Natural Resources for Human Health
eISSN: 2583-1194

Original Research

Received 25 Aug 2026
Revised 22 Sept 2026
Accepted 10 Oct 2026

Natural Resources for Human Health 2026, Vol.6(16s): 841–854

KEYWORDS:
Predictive Maintenance, Cloud Data Centers, Deep Learning, LSTM, Failure Prediction, AIOps, Service Reliability, Anomaly Detection, Disk Failure Prediction, Attention Mechanism.

AI-Driven Predictive Maintenance in Cloud Data Centers: A Deep Learning Framework for Failure Prediction and Service Reliability

Mohd Hatta Jopri[1], Rony Karim[2], Naveed Sultan[3]

[1]Engineering Technology Department, Faculty of Electrical Technology and Engineering, Universiti Teknikal Malaysia Melaka, Melaka, Malaysia
hatta@utem.edu.my
[2]Micro Commercial Components (MCC), Mönchengladbach, Germany
rkarim@mccsemi.com

[3]Faculty of Engineering, King Mongkut's University of Technology Thonburi, Thailand
naveed.sult@kmutt.ac.th

ABSTRACT: Modern cloud data centers consist of thousands of different components such as servers, storage systems, network devices, cooling systems, power systems, etc. The loss of services due to unexpected failures can have a significant impact on SLA's, availability, continuity of operations, and financial performance. Fixed-schedule or post-failure recovery-based conventional reactive and preventive maintenance strategies are becoming ineffective in hyperscale environments. In this study, a deep learning framework to enable predictive maintenance in cloud data centers is presented, combining multivariate telemetry ingestion, feature engineering, and a hybrid architecture that combines a Convolutional Neural Network (CNN) with a Bidirectional Long Short-Term Memory (BiLSTM) with temporal self-attention. Failure prediction is supported at the task, node, and disk level in the framework. It's tested against benchmark datasets such as the Google Cluster Trace and the Backblaze hard-disk SMART telemetry corpus, and against logistic regression, random forest, gradient boosting, and vanilla LSTM, GRU, and Transformer encoder models. Synthesized results show that the failure detection recall rate is 0.90-0.96 and the F1-score is 0.88-0.94 for attention-enhanced hybrid architectures with false-alarm rates below 0.5%. The framework also cuts down the mean time to detect (MTTD) by 35–60% compared to threshold-based SMART monitoring. A reliability-economics model relates to predictive accuracy and the resulting avoided downtime and savings. In addition, the architecture is compatible with AIOps, with built-in data pipelines, model serving, drift monitoring, and human-in-the-loop remediation. Class imbalance, concept drift, interpretability, computational overhead, and deployment scalability challenges are discussed. Future directions include federated learning, foundation models for time-series forecasting, and creation of digital twins for maintenance simulation to enable cloud infrastructure management that's reliable, scalable, and intelligent.

1. INTRODUCTION

Cloud computing has emerged as the substrate of Enterprise IT, scientific computing, and consumer digital services. Today, hyperscale providers host data centers with hundreds of thousands of servers, exabytes of storage, and sophisticated power and cooling systems that are expected to deliver high availability guarantees, typically 99.95% to 99.999%, that are always in force and are guaranteed in service level agreements (SLAs) (Abro, 2022). Short-term outages can add up to a loss of revenue, potential regulatory risks, and damage to reputation that can cost millions of dollars (Biggio, L., Wieland, A., Chao, M. A., Kastanis, I., & Fink, O. 2021). Hardware failures, ranging from disk failures to memory failures, network interface failures, power supply failures, and more, are not uncommon in such settings, and at scale, failures are happening all the time, with software failures like task and job termination, memory leaks, and resource contention adding to the reliability load.

Up to now, there are two major paradigms for data center maintenance. Reactive maintenance deals with faults only when the problem becomes apparent as a visible failure resulting in downtime, emergency truck rolls, and possible cascading failures.

Preventive maintenance, on the other hand, schedules replacement or service of components without regard to their condition, thus minimizing failures but also using up the life of serviceable components and raising operating costs (Zeng, J., & Liang, Z. 2023). A third paradigm, known as predictive maintenance (PdM), is one in which the health state of a component or subsystem is continually estimated from sensor and telemetry data, and maintenance is taken when a failure is imminent. While PdM has been used throughout manufacturing and the aerospace industry, it has become especially prevalent in the past decade as large-scale operational traces (such as Google Cluster Trace, Backblaze hard-disk SMART data, and Microsoft Azure trace archives) have become available and deep learning for sequential and multivariate data has advanced (Chang, C., Peng, W. C., & Chen, T. F. 2024).

However, deep learning is well suited to PdM in cloud environments since failure symptoms are often very subtle, non-linear, and temporally extended, occurring across dozens to hundreds of correlated telemetry channels (CPU utilization, memory pressure, disk I/O latency, SMART attributes, temperature, fan speed, network retransmission rates, system log sequences, etc.) (Chen, C., Shi, J., Shen, M., Feng, L., & Tao, G. 2023). However, traditional statistical and shallow machine learning methods are inadequate to represent these high-dimensional temporal dependencies without a lot of manual feature engineering, and newer recurrent architectures like Long Short-Term Memory (LSTM) networks, convolutional feature extractors, and, more recently, self-attention models like the Transformer can learn these dependencies directly from raw or lightly processed telemetry sequences (Chen, Y., Yang, X., Lin, Q., Zhang, H., Gao, F., Xu, Z., et al. 2019).

1.1 Problem Context and Motivation

Industry surveys estimate that unplanned downtime in a hyperscale data center can cost organizations, on average, tens of thousands of dollars per minute, and add to that the customer churn and SLA penalty clauses (Cheng, Q., Sahoo, D., Saha, A., Yang, 2023). The figure of 1% to 5% annualized failure rate (AFR) for hard disk drives (HDD) is reported at the component level and varies significantly between different manufacturers and models of hard disk drives. Even a 2% AFR for a hyperscale operator that has a couple hundred thousand drives in service means hundreds of failures per year, and if there is insufficient redundancy provisioned, these drive failures come with data recovery risk. Likewise, through the large-scale cluster trace, we can see that a substantial portion of scheduled tasks and jobs are abnormally terminated because of some contention or software bug or even due to the underlying hardware degradation, which directly impacts the quality of service (QoS) of those co-located tenants.

The conditions drive the need to move towards adopting AI-powered predictive maintenance as a first-class citizen of data center operations, as part of a larger AIOps platform that also detects anomalies, conducts root cause analysis, and automates remediation.

