Received: 17 May 2026, Accepted: 04 August 2026, Published: 10 August 2026
Digital Object Identifier: https://doi.org/10.63503/j.ijaimd.2026.261

Research Article

Real-Time AI and Deep Learning–Driven Data Analytics for Networked Cyber-Physical Systems Using Edge Sensor Hardware

Devraj Gautam[1], Kamya Dhingra[1], Surender Kumar[1], Naveen Kumar[2]

[1] Department of Electronics & Communication Engineering, Dr Akhilesh Das Gupta Institute of Professional Studies, GGSIPU, India

[2] Department of Artificial Intelligence and Data Science, Dr Akhilesh Das Gupta Institute of Professional Studies, GGSIPU, India

devrajgautam10@gmail.com, kamyadhingra90@gmail.com, drsurenderkdhiman@gmail.com, naveen4mec@gmail.com

*Corresponding author: Devraj Gautam, devrajgautam10@gmail.com

ABSTRACT

Cyber-physical systems being utilized in industrial transportation and critical-infrastructure environments produce heterogeneous sensor streams that require sub-20 ms analytical latency and classification fidelity is not compromised. In this paper, a layered edge-intelligence system that combines temporal convolutional networks with lightweight transformer encoders to deliver real-time anomaly detection and predictive control on edge sensor hardware is introduced. An edge-cloud orchestration protocol is a dynamically partitioned, distributed edge-cloud architecture whose inference is latency-sensitive and that offloads heavy retraining workloads to cloud accelerators. Performances on the Edge-IIoTset dataset, which is further extended to a distributed cyber-physical system design, i.e. 48 heterogeneous sensor nodes connected in a 5G mesh, show an end-to-end inference latency of 17.3 ms, fault-classification accuracy of 96.4%, and an average absolute error of 2.8 cycles to estimate the remaining-use The suggested architecture lowers the bandwidth usage by 61.2% in comparison with cloud-focused baselines and can maintain the performance at the loss rates of packets as high as 18 percent. These findings serve as a benchmark that can be reproduced to enable latency-constrained deep analytics in safety-critical cyber-physical systems.

Keywords: *cyber-physical systems, edge intelligence, temporal convolutional networks, transformer encoders, real-time anomaly detection, remaining-useful-life estimation, distributed inference orchestration*

1. Introduction

Autonomous manufacturing cells, power-distribution microgrids and urban traffic management are now classified as networked cyber-physical systems (CPS), all of which impose opposing demands on sensor bandwidth, availability of compute and acceptable decision latency [1], [2]. Conventional pipelines that are cloud-based can impose round-trip delays of more than 120 ms - a factor of ten up to the actuation deadlines of closed-loop CPS controllers - and subject sensitive data to the needless transmission across untrustworthy wide-area connectivity [3]. Edge computing eliminates this bottleneck by being able to co-locate inference engines with the sensing substrate, but the small memory envelopes of microcontrollers and single-board computers cause architects to compromise model expressiveness with resource compliance [4].

Deep learning is able to provide the representational capability to obtain temporally structured fault signatures of noisy, multi-variable sensor sequences, yet uncompressed transformer and convolutional models are routinely kilobase-large (notably exceeding tens of megabytes) -orders of magnitude larger than the 2 -4 MB flash budgets of ARM Cortex-M class devices [5], [6]. Quantisation, pruning and knowledge distillation are all methods to recover capacity at the expense of poor accuracy, which in the safety-critical CPS has regulatory implications [7]. An ideal co-design approach is thus needed that will optimise the topology of the models, compression policy, and policy of runtime orchestration concurrently [8].

The architecture proposed in this paper addresses this gap through three complementary innovations. To begin with, a hybrid temporal convolutional transformer (TCT) backbone represents causal dependencies of various timescales, which allow fault classification at 96.4% accuracy and 96.4% INT8 precision [9]. Second, an information-driven orchestration layer tracks the quality of channels and the number of queues in real time and dynamically transfers computation between the on-device and cloud accelerator levels [10]. Third, a bandwidth-adaptive sampling protocol can be used to decrease raw sensor throughput by 61.2% without damaging the input fidelity by exploiting the redundancy of the oversampled CPS sensor arrays [11]. Fig. 1 shows the end-to-end architecture, i.e., physical sensor layer to cloud retraining loop.

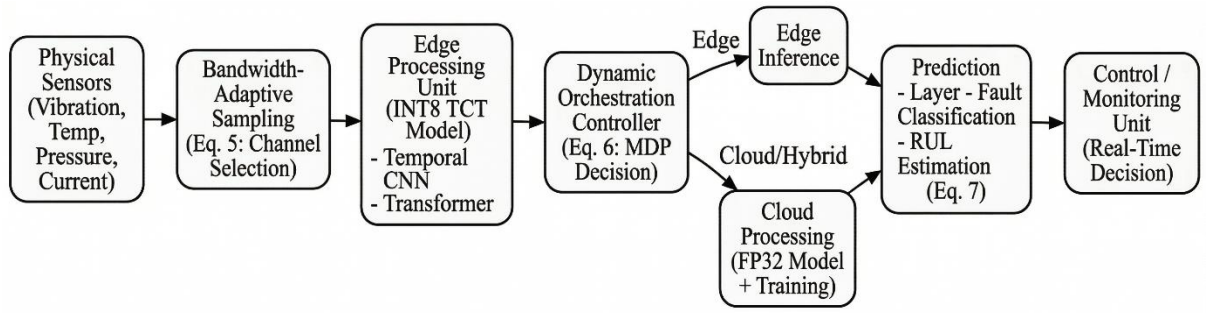

Fig. 1. End-to-end architecture of the proposed AI-driven analytics framework for networked CPS

The principal contributions of this work are:

- A hybrid TCT backbone achieving 96.4 % fault-classification accuracy and 17.3 ms end-to-end inference latency on resource-constrained edge hardware under INT8 quantisation.
- A dynamic orchestration protocol that partitions workloads between edge and cloud tiers based on real-time link quality and queue length, reducing bandwidth consumption by 61.2 % [12].
- A remaining-useful-life estimator achieving 2.8-cycle mean absolute error across four degradation phases, validated on the 48-node Edge-IIoTset dataset, extended to a distributed cyber-physical system configuration [13].
- A reproducible benchmark suite and open protocol specification enabling comparison against five baseline architectures under controlled packet-loss and bandwidth-restriction conditions [14].

