset of accelerated wear in the lubrication system.

3.3 Data Preprocessing and Feature Engineering: 3.3

The raw sensor streams go through a common pre-processing pipeline: segmentation into short windows (typically 1-5 seconds for vibration and acoustic data, and minutes to hours for temperature, current and lubricant data), normalization of each channel by using running statistical data to compensate for inevitable sensor drift and general differences in the calibration of the sensors, and rejection of the outliers which represent transient disturbances that are not related to the condition of the equipment (e.g. temporary disconnection of a sensor). As well as the time-domain raw windowed waveform, time-domain statistical features such as root-mean-square amplitude, kurtosis, skewness, and crest factor are computed, and frequency-domain features are computed using Fast Fourier Transform and embedded in the bands of known bearing and gear-mesh characteristic frequencies, and this raw windowed waveform and these features are passed to the CNN portion of the prediction model (Section 3.4) for automatic feature learning. This is a combination of engineered and learned features, and this is a result of what many fault diagnosis papers have reported: that hybrid feature representations work better than hand-engineered or learned representations by themselves, especially if there is limited number of training examples for rare faults.

3.4 AI/ML Prediction Core

The prediction core is a hybrid deep learning system consisting three components: a spatial/spectral feature extractor (CNN); a temporal degradation encoder (LSTM); and an attention-weighted fusion block connected to two heads: a fault-classification head and a RUL-regression head, both of which are trained simultaneously.

3.4.1 CNN Spatial/Spectral Feature Extraction Block

The one-dimensional convolutional layers (16–64 filters, 3–9 samples deep) are stacked on each windowed, multi-channel sensor segment with max-pooling and batch normalization layers in between, which learn local spatial and spectral patterns directly from the raw and lightly processed waveform, complementing the hand-designed frequency-domain features described in Section 3.3. The output of this block is a per-window embedding vector containing a summary of the local signal characteristics that are important for fault signatures, but which is small in size.

This is an LSTM Temporal Degradation Encoder.This is an implementation of LSTM temporal degradation encoder.

A vector of embeddings is passed per window for a longer history of operation of the machine (usually the last 30-100 windows, or minutes to hours of operation depending on the sampling setup), and these embeddings are passed through two layers of bidirectional LSTM (64-128 hidden units each), which model the machine's condition over time, capturing the gradual degradation trajectory characteristic of most wear-based failure mechanisms. This block directly controls the model's ability to differentiate a truly evolving fault from a transient non-progressing anomaly (such as a short load spike) which is not an indication of a developing fault in the equipment.

Aim to create a beautiful image with the output heads.Try to achieve a beautiful image with the output heads, 3.4.3 Attention-Weighted Fusion.

The attention mechanism calculates a learned weighting on the output sequence of the LSTM and the model places more attention on the most informative time steps for the current diagnosis and prognosis, namely the most recent and most abnormal time steps in the input window. The attention-weighted representation is then sent to two output heads trained jointly with a combined loss function: a softmax classification head, which is used to predict the current fault category including a “normal/healthy” class, which is relevant to the monitored asset class; and a regression head, which is used to predict the RUL (in operating cycles or hours). The joint loss function is a weighted sum of two loss functions: one from the classification head (categorical cross-entropy), and one from the shared representation (mean squared error) for the RUL regression head; the weights of the two loss components are optimized by using the validation-set performance, enabling the joint training of the two heads; the degradation patterns learned by the classification head are used to inform, and be informed by, the shared representation used for RUL regression.

3.5 The models are combined with a weight of 3.5.The models are fused together at a weight of 3.5.

The framework provides an option for a fusion mode, in which the data-driven RUL estimate is fused with a physics-based RUL estimate using a weighting method based on the accuracy of each estimate, determined by its recent validation on the specific asset class, for the assets where a physics-based degradation model is available and deemed reliable (see Section 2.6). The framework is mostly data-driven, as tested in Sections 4–5 of this paper, for most assets for which no physics-based model is available.

3.6 Edge-Cloud Deployment Architecture

Table 3 shows the summary of the computational partitioning strategy in the device, edge, fog and cloud tiers. The most latency-sensitive inference is coarse fault detection that requires immediate detection of an alarm, which is made to operate on a compressed (quantized and pruned) version of the CNN component at the edge gateway, and generally operates with sub-50 millisecond inference latency on a single-board-computer-class processor. The full

capacity joint classification and RUL inference, including the entire CNN-LSTM-Attention model is performed periodically (e.g., every 1-15 minutes depending on the criticality of the asset), as RUL estimates also don't need to be performed in milliseconds, but they would benefit from the model's higher accuracy with the uncompressed model. These model trainings and retrainings use significantly more compute resources and access to the full historical data set, and are performed at the cloud tier.