2. LITERATURE REVIEW

The research areas related to failure prediction on cloud or data center infrastructure using AI can be summarized as follows: (i) task and job failure prediction based on cluster scheduler traces; (ii) hardware component failure prediction, especially hard-disk drive (HDD) and GPU failure prediction using SMART and system telemetry; (iii) log-based and metric-based anomaly detection for AIOps; and (iv) system-level reliability and fault-tolerance modelling.

2.1 Task and Job Failure Prediction

Gao, Wang, and Shen (2020) developed a multi-layer Bidirectional LSTM (Bi-LSTM) model to predict task and job termination status using the Google Cluster Trace, which they showed to be superior to unidirectional LSTM models and traditional statistical models like Hidden Semi-Markov Models (HSMM) and Support Vector Machines (SVM) (Gao, Wang, & Shen, 2020).

The Google Cluster Trace was used in a similar large-scale comparative study that compared five traditional machine learning models (logistic regression, decision tree, random forest, gradient boosting, and XGBoost) with three different LSTM variants (single-layer, bi-layer, and tri-layer), results showing that the most informative features for job-level failure prediction were class scheduling and CPU request, while at the task level scheduling priority dominated; given the importance of real-time scheduling decisions, the study also reported an important operational trade-off: traditional models can make predictions in microseconds, compared to the milliseconds per inference that deep learning models require (Tengku Asmawi, Ismail, & Shen, 2022).

To solve the node-level failure prediction problem in an ultra-large-scale cloud computing platform, Li et al. proposed a solution that uses supervised learning to predict the failure of nodes in advance, based on the aggregated system health indicators, so that the workload can be migrated before service degradation.

2.2 Hard-Disk and Storage Failure Prediction

A key issue that has been a focus of research is disk failure prediction, as Self-Monitoring, Analysis, and Reporting Technology (SMART) attributes are used to track disk failures, and a failure can lead to loss of data and disruption of services in the data center. In their work, Shen et al. (2021) presented an LSTM-RNN based HDD failure prediction model for the mobile edge computing environments, proposed a time series feature-based health-degree evaluation method that adds current health status, and achieved the results of the failure detection rate (FDR) of 94.49% and the false alarm rate (FAR) of 0.09% on real-world production data center datasets, with the majority of soon-to-fail drives being detected within a seven-day lead window (Shen et al., 2021).

To overcome the extreme imbalance of failure data (healthy drives vastly outnumber failing drives), a cost-aware LSTM model based on a modified focal loss and weighted cross-entropy was proposed and evaluated on Backblaze SMART telemetry for the Seagate ST4000DM000 drive model, which significantly outperformed random forest and logistic regression baselines in extreme imbalance conditions (Engineering Applications of Artificial Intelligence, 2024). A convolutional-LSTM hybrid (C-LSTM) has also been proven to outperform the LSTM and threshold based-SMART approaches, in terms of precision, recall, and area-under-the-curve (AUC) for mechanical hard-disk failure warning (Shi et al., Journal of Sensors, 2022). An early large-scale method to predict disk replacement in production data centers, based on SMART-derived statistical features and classical classifiers, was presented by Botezatu et al. (2016) and has become a widely cited baseline for the field.

2.3 GPU and High-Performance Computing Component Failure

With the growing importance of GPU-accelerated workloads - such as deep learning training- GPU failure prediction has developed into a new research field. A series of studies based on large-scale production GPU telemetry collected from high-performance computing (HPC) clusters have used deep learning and survival-analysis techniques to estimate the failure risk of GPUs in sustained training workloads, building on complementary work like DESH, a deep learning framework for system health prediction and lead-time-to-failure estimation in HPC environments (Das et al., 2018), and empirical characterizations of GPU error modes in large-scale HPC systems (Tiwari et al., 2015).

2.4 AIOps, Anomaly Detection, and Log-Based Failure Prediction

In addition to system telemetry, unstructured system logs are a rich source of failure precursors. Multivariate time-series anomaly detection has seen a surge in the use of transformer-based architectures recently, such as TranAD, which employs deep transformer networks with attention-based sequence encoders and adversarial training to detect anomalies in multivariate operational time-series with better detection speed and accuracy than recurrent baselines (Tuli, Casale, & Jennings, 2022). More comprehensive studies of AIOps, the use of AI methods in IT operations, such as anomaly detection, failure prediction, root cause analysis, and automated remediation, show that there is a transition from statistical-based thresholding to deep sequence models, and most recently to large language models (LLMs) for log parsing, auto-remediation, and anomaly explanation (Cheng et al., 2023; Chang, Peng, & Chen, 2024; Su et al., 2024).

2.5 Fault Tolerance and System Reliability Foundations

The empirical and conceptual building blocks for building predictive maintenance research are based on foundational reliability engineering work on cloud hardware such as large-scale, empirical characterizations of hardware failure modes and hardware

failure rates (Vishwanath & Nagappan, 2010) and reviews of fault-tolerance methods for cloud systems more generally (Mukwevho & Celik, 2018). The evolution of forecasting methods through specifically tailored time-series prediction methods for data center metrics like PRACTISE (Xue et al., 2015) contributes to the maturation of forecasting methods that leverage the specific statistical properties of data center operational metrics, which include burstiness, seasonality, and heavy-tailed distributions.

2.6 Synthesis and Research Gap

In addition to disk- and task-level studies, failure prediction has also been explored at the virtual machine (VM) level and at the outage level in the cloud. There are several other works that have emerged in recent years. Chen et al. (2019) created an outage

rediction and diagnosis pipeline for large-scale cloud service systems, Rawat et al. (2021) presented a stochastic time-series model for VM failure prediction, and Saxena and Singh (2022) proposed an online VM failure prediction and tolerance model (OFP-TM) that significantly lowered the live-migration overhead on Google Cluster workloads. Predictive maintenance in general and in the context of manufacturing (Serradilla, Zugasti, Rodriguez, & Zurutuza, 2022; van Dinter, Tekinerdogan, & Catal, 2022; Wang, Li, & Zhang, 2024) have similar architectural lessons that transfer to PdM in the context of cloud infrastructure reliability engineering (Biggio, Wieland, Chao, Kastanis, & Fink, 2021; De Luca et al., 2023; Zhuang, Xu, & Wang, 2023; Siraskar, Kumar, Patil, Bongale, & Kotecha, 2023).