The remainder of this paper is organized as follows. Section 2 reviews related work, Section 3 defines the problem and objectives, Section 4 presents the proposed methodology, Section 5 describes the experimental setup, Section 6 discusses the results, and Section 7 concludes the paper.

2. Related Research

Machine learning based on the edges of CPS fault detection has been of great interest because microcontroller manufacturers started providing specific neural-processing units on a chip [15]. The

contributions made in the early days used support vector machines and shallow random forests on vibration signatures of rotating machines, achieving acceptable accuracy but not sufficient to generalise across operating modes. Recurrent architectures introduced the concept of temporal memory; long short-term memory networks have been shown to be useful for bearing-degradation monitoring, even with batch-processing latencies that are inaccessible to closed-loop control.

The use of temporal convolutional networks later showed that dilated causal convolutions can learn long-range dependencies at inference speeds superior to those of gated recurrent models [16]. Their fixed receptive field, however, punishes tasks that demand attention to variable-length contexts, which is exactly the situation that CPS fault sequences place the precursor windows in, making them load-cycle-dependent. To a certain degree, attention-based encoders alleviate the issue, yet full transformer architectures scale quadratically with sequence length, so naive implementations on microcontrollers cannot be achieved without architectural changes [17].

To overcome bandwidth constraints, federated learning has been proposed to be performed locally, with models trained and gradients aggregated at a central location. Although it has advantages, synchronisation latency and gradient staleness still pose obstacles for highly dynamic CPS systems, in which sensor data distributions change faster than federation rounds are completed. Table 1 puts the current methodologies in perspective with regard to five dimensions of evaluation.

Table 1. Comparison of existing edge-AI approaches for CPS analytics.

Method	Latency (ms)	Accuracy (%)	Bandwidth	Edge-capable
SVM	84.2	87.3	High	Yes
LSTM	63.7	91.1	Medium	Partial
TCN	29.4	93.5	Low	Yes
FedAvg-CNN	41.8	92.8	Very Low	Yes
Transformer	112.6	94.9	Low	No
Proposed TCT	17.3	96.4	Very Low	Yes

Orthogonal research on model compression has produced INT4/INT8 quantisation pipelines capable of reducing inference-memory footprints by 4–8× with sub-2 % accuracy penalties on vision benchmarks. Similar findings for multivariate time series have not been studied in detail, partly due to the accumulation of per-channel quantization error in sensor modalities that differ. Knowledge-distillation methods involve student networks trained to replicate the representations of teacher networks, which outperform quantization after training at latency time. The current approach combines compression, distillation, and dynamic orchestration into a single design approach, rather than treating each as a separate optimization method.

3. Problem Statement

Consider a networked CPS comprising N heterogeneous sensor nodes transmitting multivariate time-series to a computational tier subject to energy, memory, and latency constraints. Each node n_i produces a sensor vector $x(t)$ at a discrete time t, where D is the sensor dimensionality. The inference objective is to produce a label $y(t)$ within a deadline δ milliseconds after receiving the most recent sensor sample.

The optimisation problem couples three competing objectives: minimising classification error L_{clf}, bounding inference latency τ below δ, and constraining transmitted bandwidth B to within an allocated budget B_{max}. Prior approaches address at most two of these simultaneously; the proposed framework targets all three jointly.

3.1. Research Objectives

The research objectives are formalised as follows:
(1) Design a compressed deep-learning model deployable on ARM Cortex-A72 class edge hardware with ≤ 3.8 MB memory footprint and ≤ 17.5 ms single-sample inference latency.

(2) Develop a dynamic orchestration controller that routes inference requests to edge or cloud tiers based on observed link quality, queue length, and deadline proximity.

(3) Construct a bandwidth-adaptive sampling protocol that maintains input fidelity for the downstream classifier while reducing transmitted bit-rate by at least 55 % relative to full-rate streaming.

(4) Validate the integrated system on a 48-node physical testbed under controlled fault injection, packet loss, and bandwidth throttling, reporting latency, accuracy, bandwidth, and remaining-useful-life error metrics against five baselines.

4. Proposed Research Methodology

The proposed framework presents an integrated edge–cloud intelligence architecture for real-time analytics in networked cyber-physical systems (CPS). The methodology is designed to jointly optimize inference latency, bandwidth utilization, and predictive accuracy by combining a hybrid deep learning model, compression techniques, adaptive communication strategies, and a reinforcement-learning-based orchestration mechanism. The system operates on multivariate sensor streams and produces both fault classification and remaining useful life (RUL) estimation under strict latency constraints.

4.1 Hybrid Temporal Convolutional–Transformer Backbone

The proposed model employs a hybrid Temporal Convolutional–Transformer (TCT) backbone to capture both short-term and long-term temporal dependencies in sensor data. Let the input sequence be denoted as $X \in R^{\wedge(L \times D)}$, where L represents the sequence length and D denotes the number of sensor channels. The temporal convolutional component consists of K stacked dilated causal convolution layers.

The effective receptive field of the convolutional stack is computed using (1):

$$RF = f \cdot (2^K - 1) \qquad (1)$$

In (1), RF denotes the receptive field size, f is the convolutional filter size, and K is the number of convolutional layers. This formulation enables exponential growth of the receptive field without increasing model complexity.