Table 3. The computational partitioning over the hierarchy of the device–edge–fog–cloud in the AIoT context.

Tier 1	Transmission Control Protocol (TCP) Server Socket	TA10201052A	Model (wired)	50nS

This means that a signal at the sensor node with a microcontroller needs to be acquired and processed with simple filtering, and the result is sent without further processing, with a response time of < 5 ms.

Edge Gateway	Single Board Computer (e.g., Raspberry Pi / Jetson-class)	Feature Extraction, Coarse Fault Detection	Compressed CNN, quantized and pruned	< 50ms
Fog server	On-premises server	Full joint classification + RUL inference and cross-machine aggregation	Full CNN-LSTM-Attention model	Seconds (periodic)
Cloud	Data-center compute (GPU-accelerated)	Model training/retraining, fleet-wide analytics, long-term archive	Full model + training pipeline	Not latency-critical (batch)

3.7 Algorithmic Summary

The end-to-end pipeline of the AIoT-PdM as discussed in each monitored asset includes (1) acquire and locally filter the raw multi-modal sensor data at the device tier; (2) segment the data into sliding windows and extract engineered time- and frequency-domain features at the edge gateway; (3) run the compressed CNN component on each window to get a coarse fault-detection signal and immediately alert if a high-confidence fault is detected; (4) periodically forward the windowed feature sequence to the fog tier and run the full CNN-LSTM-Attention model on the window to get a joint fault classification and RUL estimate; (5) update the asset's priority score in the decision-support layer based on the latest classification, RUL, and asset-criticality information; (6) generate the maintenance work order if the priority score exceeds the configured threshold, sending it to the maintenance planner for review; (7) periodically push the windowed feature sequence to the cloud tier, where the full model is re-trained using the newly accumulated labeled data, and push the updated model weights to the fog and edge tiers.

4. Materials and Methods

4.1 Datasets

The proposed system is tested on two well-known public benchmark datasets, as well as a simulated testbed comprising a multi-machine factory floor for testing scalability and robustness beyond the scope of either public benchmark.

The NASA Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) dataset consists of simulated trajectories of degradation for turbofan engines, consisting of multiple operating conditions and fault modes, with each trajectory ("engine unit") instrumented with 21 sensor channels that log all their data throughout their entire operational life until the point of failure. This paper evaluates the performance of the RUL-estimator under simpler

and more challenging situations using the sub-datasets FD001 and FD004, respectively, corresponding to a single operating condition and multiple operating conditions.

The Case Western Reserve University (CWRU) bearing database consists of vibration measurements generated from seeded faults (inner race fault, outer race fault, and ball fault) of different bearing diameters under four motor-load conditions, along with baseline data of vibration measurements from a healthy bearing. This dataset is applied to test the CNN-LSTM-Attention core that is proposed in this work to classify faults.

In addition, a simulated multi-machine factory-floor testbed was also developed, consisting of 40 virtual assets representing a diverse combination of motor-driven pumps, conveyor drive units and CNC spindle motors, where degradation trajectories were simulated using a combination of bearing/gear fault-frequency models and stochastic degradation-rate and noise parameters calibrated to be roughly compatible with the wide-ranging statistical characteristics reported in the public PdM literature. This synthetic testbed is designed to test scalability with machine numbers, robustness to sensor noise and missing data, and end-to-end latency that can be used to assess the relative behaviour of a system, as with any synthetic evaluation, the behaviour found on this testbed should not be interpreted as a validated real-world performance.

Table 4. The experiment used the following datasets:

Dataset	Type	Assets / Units	Primary Task	Notes
NASA C-MAPSS (FD001)	Public benchmark (simulated turbofan)	100 training, 100 test units	RUL estimation	Single operating condition, single fault mode
NASA C-MAPSS (FD004)	Public benchmark (simulated turbofan)	249 training units, 248 test units	RUL estimation	Six operating conditions, two fault modes

Bearing dataset1: CWRU data set2: Public, physical test rig data set3: Healthy + 3 fault types × multiple severities + 4 load conditions Data set4: Fault classification: 12 kHz + 48 kHz sampling variants

Testbed on simulated factory floor	This study: synthetic (testbed)	40 virtual assets (pumps, conveyors, CNC spindles)	Joint classification + RUL, scalability, robustness	Calibrated to reflect published PdM statistical characteristics