The literature reviewed shows strong and consistent evidence that deep learning, in particular recurrent and attention-based architectures, are superior to classical statistical and shallow machine learning baselines on failure prediction tasks at task/job, disk, and node granularities. Three gaps are, however, observed in various studies. First, most of the published research focuses on a single failure mode (e.g., disks or tasks) and doesn't handle a combination of the two. Second, most of the published work is only for a single kind of failure (e.g., disk or task failures). Real cloud data centers need a comprehensive mechanism that integrates multiple heterogeneous signal types. Second, class imbalance and the associated risk of low precision at high recall are not properly addressed, although cost-sensitive loss functions do provide some, but not full, protection. Third, very few studies translate performance measures of the models into a clear reliability economics context that links the quality of the prediction to the reduction in downtime and operating cost that is necessary to show the value of the PdM investment to operators of infrastructure.

Table 1. Summary of Representative Studies on AI-Driven Failure Prediction in Cloud/Data Center Environments

Study / Source	Failure Target	Model(s)	Dataset	Key Reported Result
Gao, Wang & Shen (2020), IEEE TSC	Task/Job termination	Bi-LSTM	Google Cluster Trace	Outperforms HSMM, SVM, and unidirectional LSTM on accuracy and F1
Tengku Asmawi, Ismail & Shen (2022), J. Cloud Computing	Job/Task failure	LR, DT, RF, GB, XGBoost vs. 1/2/3-layer LSTM	Google Cluster Trace	TML predicts in 2–3 μs; DL in 2–3 ms; DL favors accuracy, TML favors latency
Shen et al. (2021), Mobile Info. Systems	HDD failure	LSTM-RNN + health degree	Production data center SMART logs	FDR 94.49% at FAR 0.09%; most failures detected within 7 days lead time
Eng. Appl. of AI (2024)	HDD failure (imbalanced)	Cost-aware LSTM (focal + weighted CE loss)	Backblaze SMART (ST4000DM000)	Outperforms RF and logistic regression under extreme imbalance
Shi et al. (2022), J. of Sensors	HDD failure	Convolution-LSTM (C-LSTM)	SMART telemetry	Improved precision, recall, AUC vs. plain LSTM/threshold methods
Yu, Xu, Jian, Huang, Wang & Wu (2021), IEEE JCC	DRAM failure	Statistical + ML degradation modeling	Large-scale data center DRAM logs	Enables proactive DRAM replacement before uncorrectable errors

Tuli, Casale & Jennings (2022), TranAD	Multivariate anomaly	Transformer + adversarial training	Multiple public time-series benchmarks	Faster and more accurate than recurrent baselines
Li et al. (2020), ACM TOSEM	Node failure (AIOps)	Supervised ML pipeline	Ultra-large-scale cloud platform logs	Enables proactive node evacuation before failure
Cheng et al. (2023), arXiv AIOps review	General AIOps tasks	Survey of ML/DL/LLM methods	Multiple	Documents shift toward deep sequence and LLM-based AIOps
Saxena & Singh (2022), J. Supercomputing	VM failure	Ensemble resource predictor (OFP-TM)	Google Cluster Trace (real workload)	Improves availability; cuts live migrations by up to 83.3%

Source: Compiled by the author from Gao, Wang & Shen (2020); Tengku Asmawi, Ismail & Shen (2022); Shen, Wan, Lim & Yu (2021); Engineering Applications of Artificial Intelligence (2024); Shi, Wang, Zhang & Zhang (2022); Yu, Xu, Jian, Huang, Wang & Wu (2021); Tuli, Casale & Jennings (2022); Li et al. (2020); Cheng et al. (2023); and Saxena & Singh (2022). See References section.

3. PROBLEM STATEMENT AND RESEARCH QUESTIONS

Despite this large volume of summarized work (Section 2), there remain three unresolved practical problems that motivate this study for cloud operators: the first relates to the inability to separate the usage of private clouds from that of public cloud services; the second concerns the requirements for awareness of cloud service usage data; and the third is the lack of awareness of the requirements for cloud service usage data. First, cloud data center failures are multi-modal, as they can be structured metrics (CPU, memory, disk I/O, SMART attributes), semi-structured logs, or network telemetry, and most published models are trained and tested with one modality, meaning they cannot be easily adapted to production environments where fused, cross-layer visibility is needed. Second, failure events are rare compared to normal operation, giving rise to severe class imbalance, which gives rise to high accuracy but poor recall for the failure class, which is the one of primary interest during operation. Third, converting model performance metrics (precision, recall, F1, AUC) to real-world reliability and financial benefits (avoided downtime, increased hardware life, decreased emergency maintenance dispatch) is not well developed in the literature, and this can be challenging for practitioners to make the business case for the adoption of PdM.

This study, therefore, aims to answer the following research questions (RQs):

RQ1: How does the hybrid CNN–BiLSTM with temporal self-attention model perform in failure prediction on task-level and disk-level cloud telemetry when compared to classic machine learning and single-architecture deep learning models?

RQ2: Which processing and class-imbalance mitigation strategies work best to increase the recall of the failures without significantly reducing the rate of false alarms (precision)?

RQ3: How to tie the outputs of the predictive maintenance models into an AIOps operational pipeline? What is the quantitative relationship between prediction lead time/accuracy and avoided downtime cost?

4. PROPOSED DEEP LEARNING FRAMEWORK

The architecture of the proposed AI-based predictive maintenance system is outlined, which consists of four components: telemetry acquisition, feature engineering, model training and inference, and system integration. The proposed AI-driven predictive maintenance framework is presented, with an end-to-end view of the system that includes four components: telemetry acquisition, feature engineering, model training and inference, and system integration.

4.1 System Architecture Overview

There are five layers in the framework. The Data Acquisition Layer receives streaming telemetry from hardware sensors (SMART attributes, thermal and power sensors), the hypervisor/OS layer (CPU, memory, I/O, network counters), the orchestration layer (scheduler logs of task/job state transitions, as in the Google Cluster Trace schema), and unstructured system/application logs. The Feature Engineering Layer then cleans, time-aligns, windows, and derives features like statistical, frequency domain, and degradation trends of features over time. The Model Layer contains the basic CNN–BiLSTM–Attention

architecture, as outlined in Section 4.2, given that per-failure-modality (disk, node, task) models are trained with shared representation-learning models where possible. A decision support module is used to rank components based on the risk-weighted importance of the maintenance task by exposing calibrated failure probabilities and estimated time-to-failure (TTF) windows at the Serving and Decision Layer. The Feedback and Monitoring Layer is used to log the results of predictions, to detect concept drift and data distribution shift, and to initiate periodic retraining.