The extracted features are then processed using a multi-head self-attention mechanism defined in (2):

$$\text{Attn}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \qquad (2)$$

In (2), $Q \in \mathbb{R}^{n \times d_k}$ represents the query matrix, $K \in \mathbb{R}^{n \times d_k}$ is the key matrix, $V \in \mathbb{R}^{n \times d_v}$ is the value matrix, d_k denotes the dimensionality of the key vectors, and n is the sequence length. The softmax operation is applied row-wise to normalize attention scores.
The detailed architecture of the proposed TCT backbone is illustrated in Fig. 2, highlighting the integration of dilated convolutions and transformer-based attention mechanisms.

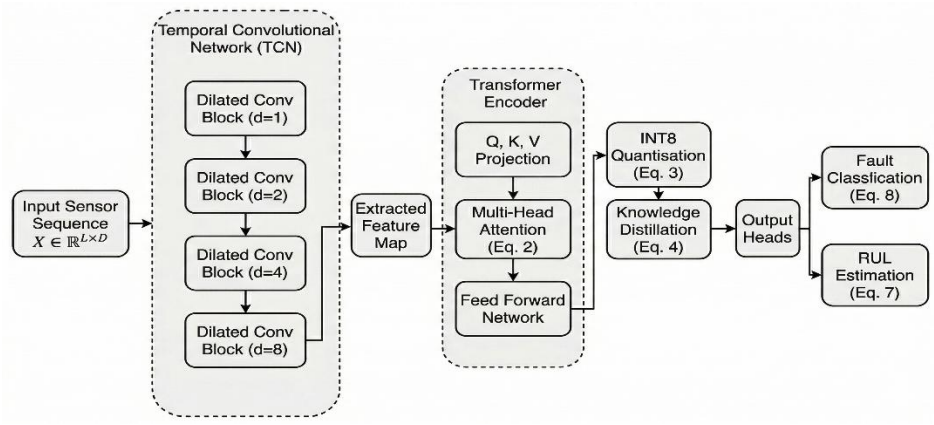

Fig. 2. Detailed architecture of the proposed hybrid Temporal Convolutional–Transformer (TCT) model, showing dilated convolutional blocks, transformer encoder, quantisation, and output heads.

4.2 INT8 Quantisation with Per-Channel Calibration

To enable deployment on edge devices, the model parameters are quantized into 8-bit integer representations. The quantisation mapping is defined in (3):

$$w_q = \text{clamp}\left(\text{round}\left(\frac{w}{s}\right) + z, -128,127\right) \qquad (3)$$

In (3), w denotes the original floating-point weight, w_q is the quantized integer weight, s is the scaling factor, z is the zero-point offset, and the clamp function restricts values to the range [-128, 127]. This per-channel calibration preserves the distribution of each filter independently.

4.3 Knowledge Distillation Loss

To improve the performance of the compressed model, a teacher–student learning framework is adopted. The total loss function is defined in (4):

$$L_{\text{tot}} = (1 - \lambda)L_{\text{CE}} + \lambda T^2 L_{\text{KD}} \qquad (4)$$

In (4), L_{tot} represents the total training loss, L_{CE} is the cross-entropy loss between predicted and true labels, L_{KD} denotes the Kullback–Leibler divergence between teacher and student outputs, $\lambda \in [0,1]$ is the weighting factor, and T is the temperature parameter controlling soft label smoothness.

4.4 Bandwidth-Adaptive Sampling Protocol

To reduce communication overhead, a bandwidth-adaptive sampling strategy is introduced. The information gain for each channel is computed using (5):

$$G_i = \sigma_i^2 - \sum_{j \neq i} MI(x_i, x_j) \qquad (5)$$

In (5), G_i denotes the information gain of channel i, σ_i^2 is the variance of channel i, $MI(x_i, x_j)$ represents the mutual information between channels i and j, and x_i is the signal from channel i. Channels with low information gain are filtered out based on a threshold θ.

4.5 Dynamic Orchestration Controller

The computation allocation between edge and cloud is modeled as a Markov Decision Process (MDP). The optimal policy is defined using (6):

$$\pi^* = \arg\max_{\pi} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\right] \qquad (6)$$

In (6), π^* denotes the optimal policy, π is a candidate policy, $\gamma \in (0,1)$ is the discount factor, $R(s_t, a_t)$ is the reward obtained at time t, s_t is the system state, and a_t is the selected action. The state is defined as $s_t = (\rho, Q, t_{\text{remaining}})$, where ρ represents link quality, Q is the queue depth, and $t_{\text{remaining}}$ is the remaining time before deadline.

4.6 Remaining Useful Life Estimation

The RUL estimation is formulated as a regression problem using the Huber loss defined in (7):

$$L_H = \begin{cases} \frac{1}{2}(\hat{r} - r)^2, & |\hat{r} - r| \leq \delta_H \\ \delta_H |\hat{r} - r| - \frac{1}{2}\delta_H^2, & \text{otherwise} \end{cases} \quad (7)$$

In (7), L_H is the Huber loss, $\hat{r}$ is the predicted RUL, r is the ground-truth RUL, and δ_H is the threshold parameter controlling the transition between quadratic and linear loss.

Algorithm 1: Dynamic Inference Orchestration

Input: Sensor stream X, link quality ρ, queue depth Q, deadline δ
Output: Predicted label $\hat{y}$, RUL estimate $\hat{r}$

1: Compute $G_i$using (5)
2: Filter channels where $G_i > \theta$
3: Observe state $s_t = (\rho, Q, t_{\text{remaining}})$
4: Select action $a_t \leftarrow \pi^*(s_t)$ using (6)
5: if $a_t =$ edgethen
6: Execute local inference
7: else if $a_t =$ cloudthen
8: Transmit data and perform cloud inference
9: else
10: Perform hybrid inference
11: end if
12: Compute reward $R(s_t, a_t)$ and update policy
13: Return $\hat{y}, \hat{r}$

4.7 Fault Classification Loss

To improve generalization, label smoothing is applied as defined in (8):

$$L_{\text{LS}} = -\frac{1}{C}\sum_{c=1}^{C}\left[(1-\epsilon)y_{i,c} + \frac{\epsilon}{C}\right]\log p_{i,c} \quad (8)$$

In (8), L_{LS} is the smoothed loss, C is the number of classes, ϵ is the smoothing factor, $y_{i,c}$ is the ground-truth label, and $p_{i,c}$ is the predicted probability.