4.2 Data Preprocessing

As is common in the RUL-estimation literature, the C-MAPSS dataset was preprocessed by discarding sensor channels which are close to being constant for all operating conditions, and then normalizing the remaining channels per operating condition by using the z-score normalization. The RUL was limited to an upper bound (usually 125-130 cycles) typical of the practice because degradation in the early life is not well captured by sensor readings and maintenance planning is focused on the end of life. In the case of CWRU, signals were divided into fixed-length windows (2,048 samples) with a 50% overlap, and both time and frequency domain features were extracted, as well as engineered features (Section 3.3), for each window. The simulated factory-floor testbed is preprocessed as described in Section 3.3, and the same preprocessing steps were applied to all 40 simulated factory floor assets.

The task involves the creation of a model and setting up its parameters.The task is to build a model and to configure its parameters.

The architecture of the proposed CNN-LSTM-Attention model for all three evaluation sets is summarized in Table 5, and only a few minor architectural changes were made for the different tasks, such as the classification head (softmax) for CWRU, the regression head for C-MAPSS, and both heads for the factory-floor testbed.

Table 5. Hyperparameter setting of the proposed model.

Component	Parameter	Value

CNN Block is a neural network block that consists of four convolutional layers.

CNN Block	Filters per layer	16 / 32 / 64 / 64
CNN Block	Kernel size	7, 5, 3, 3

LSTM Block refers to the number of layers (bidirectional) in the LSTM.

LSTM Block	128	number of hidden units per LSTM block

The Input sequence length in LSTM Block is 50 windows.

Layer is a field of input type that specifies what kind of attention to use, either additive (Bahdanau-style) or not.

Training Algorithm	Optimizer	Learning Rate	Adam	1e-3 (cosine decay)
Training	Batch size	64		
Training epochs	100 (early stopping, patience = 10)			

Loss Weight: Classification (tuned using validation): 0.5: 0.5 (regression loss weight: 0.5)

Dropout rate	Regularization	0.3 (CNN and LSTM blocks)

4.4 Baseline Models

The proposed CNN-LSTM-Attention model is tested and compared with following baseline models which are also trained and tuned using the same preprocessed data splits: (i) Random Forest (only using engineered statistical/frequency features); (ii) Support Vector Machine (SVM) (using engineered features only with the radial basis function kernel); (iii) standalone CNN (no LSTM/attention); (iv) standalone LSTM (no CNN/attention); (v) CNN-LSTM (no attention) (architecture without the attention weighted-fusion-layer): and (vi) the proposed full CNN-LSTM-Attention model. All deep-learning baselines have the same regression head design for RUL estimation, and the same softmax head design for fault classification, allowing the effect of the underlying feature-extraction architecture to be separated out.

4.5 Evaluation Metrics

The evaluation of fault-classification performance is done by using accuracy, precision, recall and F1-score (macro-averaged across fault classes to overcome the class imbalance between healthy and faulty operating state). Root Mean Squared Error (RMSE) in operating cycles (OCs), Mean Absolute Error (MAE) and the asymmetric scoring function that was introduced with the C-MAPSS dataset which gives higher penalty for late RUL predictions (causing maintenance to be missed) than to early ones are used to evaluate the performance of RUL-estimation. We measure computational performance by mean of inference latency per prediction window at the edge and fog tiers, and model size (number of parameters, on-disk size) related to deployability at the edge. Scalability is measured in terms of the mean end-to-end processing time when the number of assets being monitored scales up. The robustness is considered by re-computing the classification accuracy and RUL RMSE under simulated sensor dropout and additive noise of increasing severity.

5. Results and Discussion

This section presents comparative results for the three evaluation settings outlined in Section 4: Fault classification on the CWRU bearing dataset, RUL estimation on the NASA C-MAPSS dataset, and joint classification/RUL performance, scalability and robustness on the simulated multi-machine factory-floor testbed. The

factory floor test numbers are always to be taken as a reference of the relative behaviour of the system, and should not be considered as real world test numbers for any given synthetic testbed component.

5.1 Classify faults in a bearing dataset from 5.1 Fault Classification Performance.

Table 6. The classification accuracy for the CWRU bearing dataset (macro-averaged).