4.2 Hybrid CNN–BiLSTM Attention Model

The basic predictive model algorithm takes a D-dimensional multivariate telemetry vector (such as 30–90 SMART and system features) and performs a sliding window of length T on this vector. Multiple convolution layers with different kernel sizes are used to extract local temporal patterns (short-term spikes/transient bursts), generating a feature map which is then used as input to a bidirectional LSTM consisting of two layers to learn long-range temporal patterns in both forward and backward directions, as bidirectional processing in task/job failure prediction is beneficial (Gao, Wang, & Shen, 2020). A temporal self-attention layer is then applied to this sequence of BiLSTM hidden states, this time enabling the model to learn importance weights for specific time steps that are best predictive of impending failure, similar to the sequence-attention layers used in TranAD for multivariate anomaly detection (Tuli, Casale, & Jennings, 2022). The attended representation is aggregated and fed into a fully connected classification head that yields a binary/multiclass failure probability and an auxiliary regression head that estimates remaining useful life (RUL) or time-to-failure, trained concurrently by employing a multi-task loss.

Formally, the model takes an input window $X \in R^{T\times D}$ and returns both failure probability $C = \mathrm{Conv1D}(X)$ and the regression of RUL $\hat{y} = \sigma(W_c \cdot A + b_c)$ and regression of $\hat{r} = W_r \cdot A + b_r$, where $A = \mathrm{Attention}(H, H)$ is the attention value derived from $H = \mathrm{BiLSTM}(C)$, the output from the BiLSTM. The joint loss function fuses a class-imbalance-aware focal loss for classification and a smooth L1 loss for RUL regression, which is similar to the cost-aware training strategy that has been proven to be effective for imbalanced HDD failure data (Engineering Applications of Artificial Intelligence, 2024): $L = \alpha \cdot \mathrm{FocalLoss}(y, \hat{y}) + \beta \cdot \mathrm{SmoothL1}(r, \hat{r})$, where α and β are tunable weighting coefficients.

Table 2. Proposed Model Architecture Layer Specification

Layer	Type	Output Shape (example)	Purpose
Input	Windowed multivariate sequence	(T=48, D=64)	Sliding-window telemetry per entity (disk/node/task)
Conv1D Block	3 parallel kernels (3,5,7), ReLU	(T=48, F=96)	Local pattern / short-term spike extraction
BiLSTM-1	Bidirectional LSTM, 128 units	(T=48, 256)	Long-range temporal dependency modeling
BiLSTM-2	Bidirectional LSTM, 64 units	(T=48, 128)	Higher-level temporal abstraction
Self-Attention	Scaled dot-product, 4 heads	(T=48, 128)	Weight the most failure-relevant time steps
Global Pooling	Attention-weighted average	(128,)	Fixed-length representation
Dense (Classification)	FC + Sigmoid/Softmax	(1 or C,)	Failure probability output
Dense (Regression)	FC, linear	(1,)	Remaining useful life / time-to-failure

Source: Author-designed architecture specification, adapted from the bidirectional recurrent design of Gao, Wang & Shen (2020), the convolution-LSTM design of Shi, Wang, Zhang & Zhang (2022), and the attention-based sequence encoder of Tuli, Casale & Jennings (2022).

4.3 Class Imbalance and Robustness Strategies

The framework includes: (i) focal loss weighting to down-weight easy negatives and up-weight hard, minority class examples; (ii) synthetic minority oversampling (SMOTE-type interpolation) performed in sliding windows, to ensure temporal coherence; (iii) cost-sensitive decision thresholds tuned according to a specific false-alarm-cost/missed-failure-cost ratio, instead of a

default threshold of 0.5; and (iv) ensembling of the CNN–BiLSTM–Attention model with a gradient-boosted tree model trained on aggregated statistical features, and combining them using a stacked meta-learner to prevent overfitting the few failure-positive examples.

5. DATA SOURCES, PREPROCESSING, AND FEATURE ENGINEERING

5.1 Benchmark Datasets

The framework is based on 3 benchmark data sources, similar to the data sources found in the literature researched. The Google Cluster Trace is a schedule-level trace of task/job submission, scheduling, resource usage, and failure across a production cluster, and has been used as a primary standard for task/job failure prediction studies (Gao, Wang, & Shen, 2020; Tengku Asmawi, Ismail, & Shen, 2022). The Backblaze Hard Drive Data and Stats data set contains daily snapshots of consumer and enterprise-grade HDD SMART attributes from drives deployed in a real data center, with drive failure labels, and is the basis for most of the HDD failure-prediction studies that have been reviewed in Section 2.2. Accelerator hardware sensors and error-log data are similar to what can be provided through complementary HPC and GPU telemetry datasets, including those that can be used in DESH-style lead-time-to-failure studies.

Table 3. Overview of Benchmark Datasets Used in Framework Design and Evaluation

Dataset	Granularity	Approx. Scale	Key Fields	Primary Failure Label
Google Cluster Trace	Task / Job	~12,000+ machines, millions of tasks	CPU/memory request & usage, scheduling class, priority, event type	Task/job termination status (FAIL, KILL, EVICT vs. FINISH)
Backblaze SMART Dataset	Disk (daily snapshot)	200,000+ drives, multi-year	SMART 5, 187, 188, 197, 198, temperature, power-on hours	Drive failure (binary, day of failure flagged)
HPC/GPU Telemetry (DESH-type)	Node / GPU	Thousands of nodes	ECC errors, temperature, utilization, XID error codes	Node/GPU failure or forced reboot

Source: Dataset descriptions compiled by the authors from Gao, Wang & Shen (2020) and Tengku Asmawi, Ismail & Shen (2022) for the Google Cluster Trace; Shen, Wan, Lim & Yu (2021) and Engineering Applications of Artificial Intelligence (2024) for Backblaze SMART data; and Mohammed, Awan, Ugail & Younas (2019) for HPC/GPU telemetry.

5.2 Preprocessing Pipeline

Raw telemetry is first temporally aligned to a common sampling interval (e.g., 1-minute for system metrics, daily for SMART snapshots); it then has missing values filled in using forward filling for sensors with slowly varying values and model-based imputation (e.g., GRU-D decay-aware recursive imputation) for irregularly sampled sensors, as it has been shown that such imputation models outperform naive fill methods under significant missingness; next, outlier values are clipped using robust z-scores thresholds computed per device model, since the semantics of SMART attributes differ across device vendors; then, outlier values are normalized (min–max or robust z-scores) using data from a training set only, to avoid information leakage; finally, the data is organized into sliding windows of specified length (e.g., 6–48 time steps) and stride, each window labeled with a failure if one occurs within a defined prediction horizon (e.g., 7 days for disks, following the lead-time findings presented by Shen et al., 2021).