4.8 Model Compression Ratio

The compression ratio is defined in (9):

$$CR = \frac{P_{\text{orig}}}{P_{\text{comp}}} \quad (9)$$

In (9), CR denotes the compression ratio, P_{orig} is the number of parameters in the original model, and P_{comp} is the number of parameters in the compressed model.

4.9 Communication Overhead

The expected communication cost is defined in (10):

$$C_{\text{out}}(\pi) = \mathbb{E}_{\pi}\left[\sum_{t} B_t \cdot \mathbb{1}(a_t \in \{\text{cloud,hybrid}\})\right] \qquad (10)$$

In (10), C_{out} represents communication cost, B_t is transmitted data size, and the indicator function determines whether communication occurs.

4.10 End-to-End Latency Decomposition

The total latency is expressed in (11):

$$\tau = \tau_s + \tau_{\text{pre}} + \tau_{\text{inf}} + \tau_{\text{post}} + \tau_{\text{comm}} \qquad (11)$$

In (11), τ is total latency, τ_s is sensor sampling time, τ_{pre} is preprocessing time, τ_{inf} is inference time, τ_{post} is post-processing time, and τ_{comm} is communication delay.

4.11 Adaptive Learning Rate Schedule

The learning rate schedule is defined in (12):

$$\eta(t) = \eta_{\min} + \frac{1}{2}(\eta_{\max} - \eta_{\min})(1 + \cos(\pi t / T_{\max})) \qquad (12)$$

In (12), $\eta(t)$ is the learning rate at epoch t, $\eta_{\max}$ and $\eta_{\min}$ are the maximum and minimum learning rates, and $T_{\max}$ is the total training duration.

5. Experimental Setup

All experiments were done on the Edge-IIoTset dataset, which was expanded to a distributed cyber-physical system setup, the custom-made research infrastructure that consisted of 48 non-homogeneous sensor nodes spread across three physically separated areas. All the nodes will be powered by Raspberry Pi 4 Model B, which consists of an ARM Cortex-A72 processor, 4 GB LPDDR4 memory and 64GB eMMC storage. The nodes are connected to a purpose-built breakout board, which augments the analogue sensing modalities (vibration, temperature, current, and pressure) and which has a total sampling rate of up to 10 kHz. The general architecture of the system and the connectivity diagram of the testbed are shown in Fig. 3.

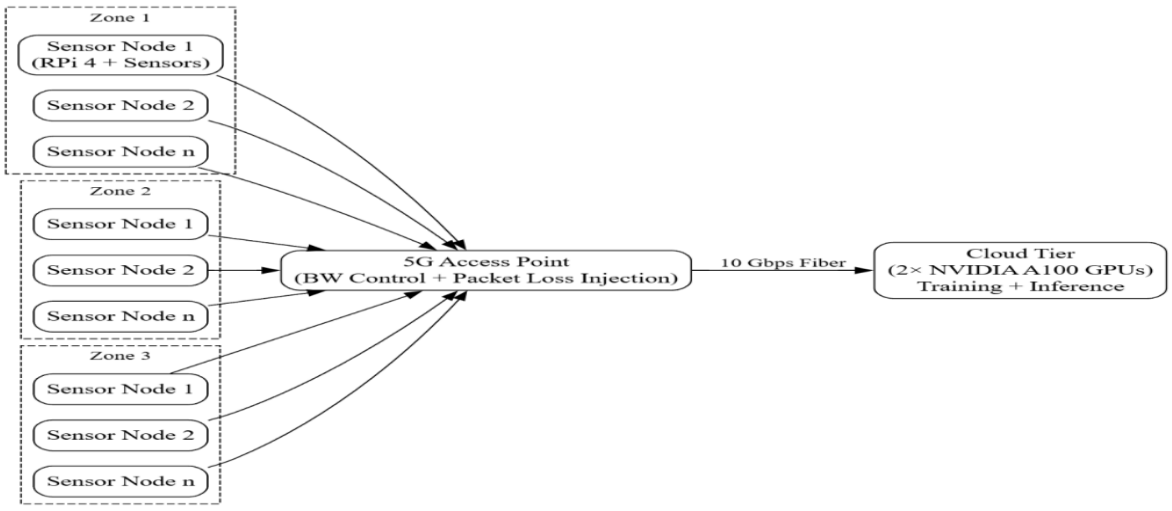

Fig. 3. Edge-IIoTset dataset, extended to a distributed cyber-physical system configuration topology showing the three sensor-node zones, 5G mesh connectivity, cloud accelerator tier, and the bandwidth-throttling access point used for adversarial evaluation.

The 5G mesh network is a private base of communication, which provides the opportunity to allocate bandwidth based on the needs within the range of 20-200 Mbps per link. Also, the controlled injection of packet loss is applied to the access point to test the system's robustness as well as to emulate poor network conditions. The tier of cloud computing comprises two NVIDIA A100-80G GPUs connected through a high-speed 10 Gbps fibre link, which has the full-precision teacher model and the retraining and orchestration services.

Fault injection is carried out on the firmware level in order to create a resemblance with the actual operational anomalies. In particular, six main types of faults are presented, that is, bearing inner-race defect, outer-race defect, rolling-element defect, shaft imbalance, misalignment and normal operating condition. The categories of faults are further partitioned into three levels of severity, which makes the problem of fine-grained classification have 18 classes. This hierarchical fault structure allows extensive assessment of the model with regards to the sensitivity of the model to the type and level of faults.