Model 1	Recall 0.905	Accuracy 0.945	Precision 0.750	F1 score 0.830
Random Forest (engineered features)	0.912	0.897	0.884	0.890
SVM-RBF (engineered features)	0.898	0.881	0.869	0.875
Standalone CNN (raw waveform)	0.947	0.936	0.929	0.932
Standalone LSTM (engineered features)	0.921	0.909	0.902	0.905
CNN-LSTM (no attention)	0.966	0.958	0.953	0.955
Proposed CNN-LSTM-Attention	0.981	0.977	0.973	0.975

The proposed hybrid model yields the best classification accuracy (98.1%) in comparison with all the tested configurations, which is 3.4% higher than the classification accuracy of the standalone CNN and 1.5% higher than the classification accuracy of the CNN-LSTM without attention. This aligns with the interpretation that the attention layer enables the model to pay attention to the most fault discriminating portions of each input window, which can be especially useful in distinguishing between early-stages faults of different types, that may yield broadly similar statistics.

5.2 Estimate the performance of a RUL. (NASA C-MAPSS)

Table 7. Performance of RUL-estimators in the NASA C-MAPSS FD001 sub-dataset and FD004 sub-dataset.

Model	FD001 RMSE	FD001 Score	FD004 RMSE	FD004 Score
Random Forest (engineered features)	19.8	612	27.4	2,845
SVM-RBF (engineered features)	20.6	648	28.9	3,012
Standalone CNN	16.2	421	23.1	1,987
Standalone LSTM	14.9	372	21.6	1,764
CNN-LSTM (no attention)	13.1	301	19.4	1,462
Proposed CNN-LSTM-Attention	12.3	267	17.8	1,289

The lower the RMSE and score values, the better the performance. The proposed model achieves a reduction of RMSE over the standalone LSTM baseline model by 17.4% on FD001 and 17.6% on FD004 and by 37.9% and 35.0%, respectively, over the Random Forest baseline model. The performance gap between models is larger for FD004 (six operating conditions, two fault modes) compared to FD001 (single operating condition, single fault modes) suggesting that the CNN and attention components are most useful when the operating conditions are more heterogeneous, making it more challenging to separate between genuine degradation and operating-condition induced variation.

5.3 Joint Classification & RUL Performance (Simulated Factory-Floor Testbed)

Table 8. Joint fault-classification and RUL-estimation results on the factory floor testbed with simulated 40 assets.

Model 1: Classification F1-score: 0.899	**RUL RMSE (hours): 25.913**	**RUL MAE (hours): 23.735**	
Random Forest + separate RUL regressor	0.874	38.6	29.7
Standalone LSTM (joint heads)	0.902	31.4	24.1
CNN-LSTM (no attention, joint heads)	0.931	26.8	20.3
Proposed CNN-LSTM-Attention (joint heads)	0.958	22.1	16.9

The joint classification and RUL results for the proposed model are also the best of the assessed models on the more heterogeneous factory-floor testbed (which includes three asset classes: pump, conveyor and CNC spindle, with varying distributions of failure modes, and lower training data volume per asset class compared to the C-MAPSS and CWRU testbeds).

5.4 Ablation Study

Table 9. The simulated factory-floor testbed is used to evaluate the ability to classify each joint based on the RUL scores, as measured by the Ablation study F1-score and RUL RMSE.

Best Configuration	**Classification F1-score**	**Best RUL RMSE (hours)**
Full proposed model	0.958	22.1
− Attention layer	0.931	26.8
− LSTM block (CNN + attention only)	0.887	34.9
− CNN block (LSTM + attention on engineered features)	0.914	29.2
− Joint Training (separate classification/RUL models)	0.939	25.6
− Engineered features (raw waveform)	0.374	37.4

All the components of the buildings make measurable contributions to the overall performance. The removal of LSTM block results in the highest amount of degradation in RMSE of RUL (+12.8 hours), which validates the importance of temporal modelling of the degradation trajectory as the single most important architectural component for prognosis. Joint training of both the classification and the RUL models is an effective way to design the model, as removing training of both heads, instead of training them together, causes a small drop in both metrics. Even when the CNN is able to learn the features from the raw wave-forms, removing engineered features and replacing them with the raw wave-forms (without the features) yields some degradation, but it is measurable, suggesting that hand-engineered frequency-domain features have some value.

The efficiency of computing processes is also crucial, as is the latency of the processes concerned.

Table 10. Latency and size of the model for inference on different deployment tiers.