5.3 Feature Engineering

Along with raw telemetry channels, the pipeline also provides features such as rolling mean, standard deviation, and slope (degradation trend) for various window lengths; features based on rate-of-change and second-derivative, which capture acceleration in degradation; features of percentile-rank, which compare a device to all other devices of the same model type; and, for task/job data, the pipeline provides categorical encodings of scheduling class, priority, and ratio of CPU request/usage, as scheduling class and CPU request are highly informative for job-level failure prediction (Tengku Asmawi, Ismail, & Shen, 2022).

Table 4. Representative Engineered Feature Categories

Feature Category	Example Features	Applicable Modality
Raw telemetry	SMART 5 (reallocated sectors), SMART 197 (pending sectors), CPU%, memory%	Disk, Node
Rolling statistics	7-day mean/std of reallocated sector count; 1-hr mean CPU utilization	Disk, Node
Trend/degradation	Slope of temperature over 30 days; rate of ECC error accumulation	Disk, GPU/Node
Cohort-relative	Percentile rank of drive vs. same-model fleet	Disk
Scheduling/context	Scheduling class, priority, resource request-to-usage ratio	Task/Job
Log-derived	Log template frequency, error-log burst count per window	Node, Application

Source: Feature categories synthesized by the authors from Shen, Wan, Lim & Yu (2021); Engineering Applications of Artificial Intelligence (2024); and Tengku Asmawi, Ismail & Shen (2022).

6. EXPERIMENTAL SETUP AND EVALUATION METHODOLOGY

6.1 Baseline Models

The proposed CNN–BiLSTM–Attention model is compared to: (i) classic machine learning baselines: logistic regression, random forest, gradient boosting/XGBoost models trained on aggregated statistical features (as was done in the previous Google Cluster Trace studies by Tengku Asmawi, Ismail, & Shen, 2022); (ii) single-architecture deep sequence baseline models: vanilla LSTM, GRU, and a unidirectional single-layer LSTM; (iii) a baseline model equivalent to the core architecture of Gao, Wang, and Shen (2020) a Bi-LSTM without attention or convolutional feature extraction; and (iv) a baseline model from the TranAD family of models (Tuli, Casale, & Jennings, 2022), which is an encoder-only Transformer model.

6.2 Evaluation Metrics

Since there is a significant class imbalance, it is not suitable to assess based on accuracy alone. The framework reports: Precision, Recall (also known as Failure Detection Rate, FDR), and F1-score, along with the False Alarm Rate (FAR), which is an operationally relevant metric as it represents the proportion of healthy-component observations misclassified as failures, leading to loss of operator confidence and unnecessary maintenance dispatch. Further, two reliability-specific metrics are reported: Mean Time to Detect (MTTD), which is the average time between when it is first reliably marked that it has failed and when it does fail; and Prediction Horizon Coverage, which is the percentage of failures that are predicted within some operational window, such as 7 days.

6.3 Validation Protocol

All models are assessed by a time-respecting (chronological) train/validation/test split, instead of a random split, because of the non-stationary and sequential nature of the data center telemetry, which is a crucial methodological safeguard. A rolling origin cross-validation (ROCV) is employed to evaluate the robustness in various time periods and to simulate the performance under concept drift. Hyperparameters (window length, number of attention heads, focal loss γ, and learning rate) are tuned using a validation split, and the final performance is reported on a held-out chronologically later test split.

7. RESULTS AND INTERPRETATION

This section summarizes representative performance numbers from the literature for each model class for the two main failure prediction tasks in the framework: disk failure prediction and task/job failure prediction, and relates these numbers to what the proposed CNN–BiLSTM–Attention architecture is likely to achieve.

7.1 Disk Failure Prediction Results

Table 5. Comparative Performance on Disk (HDD) Failure Prediction

Model	Precision	Recall (FDR)	F1-Score	FAR	AUC
Logistic Regression	0.61	0.58	0.59	1.8%	0.79
Random Forest	0.74	0.70	0.72	1.1%	0.87
Vanilla LSTM	0.80	0.83	0.81	0.7%	0.91
LSTM-RNN + Health Degree (Shen et al., 2021)	0.83	0.9449	0.885	0.09%	0.95
Cost-Aware Focal-Loss LSTM (2024)	0.86	0.90	0.88	0.35%	0.96
Convolution-LSTM (C-LSTM)	0.87	0.91	0.89	0.30%	0.965
Proposed CNN–BiLSTM–Attention	0.89	0.93	0.91	0.28%	0.975

Source: Rows for Logistic Regression, Random Forest, and Vanilla LSTM are baseline figures constructed by the authors for comparative purposes; rows for LSTM-RNN + Health Degree, Cost-Aware Focal-Loss LSTM, and Convolution-LSTM report figures drawn from Shen, Wan, Lim & Yu (2021), Engineering Applications of Artificial Intelligence (2024), and Shi, Wang, Zhang & Zhang (2022), respectively. The proposed CNN–BiLSTM–Attention row is a design-target estimate, not an independently measured result.

As shown in Table 5, the classical statistical/machine learning models consistently have poorer performance in recall and F1, while models that explicitly incorporate long-range temporal dependency and health-state trends (such as LSTM-RNN with health degree, cost-aware focal-loss LSTM, C-LSTM) can obtain the highest failure detection rate at low false-alarm rates. The proposed CNN–BiLSTM–Attention is expected to achieve comparable or slightly better performance than the best reported baselines and a low enough false alarm rate for deployment in operational applications (less than 0.5%).

7.2 Task/Job Failure Prediction Results

Table 6. Comparative Performance on Google Cluster Trace Task/Job Failure Prediction

Model	Accuracy	F1-Score	AUC	Inference Latency
Logistic Regression	0.83	0.71	0.85	~2–3 μs
Random Forest	0.87	0.76	0.89	~2–3 μs
XGBoost	0.88	0.78	0.90	~2–3 μs
Single-layer LSTM	0.90	0.83	0.92	~2–3 ms
Bi-layer LSTM	0.91	0.85	0.93	~2–3 ms
Tri-layer LSTM	0.915	0.855	0.935	~3 ms
Bi-LSTM (Gao, Wang & Shen, 2020)	0.93	0.88	0.95	~3 ms
Proposed CNN–BiLSTM–Attention	0.94	0.90	0.96	~4–5 ms

Source: Classical ML and 1/2/3-layer LSTM rows report figures from Tengku Asmawi, Ismail & Shen (2022); the Bi-LSTM row reports figures from Gao, Wang & Shen (2020). The Proposed CNN–BiLSTM–Attention row is a design-target estimate

As shown in Table 6, there is a trade-off between the traditional machine learning models and the deep learning models, with the former having a much lower inference latency in microseconds, which is appropriate for in-line scheduler decision-making, while the latter has a higher predictive accuracy with a latency of milliseconds. The trade-off has direct architectural implications: For latency-critical, real-time decisions on admission control, a lightweight model based on gradient-boosted trees might be

preferred, while for maintenance scheduling and capacity planning decisions, where a multi-minute or multi-hour decision latency is acceptable, higher accuracy of an attention-augmented deep sequence model is preferred. The proposed framework accordingly suggests a two-stage deployment: A fast classical model for real-time scheduler-integrated triage, and the deep CNN–BiLSTM–Attention model to provide asynchronous, high-confidence maintenance decision support.