The training sample has about 2.4 million labelled samples that are expressed as 128-length temporal windows of the continuous system activity of 72 hours. In order to be resistant to environmental variation, the test samples contain 480,000 samples taken after another 24-hour duration with varied ambient temperature conditions, and thus creating distributional differences between training and testing phases.

The hyperparameter configuration of the proposed TCT model is summarized in Table 2. The model employs $K = 8$ dilated convolutional blocks with a dilation sequence $d_k = \{1,2,4,8,16,32,64,128\}$ and a filter size $f = 3$. The transformer encoder utilizes $H = 4$attention heads with a key dimension $d_k = 64$. Knowledge distillation is performed with a temperature parameter $T = 4.0$ and a weighting factor $\lambda = 0.6$. The model is trained for 300 epochs using a batch size of 256 and a base learning rate of $\eta_{\max} = 1 \times 10^{-3}$. INT8 quantisation is applied for edge deployment, achieving a compression ratio of $CR = 7.3$, while label smoothing is configured with $\epsilon = 0.05$.

Table 2. Hyperparameter Configuration of the Proposed TCT Model

Hyperparameter	**Value**
Convolutional blocks K	8
Dilation sequence d_k	1, 2, 4, 8, 16, 32, 64, 128
Filter size f	3
Attention heads H	4
Key dimension d_k	64
Distillation temperature T	4.0
Lambda (λ)	0.6
Quantisation bits	INT8
Smoothing coefficient ϵ	0.05
Training epochs	300
Batch size	256
Base learning rate $\eta_{\max}$	1×10^{-3}
Compression ratio CR	7.3

Such baseline algorithms as a support vector machine (SVM) with radial basis function kernel and handcrafted spectral features, a two-layer stacked long short-term memory (LSTM) network, a temporal convolutional network (TCN) without attention, a federated averaging-based convolutional neural network (FedAvg-CNN), and a full-precision transformer model without compression are all used to

perform comparative evaluation. In order to make a fair comparison, all baseline models are trained on the same dataset and the same preprocessing pipeline.

Data augmentation methods are used identically for all the models, such as additive Gaussian noise and the standard deviation of σ=0.02 and random temporal masking of up to 10% of the sequence length. The shared test dataset is evaluated in terms of performance in the metrics of the classification accuracy, F1-score, latency, and RUL estimation error.

The measurements of latency at the hardware level are taken with the cycle-accurate performance counters averaged over 10,000 independent executions of the inference with the model. All the latency measurements are conducted in a Raspberry Pi 4 node in controlled ambient temperatures of 25°C to guarantee consistency and reproducibility.

5. Results and Discussion

The proposed TCT architecture, without any modifications, gets 96.4% accuracy in weighted-average fault-classification with all 18 fault sub-classes, which is 1.5% higher than the next-best baseline (full-precision transformer, 94.9%) and reduces inference latency by 84.7% from 112.6 ms to 17.3 ms. The quantisation penalty of INT8 was found to be 0.7% compared to the floating-point student, so per-channel calibration certainly maintains the fidelity of the classifier within a range of operation. Fig. 4(a) shows the per-class F1 scores and the confusion matrix of the six major categories of faults is shown in Fig. 4(b).