Standard (default)	0.36ms	1.41 million	2.77million
Edge gateway (compressed CNN, coarse detection)	< 45 ms	≈ 180,000	≈ 0.8 MB (quantized, int8)
Fog server (full CNN-LSTM-Attention model)	≈ 210ms per asset	≈ 2 million	≈ 9.6MB (float32)
Not latency critical (batch / hours)	Same as fog model		

The compressed edge-tier model offers sub-45 millisecond inference latency, which is suitable for near-real-time coarse fault detection, and the full fog-tier model, which is run periodically (not continuously) and provides a higher accuracy joint classification and RUL estimation at a latency (around 210 milliseconds per asset) within the minutes-scale periodic update interval required for RUL tracking. Compared to the uncompressed model, the on-disk CNN model size was reduced by about 4x with an observed reduction in classification accuracy of less than 1 percentage point, similar to results seen in the quantization of other CNN models in the edge-AI literature.

5.6 Scalability Analysis

Table 11. Processing time of fog tier with increasing number of assets being monitored at the same time.

10,000 MAs	900 Milliseconds	40% Memory Usage
10	2.1 s	1.2 GB
40	8.4 s	2.9 GB
100	20.9 s	6.1 GB
500	104.3 s	24.8 GB

The processing time of fog-tier operations scales about linearly with the number of assets being monitored, with most of the processing of each asset being largely independent in the context of batched execution; for a full fleet with 500 assets a full fleet scoring cycle takes less than two minutes, which fits the RUL-update interval of minutes for this framework. In practice, the update frequency that this study demonstrated would probably require a horizontal scalability extension using multiple fog servers with each one managing a subset of the assets of the plant, which has not been explored to date.

5.7 The solution is robust to sensor noise and missing data with 5.7 wins.

Table 12. F1 score and RUL RMSE (simulated sensor degradation conditions: factory floor testbed, full proposed model).

Cleaning up the filter classification	0.84	RMSE (hours) 11.3
Clean data (baseline)	0.958	22.1
10% random sensor-channel dropout	0.937	25.6
30% random sensor-channel dropout	0.881	33.4
Additive Gaussian noise (SNR = 20 dB)	0.929	26.9
Additive Gaussian noise (SNR = 10 dB)	0.803	41.7

The framework's multi-modal sensing approach is apparent in this graceful degradation of performance when sensors are dropped or noise is added to any one channel, as the performance of the other channels can make up for the error due to the lack of a particular sensor. However, at higher sound levels (SNR = 10 dB), both classification and RUL performance suffer significantly, indicating that in order to ensure reliable operation, even sensor calibration and maintenance and redundancy are still required, an element which cannot be solved by the modeling layer alone.

5.8 Discussion

All of the above findings lead to four conclusions. The hybrid CNN-LSTM-Attention architecture is able to outperform the classical machine-learning baselines as well as the simpler deep-learning architecture in all three evaluation scenarios, and more significantly on the more challenging FD004 and factory-floor-testbed settings. Second, the benefit of jointly training a classification and RUL heads, from a shared representation, is measurable and is confirmed to be small compared to the performance of separately trained classification and RUL models, a rationale that is congruent with the architecture presented in Section 3.4. Third, the proposed edge-cloud partitioning strategy can deliver near real-time fault detection (edge tier) and periodic RUL re-scoring (fog tier) with nearly linear scalability as the number of monitored assets increases up to at least several hundred assets per fog server. Fourth, is graceful degradation of performance under moderate degradation in sensors, rather than catastrophic, and severe sensor degradation has significant impact on performance, but still not enough to make predictive-maintenance systems a substitute for good sensor-maintenance and instrumentation practice.

6. The 6th Case Study focuses on the deployment of a representative Smart Manufacturing production line.The 6th Case Study will concentrate on the deployment of a representative Smart Manufacturing production line.

6.1 Use Case Description

As an example of applying the AIoT-PdM framework, a case study is introduced in this section that is modeled after the testbed simulated for a representative discrete-manufacturing production line in the factory that was described in Section 4.1. The line is assumed to be in service 3 hours a day and the cost of unplanned downtime is assumed to be $8,500/hr, which is representative of the cost of unplanned downtime in mid-sized discrete-manufacturing facilities, but can vary significantly based on industry and facility.

6.2 Implementation Details

All 5 sensor modalities are instrumented on the asset, each with a number of assets in the immediate vicinity (8-12) connected via an edge gateway, that forwards the data to a single fog server located on-premises. The decisions support layer is setup to include asset-criticality weightings based on the overall layout of the line, with conveyor drive units (as there is no redundancy and these units directly affect the line's overall throughput) being weighted as highest-criticality, and pumps (which have partial redundancy in the cooling subsystem) weighted as medium-

criticality, while CNC spindle motors (which support parallel machining stations) are weighted as medium-to-high criticality, depending on the current production schedule.