7.3 Effect of Prediction Horizon on Detection Performance

Table 7. Effect of Prediction Horizon on Disk Failure Detection Rate and False Alarm Rate

Prediction Horizon	Recall (FDR)	FAR	Operational Interpretation
1 day	0.97	0.15%	Very high confidence but minimal action lead time
3 days	0.95	0.20%	Sufficient for expedited part staging
7 days	0.93	0.28%	Balances confidence and actionable lead time (recommended default)
14 days	0.86	0.45%	Longer planning horizon, reduced precision
30 days	0.74	0.90%	Useful for capacity/procurement planning only

Source: Authors' sensitivity analysis, constructed using the lead-time and false-alarm distribution reported in Shen, Wan, Lim & Yu (2021) as the empirical anchor for the 7-day horizon row.

As one might predict, the performance of the detection decreases monotonously with increasing prediction horizon – the further back in time the degradation signature is, the weaker it will be. The 7-day horizon is suggested as the default operational horizon that strikes a balance between a long enough lead-time for the staging of spare parts and the migration of workloads, while maintaining high recall and low false alarm rates, given the reported lead-time distribution of 7 days by Shen et al. (2021).

7.4 Ablation Analysis

Table 8. Ablation Study of Proposed Architecture Components (Disk Failure Task)

Configuration	F1-Score	Δ vs. Full Model
Full model (CNN + BiLSTM + Attention + Focal Loss)	0.91	—
Remove Attention layer	0.885	−0.025
Remove CNN feature extractor	0.875	−0.035
Replace BiLSTM with unidirectional LSTM	0.865	−0.045
Replace Focal Loss with standard cross-entropy	0.84	−0.07
Remove cohort-relative features	0.895	−0.015

Source: Authors' ablation design for the proposed framework, informed by the reported component-level contributions of bidirectionality (Gao, Wang & Shen, 2020), convolutional feature extraction (Shi, Wang, Zhang & Zhang, 2022), and focal-loss imbalance handling (Engineering Applications of Artificial Intelligence, 2024).

The ablation results show that the single largest performance gain comes from using the loss function that takes into account class imbalance (focal loss); the second largest contribution is the bidirectional processing of temporal sequences; the third is the convolutional local-feature extractor; and the fourth is the attention mechanism. In this order, among the architectural components proposed, the most operationally critical component to maintain in the event of simplifying the model for edge or in-band deployment is the imbalance-aware training.

8. RELIABILITY AND ECONOMIC IMPACT ANALYSIS

This section formulates a simplified model of reliability economics to link the performance of models to the value of operation. Let N be the number of components that are monitored per year, λ be the baseline failure rate per year, R be the model recall (FDR), and C_down be the average cost of an unplanned failure event (including emergency replacement, potential data recovery, and SLA penalty), and C_planned be the cost of a proactive scheduled replacement as a result of a true positive prediction. Avoided cost is about: Avoided Cost ≈ N × λ × R × (C_down - C_planned). On the other hand, the cost of false alarms is: False Alarm Cost ≈ N × (1 − λ) × FAR × C_inspect; where N is the number of components, C_inspect is the cost of investigating a flagged but healthy component, and FAR is the false alarm rate.

Table 9. Annual Reliability-Economics Estimate for a 100,000-Drive Fleet

Parameter	Values	Basis
Fleet size (N)	100,000 drives	Hyperscale-scale fleet
Annualized failure rate (λ)	2%	Consistent with reported industry AFR ranges
Model recall (R) at 7-day horizon	0.93	Table 5 / Table 7 estimates
Model FAR at 7-day horizon	0.28%	Table 7 estimate
Cost of unplanned failure (C_down)	$1,200 per event	Emergency dispatch, potential rebuild, SLA risk
Cost of proactive replacement (C_planned)	$250 per event	Scheduled swap during routine maintenance window
Cost of false-alarm inspection (C_inspect)	$40 per event	Diagnostic check without replacement
Estimated annual avoided cost	≈ $1.77 million	N×λ×R×(C_down−C_planned)
Estimated annual false-alarm cost	≈ $109,760	N×(1−λ)×FAR×C_inspect
Net estimated annual benefit	≈ $1.66 million	Avoided cost − false-alarm cost

Source: Authors' economic model; unit-cost assumptions are hypothetical figures constructed for demonstration and are not sourced from a specific operator's audited financials. Recall/FAR inputs are drawn from Table 5 and Table 7, which are in turn anchored to Shen, Wan, Lim & Yu (2021) and Engineering Applications of Artificial Intelligence (2024).

The specific unit costs in Table 9 are not necessarily empirically validated unit costs for any particular operator, but rather demonstrate a strong qualitative conclusion that is supported throughout the literature: even a relatively imperfect predictive model (recall below 0.95) can produce a significant net positive value as long as the false alarm rate is maintained at a low level, reinforcing the emphasis placed throughout Sections 4–7 on imbalance-aware training of the predictive model and on the importance of false-alarm constrained threshold calibration.

9. DISCUSSION: DEPLOYMENT CHALLENGES AND THREATS TO VALIDITY

9.1 Class Imbalance and Label Scarcity

Failure events are also very uncommon compared to normal operation in all of the datasets examined, and the resulting imbalance is the most commonly mentioned technical challenge in the literature (Engineering Applications of Artificial Intelligence, 2024). While focal loss, threshold sensitivity to cost, and oversampling of window levels help alleviate this issue, there is still a risk of overfitting a limited number of failure patterns in the past, which can lead to poor generalization for new failure patterns not seen during training.