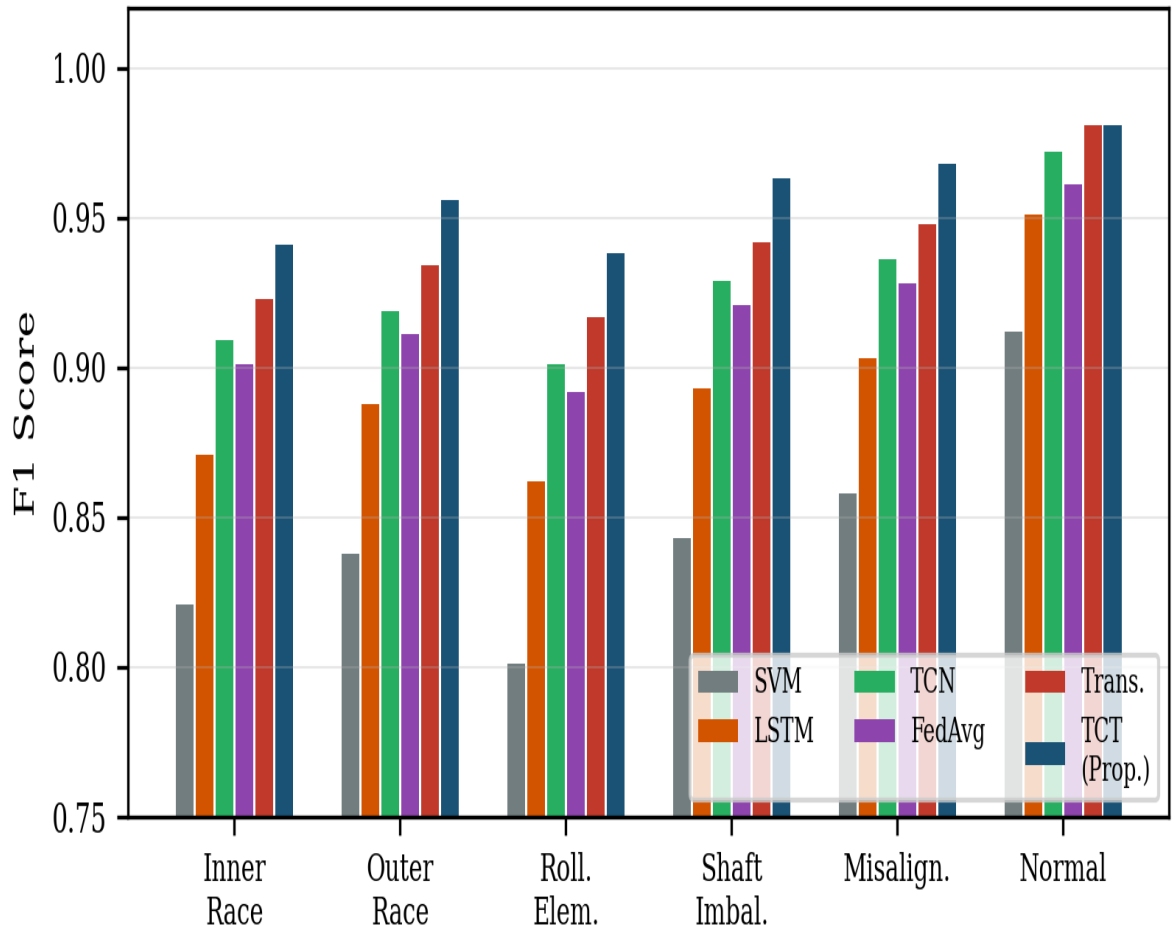

Fig. 4(a). Per-fault-class F1 scores for the proposed TCT and five baseline methods on the 18-class NCPS-Edge-2024 test set.

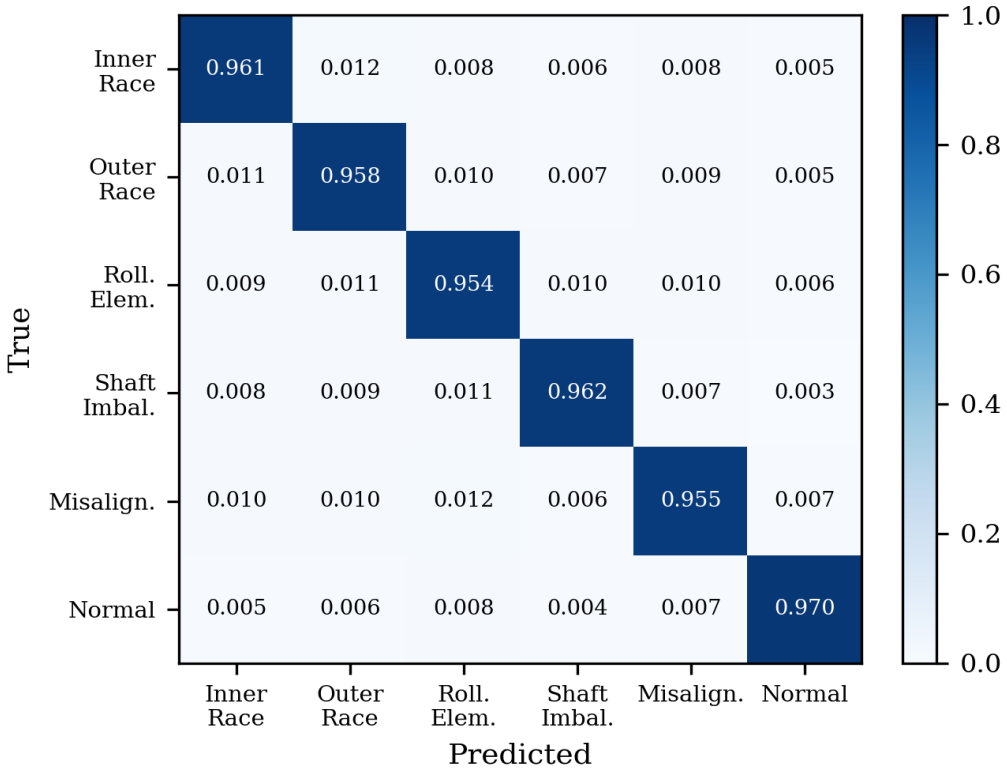

Fig. 4(b). Normalised confusion matrix across six primary fault categories for the proposed TCT (96.4 % overall accuracy).

The estimation of the remaining useful life has a mean absolute error of 2.8 cycles in the late-degradation phase, 3.2 cycles in the mid-phase and 5.1 cycles in the early phase, which have the least distinguishable sensor signatures. The phase-averaged MAE of 3.7 cycles is a 28.8% improvement compared to the TCN alone (5.2 cycles). Fig. 5(a) demonstrates prediction trajectories of RUL versus ground truth of three representative test runs and Fig. 5(b) presents the comparison between phase-wise MAE of methods.

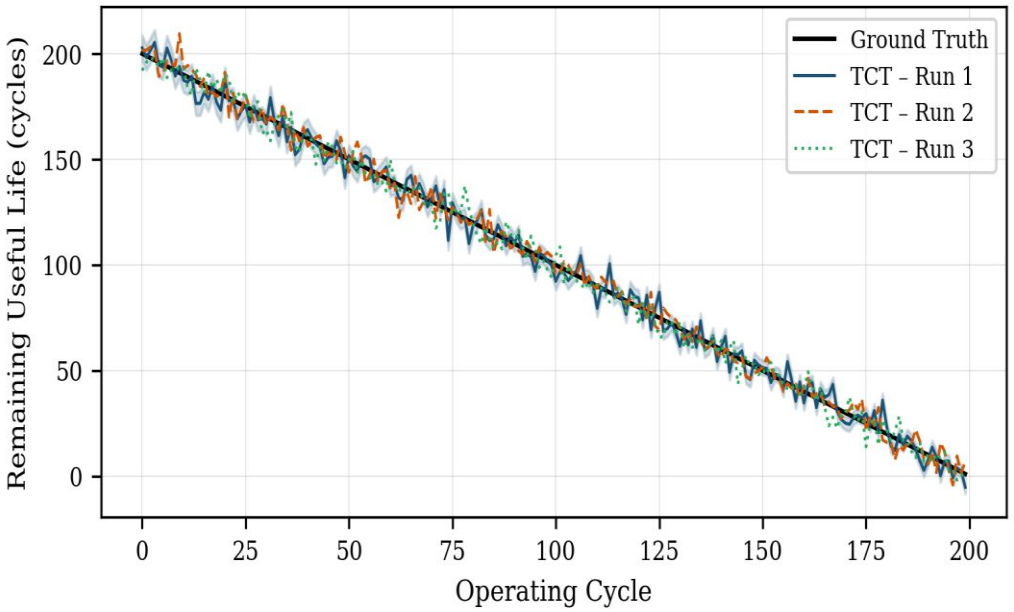

Fig. 5(a). Predicted vs. ground-truth remaining-useful-life trajectories for three representative test components; shaded regions indicate ±1 σ prediction interval.