6.3 Illustrative Results

During a simulated 12-month operating time, the deployed framework reached a proper maintenance work order with an average time of 6.4 days before the simulated failure occurred, which is generally adequate to schedule maintenance during planned stoppages (e.g. during shift changeover or during the weekend) without the need for an immediate unscheduled stoppage. Over the course of the simulation, 91.3% of the faults were correctly detected and classified before they caused a functional failure, 6.2% were detected but not classified as the correct fault type (mostly rare fault types with a small number of examples in the fault set used for training, which are discussed further in Section 8), and 2.5% were missed entirely.

6.4 Cost-Benefit and Return on Investment (ROI) Analysis

Table 13 shows an illustrative cost-benefit comparison between the simulated baseline (fixed-interval servicing) and the proposed framework (predictive-maintenance) on the basis of the simulated 12-month operating period for the case-study. The figures given are not necessarily based on a deployment, but are taken from the simulated testbed described in section 4.1, and are meant to be indicative of the direction and order of magnitude for potential benefits rather than validated financial projections for any particular facility.

Table 13. A simulated case study comparing 12-months cost-benefit of preventive maintenance baseline and proposed AIoT-PdM framework.

Plan for current condition of the system.Take proactive action to maintain the condition of the system.

Unplanned downtime (hours/year)	**168**	**108–114**	**−32% to −36%**
Unplanned downtime cost (estimated, $/year)	$1,428,000	$918,000–$969,000	−32% to −36%
Scheduled maintenance labor hours (hours/year)	2,340	1,850–1,970	−16% to −21%
Unnecessary component replacements per year (count/year)	47	12–18	−62% to −74%
Sensorial infrastructure investment (one time + annual)	Minimal (existing instrumentation)	$180,000 (one time) + $35,000/year (maintenance)	Additional investment is needed

Positive net benefit is expected after 1 year infrastructure investment: Estimated net annual benefit: $280,000-$390,000

The simulated case study indicates that, despite the one time costs of the sensing and infrastructure deployment and the costs of maintaining the system, there is a positive net financial benefit within the first year of operation, largely owing to reduced costs of unnecessary component replacement and reduced cost of unplanned downtime. The cost-benefit analysis described in this case study is indicative of the overall economic rationale behind predictive maintenance and should be tailored to the specific facility and organization to reflect their downtime cost, labour cost and spares cost data which varies considerably between industry and facility.

7. Questions and Answers

7.1 Sensor Retrofit and Instrumentation Strategy

Many of the existing production lines were not constructed with dense instrumentation for the Internet of Things, and for some sensors there are practical implementation issues, such as the need to coordinate existing electrical-safety procedures with the integration of current-clamp sensors into existing electrical panels; or suitable positions for vibration and acoustic-emission sensors in existing equipment geometry. If the organization chooses to roll-out instrumentation in a phased approach, they could deploy instrumentation to the highest criticality assets identified in the priority-weighting process outlined in Section 3.7 and gain experience and value before deploying instrumentation fleet-wide.

7.2 Integration with the existing CMMS/MES Systems

The usefulness of the decision-support layer presented in Section 3.7 will rely on the integration of this layer with the organization's existing Computerized Maintenance Management System (CMMS) and Manufacturing Execution System (MES). This usually involves the integration of the AIoT-PdM work-order outputs to the organization's CMMS/MES data schema and API (and, perhaps, the file-based interface). This is often understated in academic studies on predictive maintenance, and it has a significant impact on both deployment timelines and cost.

7.3 Data Governance and Model Retraining.Data Governance and Model Retraining 7.3

Continuous data governance is also important for the periodic re-training of prediction models described in section 3.4 and to inform model performance checkpoints over time as equipment ages and conditions vary (false/true positive rate and correct fault type classification). Organizations should set up a feedback loop where the maintenance techs document actual results from work orders that are triggered, to help retrain and gain confidence in the system's recommendations.

7.4 This unit covers organizational readiness and change management.In this Unit, you will learn about organizational readiness and change management.

Implementing a predictive-maintenance framework often involves modifying existing maintenance-planning workflows as well as any necessary changes to the role of maintenance personnel and the skills they need to acquire (such as knowing how to interpret model outputs and levels of confidence instead of a set maintenance-planning calendar). If the organisation has had little prior experience with data-driven maintenance, it might be beneficial to begin by using the framework alongside current preventive-maintenance schedules and incrementally replace the latter with the predictive ones as confidence in the accuracy of the predictive entries increases and familiarity with the outputs of the predictive maintenance increases.