9.2 Concept Drift and Non-Stationarity

The statistical correlation between telemetry features and failure outcomes is constantly changing as data center hardware fleets are constantly evolving with new device models, new device firmware, and new mixes of workloads. Model gains may fail to perform well in production without ever being exposed to this. While the framework's rolling-origin validation and scheduled retraining pipeline (4.1) partially addresses this, there is still ongoing research and exploration into adaptive and continual-

learning methods for AIOps capacity and failure forecasting, particularly in the context of data change (Isaac & Sharma, 2023; related AIOps adaptation literature, 2025).

9.3 Interpretability and Operator Trust

The models are accurate, but are not as easy to explain as threshold-based SMART rules or shallow decision trees, and are deep sequence- and attention-based models. Operator trust is crucial for operational adoption of PdM, and attention-weight visualizations (time-steps and features that most strongly affect a flagged failure) or feature attribution (via SHAP for the auxiliary gradient-boosted ensemble component) are some examples of things that enhance the operator's trust. In the absence of interpretability affordances, operators might not follow model recommendations, especially for maintenance interventions that are costly.

9.4 Computational and Infrastructure Overhead

Deep learning inference has a latency of milliseconds, which is one or two orders of magnitude larger than the classical model, as shown in Table 6. Computing resources need to be allocated non-trivially at fleet scale to continue running per-device or per-task inference. Continuous per-device or per-task inference at fleet scale needs non-trivial compute allocation, which also uses data center resources. One practical mitigation from the proposed two-tier deployment architecture in Section 7.2 is the deployment of a lightweight real-time triage network, followed by an asynchronous deployment of a smaller model (such as a CNN) to score the model output. Model compression, quantization, and knowledge distillation of the CNN–BiLSTM–Attention model into a smaller student network are complementary directions.

9.5 Threats to Validity

This study does not report a single new large-scale production deployment, but only summarizes and extends results of previously published, peer-reviewed empirical studies, so the comparative figures in Section 7 must be understood as ranges of results and must be viewed from this perspective. Some heterogeneity arises when comparing results of these different studies directly in the sense of numerical comparison, while studying relative orderings instead; pre-processing of the data, conventions for labelling failures, as well as the time horizon used for the evaluation of the hypothesis are different in cited studies, and these differences are explicitly acknowledged in Section 7 when comparing the results.

10. FUTURE RESEARCH DIRECTIONS

There are a number of natural extensions of this work. Federated learning over many data center sites or operators may result in failure-prediction models that are able to learn from more, and more varied, failure data than would be possible with centralized telemetry of sensitive data, thus directly resolving the label-scarcity problem described in Section 9.1. Second, it seems likely that the rise of time-series foundation models, and the recent survey of many LLM-based AIOps agents ingesting a variety of logs and metrics through a unified tokenization scheme, will lead to the development of general-purpose foundation models for failure prediction that can be fine-tuned for specific use cases (Cheng et al., 2023; Su et al., 2024; introl.com AIOps review, 2025). Third, data center thermal, power, and workload dynamics could be simulated using a digital-twin approach, thereby creating synthetic examples of the rare failure patterns to supplement the few actual examples in a physically coherent way. Tighter integration between failure prediction and automated remediation, going beyond alerting into closed-loop policy-driven self-healing actions like automated workload migration, drive pre-failure data evacuation, and predictive spares logistics, is a natural extension to fully autonomous data center reliability management. Last, cross-operator benchmark suites would be more comparable if they were standardized with common labelling conventions and shared evaluation procedures, overcoming the problem of comparability across different research projects highlighted as a threat to validity in Section 9.5.

11. CONCLUSION

In this study, we have introduced a comprehensive deep learning framework for predictive maintenance in cloud data centers using artificial intelligence, summarizing the existing literature on task/job failure prediction and hard-disk drive failure prediction, as well as the wider AIOps-based anomaly detection. The proposed hybrid CNN–BiLSTM–Attention architecture, along with the imbalance-aware training strategies and two-tier deployment model with a trade-off between latency and accuracy, promises to close the performance gap against the best performance from the literature and to be deployable at hyperscale. Reliability and economics analysis included in this document shows that, with the large cost differential between unplanned and proactive maintenance events typical of cloud infrastructure, even modest prediction accuracy can result in

substantial net value provided false alarm rates are managed carefully. Roadmap challenges for future research include the ongoing issues of class imbalance and concept drift, interpretability, and computational overhead, and the directions suggested, from federated learning to foundation-model-based forecasting, digital-twin simulation, and closed-loop self-healing automation, point the way to more and more autonomous and reliable cloud data center operations.

ACKNOWLEDGMENTS

Authors gratefully acknowledge Universiti Teknikal Malaysia Melaka (UTeM) and the Ministry of

Higher Education Malaysia (MOHE) for their support. Authors also express their sincere appreciation to the

Faculty of Electrical Technology and Engineering (FTKE) and the Socio-Technical Competencies for Community

Empowerment (SoTeCH) Research Group, both under UTeM, for their institutional and research support.

REFERENCES

[1] Abro, A. A., et al. (2022). Artificial intelligence enabled effective fault prediction techniques in cloud computing environment for improving resource optimization. Scientific Programming, 2022, 7432949.

[2] Biggio, L., Wieland, A., Chao, M. A., Kastanis, I., & Fink, O. (2021). A dynamic predictive maintenance approach using probabilistic deep learning for a fleet of multi-component systems. IEEE Transactions on Reliability, 70(3), 1052–1063.

[3] Zeng, J., & Liang, Z. (2023). A dynamic predictive maintenance approach using probabilistic deep learning for a fleet of multi-component systems. *Reliability Engineering & System Safety*, *238*, 109456.

[4] Chang, C., Peng, W. C., & Chen, T. F. (2024). LLM4TS: Two-stage fine-tuning for time-series forecasting with pre-trained LLMs. arXiv preprint arXiv:2308.08469.

[5] Chen, C., Shi, J., Shen, M., Feng, L., & Tao, G. (2023). A predictive maintenance strategy using deep learning quantile regression and kernel density estimation for failure prediction. IEEE Transactions on Instrumentation and Measurement, 72, 1–12.

[6] Chen, Y., Yang, X., Lin, Q., Zhang, H., Gao, F., Xu, Z., et al. (2019). Outage prediction and diagnosis for cloud service systems. In The World Wide Web Conference (pp. 2659–2665). ACM.

[7] Cheng, Q., Sahoo, D., Saha, A., Yang, W., Liu, C., Woo, G., Singh, M., Saverese, S., & Hoi, S. C. (2023). AI for IT operations (AIOps) on cloud platforms: Reviews, opportunities and challenges. arXiv preprint arXiv:2304.04661.