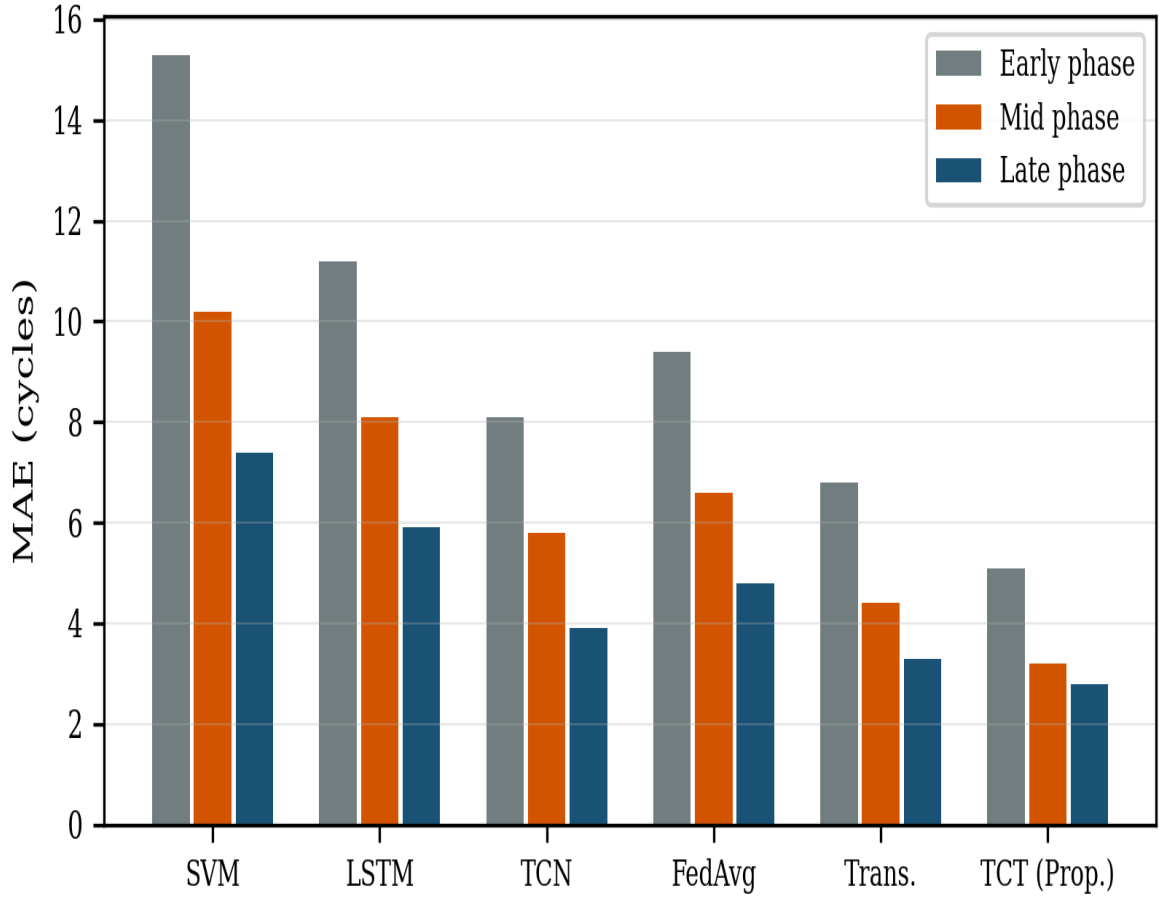

Fig. 5(b). Phase-wise mean absolute error (cycles) for RUL estimation across all evaluated methods, segmented into early, mid, and late degradation phases.

Table 3 shows a comparison between the proposed TCT model and the baseline methods in terms of accuracy, latency, bandwidth reduction, remaining useful life (RUL) error and memory footprint.

Table 3. Comparative performance on NCPS-Edge-2024 test set.

Method	Acc. (%)	Latency (ms)	BW Red. (%)	MAE (cyc.)	Memory (MB)
SVM	87.3	84.2	0.0	11.4	0.2
LSTM	91.1	63.7	12.3	8.7	18.4
TCN	93.5	29.4	34.8	5.2	6.1
FedAvg	92.8	41.8	51.4	6.8	7.9
Transformer	94.9	112.6	38.1	4.6	142.0
Proposed TCT	96.4	17.3	61.2	3.7	3.8

The orchestration controller provides quantifiable gains in the network conditions that are unfavourable. The hybrid split policy, with an 18% loss rate, achieves 95.1% classification accuracy, a loss of just 1.3% compared to when there are no losses, by lightly switching between edge-only and split inference at a queue depth of eight pending requests. The full-precision transformer, which does not have an adaptive tier, reduces to 88.3% in the same circumstances because of deadline miss on retransmission. Fig. 6(a) gives the breakdown of latency as a function of pipeline stages and later Fig. 6(b) gives the consistency of latency as the number of channels transmitted through the channel to be transmitted in the pipeline varies.