7.5 Cybersecurity Considerations

Securing the IoT sensing and edge-cloud communication layer of the AIoT-PdM framework requires standard industrial-cybersecurity measures (e.g., network segmentation of OT and IT networks, encryption between layers, and authentication of edge and fog devices), which are out of the scope of this paper's technical contribution, but are necessary preconditions for safe production deployment, especially for higher-criticality production lines.

8. Limitations

There are a number of limitations in this study. A large percentage of the joint classification/RUL evaluation, scalability analysis, and cost-benefit case study is based on a simulated multi-asset factory-floor testbed and not real-world data from an operating production line; although the testbed was calibrated to match the statistical characteristics reported in the published PdM literature, the results should be interpreted as indicative of relative system behaviour, and not as validated real-world scenario behaviour. Second, the public benchmark datasets used (NASA C-MAPSS, CWRU) cover only certain classes of equipment (turbofan engines and rolling-element bearings), which may not be representative of the variety of equipment found in a real discrete- or process-manufacturing plant. Third, the framework's performance on rare fault types, which are poorly represented on the training data, was measurably poor (Section 6.3), a type of class-imbalance and label-scarcity challenge shared by the vast majority of the fault-diagnosis

literature and not solved by the proposed architecture. Fourth, the cost / benefit analysis presented in Section 6.4 relies on facility-specific cost parameters (downtime cost, labour cost, spare parts cost) which are likely to have significant differences between industry sectors, and values should not be assumed to be universally applicable projections. Fifth, the study did not directly assess cross-machine and cross-facility transfer learning, which is another unanswered question for deployment cost and scalability, where a machine or facility trained model can be transferred to a different but similar machine or facility with minimal retraining.

Conclusion and Future Work 10.

This paper introduced the unified AI-IoT framework for predictive maintenance in smart manufacturing systems (AIoT-PdM), which consists of a multi-modal IoT sensing layer, an edge-cloud computational hierarchy, a hybrid CNN-LSTM-Attention model for jointly performing fault classification and Remaining Useful Life estimation, and a decision-support layer transforming model outputs into resource-aware prioritized maintenance work orders. The suggested model was tested on NASA C-MAPSS and CWRU public benchmark datasets as well as on a simulated multi-machine testbed on the factory floor of CWRU and it performed a fault-classification accuracy of 98.1%, RMSE reduction of 17.4–17.6% compared to a standalone LSTM baseline, edge-tier inference latency of less than 45 milliseconds, and a nearly linear scaling of processing time in the fog-tier with the number of monitored assets. The use of an illustrative case study and cost-benefit analysis indicated that the framework would provide a positive net financial benefit in the first year after deployment with respect to a baseline preventive-maintenance approach, and would reduce unplanned downtime by 32-36% as well. However, these results need to be validated in a field deployment.

Future research will be carried out in five directions. First, the framework has to be tested in a real production line, and not in the simulated factory-floor testbed employed in this study, but rather by using actual sensors and verified outcomes of maintenance, to verify the technical performance and cost-benefit projections made here. Third, a transfer-learning approach to cross-machine and cross-facility should be explored to decrease the need for labeled-data to extend the framework to new asset types and facilities, as well as the deployment cost. Third, the scarcity of labels for rare fault types, such as generating synthetic fault data, few-shot learning and physics-informed data augmentation, is a topic that deserves specific study as it has been observed to have a measurable difference for rare fault types as shown in Section 6.3. Finally, physics informed and digital-twin models, especially for expensive, well characterised equipment, could be more generalisable and yield more interpretable explanations of equipment failure mechanisms than the purely data-driven model assessed in this paper. Fifth, there is a need for longitudinal field studies over several years of operation to evaluate model performance drift with equipment age and to fine-tune model priority-weighting and decision-support processes as detailed in Section 3.7, based on actual maintenance-planner feedback. AIoT-PdM is not a fully tested and proven solution, but a reference architecture and assessed initial point of departure for researchers and practitioners working on smart manufacturing environment AI-IoT predictive-maintenance systems.

Reproducibility Statement

The key datasets presented in Sections 4–5 are publicly available benchmark datasets that are well-known in the prognostics and health management literature: NASA C-MAPSS and CWRU bearing. The generation procedure of the simulated factory-floor testbed, hyperparameter setting (Table 5), and the assumptions for edge-cloud hardware (Table 3) are described in detail enough to enable independent re-implementation. Researchers who follow the reported findings of Section 5.6, 5.7, and 6.4 with respect to scalability, robustness and cost-benefits are encouraged to test these results with data from an operational production line, and not to rely on the specific numbers reported here.