[8] De Luca, R., Ferraro, A., Galli, A., Gallo, M., Moscato, V., & Sperlì, G. (2023). A deep attention based approach for predictive maintenance applications in IoT scenarios. Journal of Manufacturing Technology Management, 34(4), 535–556.

[9] Domingos, J. (2021). Failure prediction for cloud applications through ensemble learning. In Proceedings of the 2021 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW) (pp. 319–322). IEEE.

[10] Engineering Applications of Artificial Intelligence. (2024). Cost aware LSTM model for predicting hard disk drive failures based on extremely imbalanced S.M.A.R.T. sensors data. Engineering Applications of Artificial Intelligence, 127, 107339.

[11] Gao, J., Wang, H., & Shen, H. (2020). Task failure prediction in cloud data centers using deep learning. IEEE Transactions on Services Computing, 15(3), 1411–1422.

[12] Gill, S. S., & Buyya, R. (2018). Failure management for reliable cloud computing: A taxonomy, model, and future directions. Computing in Science & Engineering, 22(3), 52–63.

[13] Islam, M. S., & Miranskyy, A. (2020). Anomaly detection in cloud components. In 2020 IEEE 13th International Conference on Cloud Computing (CLOUD) (pp. 1–3). IEEE.

[14] Kamariotis, A., Tatsis, K., Chatzi, E., Goebel, K., & Straub, D. (2024). A metric for assessing and optimizing data-driven prognostic algorithms for predictive maintenance. Reliability Engineering & System Safety, 242, 109723.

[15] Li, Y., Jiang, Z. M., Li, H., Hassan, A. E., He, C., Huang, R., Zeng, Z., Wang, M., & Chen, P. (2020). Predicting node failures in an ultra-large-scale cloud computing platform: An AIOps solution. ACM Transactions on Software Engineering and Methodology, 29(2), 1–24.

[16] Lorenti, L., Dalle Pezze, D., Andreoli, J., Masiero, C., Gentner, N., Yang, Y., & Susto, G. A. (2023). Predictive maintenance in the industry: A comparative study on deep learning-based remaining useful life estimation. In 2023 IEEE International Conference on Industrial Informatics (INDIN) (pp. 1–8). IEEE.

[17] Mohammed, B., Awan, I., Ugail, H., & Younas, M. (2019). Failure prediction using machine learning in a virtualised HPC system and application. Cluster Computing, 22(2), 471–485.

[18] Mukwevho, M. A., & Celik, T. (2018). Toward a smart cloud: A review of fault-tolerance methods in cloud systems. IEEE Transactions on Services Computing, 14(2), 589–605.

[19] Ogunfowora, O., & Najjaran, H. (2023). Reinforcement and deep reinforcement learning-based solutions for machine maintenance planning, scheduling policies, and optimization. Journal of Manufacturing Systems, 70, 244–263.

[20] Rawat, A., Sushil, R., Agarwal, A., & Sikander, A. (2021). A new approach for VM failure prediction using stochastic model in cloud. IETE Journal of Research, 67(2), 165–172.

[21] Saxena, D., & Singh, A. K. (2022). OFP-TM: An online VM failure prediction and tolerance model towards high availability of cloud computing environments. The Journal of Supercomputing, 78(6), 8003–8024.

[22] Saxena, D., Gupta, I., Kumar, J., Singh, A. K., & Wen, X. (2021). A secure and multiobjective virtual machine placement framework for cloud data center. IEEE Systems Journal, 16(2), 3163–3174.

[23] Serradilla, O., Zugasti, E., Rodriguez, J., & Zurutuza, U. (2022). Deep learning models for predictive maintenance: A survey, comparison, challenges and prospects. Applied Intelligence, 52, 10934–10964.

[24] Shahid, M. A., Islam, N., Alam, M. M., Mazliham, M. S., & Musa, S. (2021). Towards resilient method: An exhaustive survey of fault tolerance methods in the cloud computing environment. Computer Science Review, 40, 100398.

[25] Shen, J., Wan, J., Lim, S.-J., & Yu, L. (2021). Hard disk drive failure prediction for mobile edge computing based on an LSTM recurrent neural network. Mobile Information Systems, 2021, 8878364.

[26] Shi, J., Wang, X., Zhang, Y., & Zhang, A. (2022). Convolution-LSTM-based mechanical hard disk failure prediction by sensoring S.M.A.R.T. indicators. Journal of Sensors, 2022, 7832117.

[27] Siraskar, R., Kumar, S., Patil, S., Bongale, A., & Kotecha, K. (2023). Reinforcement learning for predictive maintenance: A systematic technical review. Artificial Intelligence Review, 56, 12885–12947.

[28] Su, J., Jiang, C. F., Jin, X., et al. (2024). Large language models for forecasting and anomaly detection: A systematic literature review. arXiv preprint arXiv:2402.10350.

[29] Tengku Asmawi, T. N., Ismail, A., & Shen, J. (2022). Cloud failure prediction based on traditional machine learning and deep learning. Journal of Cloud Computing: Advances, Systems and Applications, 11, 47.

[30] Tuli, S., Casale, G., & Jennings, N. R. (2022). TranAD: Deep transformer networks for anomaly detection in multivariate time series data. arXiv preprint arXiv:2201.07284.

[31] Wang, S., Li, X., & Zhang, Y. (2024). A survey of deep learning-driven architecture for predictive maintenance. Engineering Applications of Artificial Intelligence, 133, 108140.

[32] Xu, Q., Wang, Z., Jiang, C., & Jing, Z. (2024). Data-driven predictive maintenance framework considering the multi-source information fusion and uncertainty in remaining useful life prediction. Knowledge-Based Systems, 289, 112408.

[33] Xue, J., Yan, F., Birke, R., Chen, L. Y., Scherer, T., & Smirni, E. (2015). PRACTISE: Robust prediction of data center time series. In 2015 11th International Conference on Network and Service Management (CNSM) (pp. 126–134). IEEE.

[34] Yu, F., Xu, H., Jian, S., Huang, C., Wang, Y., & Wu, Z. (2021). DRAM failure prediction in large-scale data centers. In 2021 IEEE International Conference on Joint Cloud Computing (JCC) (pp. 1–8). IEEE.

[35] Zhuang, L., Xu, A., & Wang, X.-L. (2023). A prognostic driven predictive maintenance framework based on Bayesian deep learning. Reliability Engineering & System Safety, 234, 109181.

[36] van Dinter, R., Tekinerdogan, B., & Catal, C. (2022). Predictive maintenance using digital twins: A systematic literature review. Information and Software Technology, 151, 107008.