Summary of Notation (Appendix A)

Table A1. Notation summary for a description of the framework.

Symbol	Meaning	
x(t)	Raw multi-modal sensor observation at the time t of a particular asset	
L	The length of the window for feature extraction and CNN input, with it being a sliding window.	
e(w)	Embedding vector for window w of CNN model	
S	Sequence of window embeddings covering history of LSTM's input data	
h_t	The hidden state at time step t in the LSTM network.h_t	Hidden state at time t for the LSTM network.
w_t	Weight for time step t.	
rw	Representations that are not attention-weighted.rw	Representations that are not attention-weighted.
ŷ_class	Class predicted with softmax output.	

The predicted Remaining Useful Life (RUL) is output of regression.The predicted Remaining Useful Life (RUL) is a regression output.

L_total	Sum of training loss (weighted classification + regression loss)

Realized the importance of how this was used in the classification and regression tasks.Understood the significance of this in classification and regression.

Priority score for maintenance of asset i.

N_i	Number of assets in production line i that fail to achieve the production target.N_i	Number of failures to the production target for each asset i in the production line.

REFERENCES

1. R. K. Mobley, An Introduction to Predictive Maintenance, 2nd ed. Amsterdam: Butterworth-Heinemann, 2002.
2. A. K. S. Jardine, D. Lin, and D. Banjevic, "A review on machinery diagnostics and prognostics implementing condition-based maintenance," Mechanical Systems and Signal Processing, vol. 20, no. 7, pp. 1483–1510, 2006.
3. T. P. Carvalho, F. A. A. M. N. Soares, R. Vita, R. da P. Francisco, J. P. Basto, and S. G. S. Alcalá, "A systematic literature review of machine learning methods applied to predictive maintenance," Computers & Industrial Engineering, vol. 137, art. 106024, 2019.
4. T. Zonta, C. A. da Costa, R. da Rosa Righi, M. J. de Lima, E. S. da Trindade, and G. P. Li, "Predictive maintenance in the Industry 4.0: A systematic literature review," Computers & Industrial Engineering, vol. 150, art. 106889, 2020.
5. 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," Mechanical Systems and Signal Processing, vol. 104, pp. 799–834, 2018.
6. H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, and M. Hoffmann, "Industry 4.0," Business & Information Systems Engineering, vol. 6, no. 4, pp. 239–242, 2014.
7. E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial Internet of Things: Challenges, opportunities, and directions," IEEE Transactions on Industrial Informatics, vol. 14, no. 11, pp. 4724–4734, 2018.
8. W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," IEEE Internet of Things Journal, vol. 3, no. 5, pp. 637–646, 2016.
9. F. Tao, H. Zhang, A. Liu, and A. Y. C. Nee, "Digital twin in industry: State of the art," IEEE Transactions on Industrial Informatics, vol. 15, no. 4, pp. 2405–2415, 2019.

10. 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,” Robotics and Computer-Integrated Manufacturing, vol. 61, art. 101837, 2020.
11. A. Saxena, K. Goebel, D. Simon, and N. Eklund, “Damage propagation modeling for aircraft engine run-to-failure simulation,” in Proc. Int. Conf. Prognostics and Health Management (PHM08), Denver, CO, 2008.
12. W. A. Smith and R. B. Randall, “Rolling element bearing diagnostics using the Case Western Reserve University data: A benchmark study,” Mechanical Systems and Signal Processing, vol. 64–65, pp. 100–131, 2015.
13. S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.
14. Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, pp. 436–444, 2015.
15. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in Advances in Neural Information Processing Systems (NeurIPS), vol. 30, 2017.
16. D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” in Proc. Int. Conf. Learning Representations (ICLR), 2015.
17. S. Han, H. Mao, and W. J. Dally, “Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding,” in Proc. Int. Conf. Learning Representations (ICLR), 2016.
18. R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, and R. X. Gao, “Deep learning and its applications to machine health monitoring,” Mechanical Systems and Signal Processing, vol. 115, pp. 213–237, 2019.
19. G. Sateesh Babu, P. Zhao, and X.-L. Li, “Deep convolutional neural network based regression approach for estimation of remaining useful life,” in Proc. Int. Conf. Database Systems for Advanced Applications (DASFAA), 2016, pp. 214–228.
20. J. Li, X. Li, and D. He, “A directed acyclic graph network combined with CNN and LSTM for remaining useful life prediction,” IEEE Access, vol. 7, pp. 75464–75475, 2019.