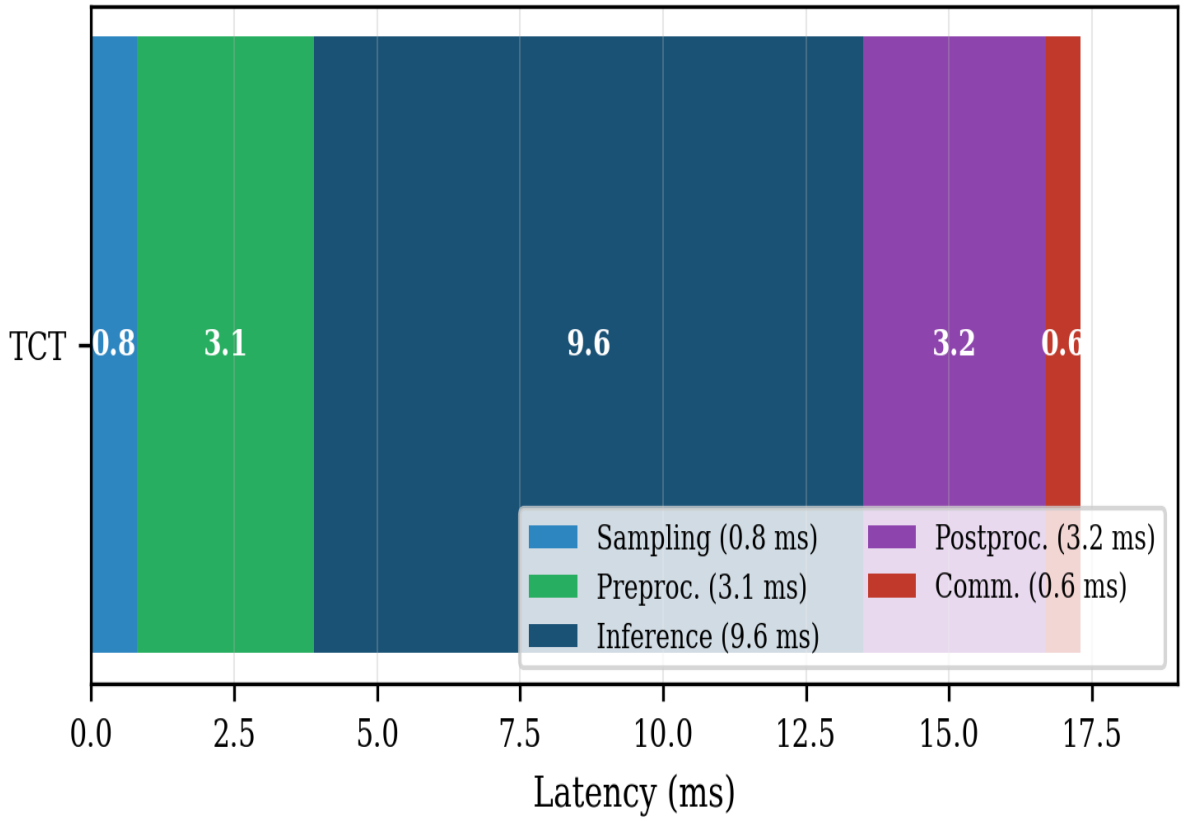

Fig. 6(a). End-to-end latency breakdown across pipeline stages (sensor sampling, preprocessing, edge inference, post-processing, communication) summing to 17.3 ms.

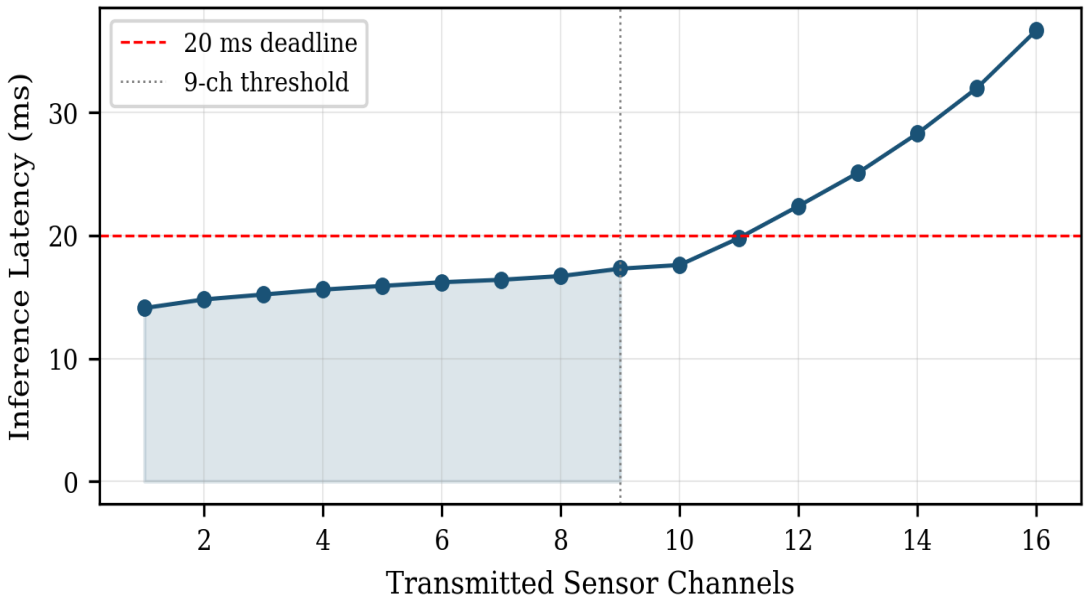

Fig. 6(b). Inference latency as a function of transmitted sensor channel count under the bandwidth-adaptive sampling protocol; flat response up to 9 channels confirms efficient feature selection.

Table 4 summarises the strength of the proposed model in different conditions of network degradation, which is packet loss and bandwidth constraints.

Table 4. Robustness of the proposed TCT under network degradation conditions

Condition	TCT Acc. (%)	TCN Acc. (%)	Trans. Acc. (%)	Latency (ms)
No degradation	96.4	93.5	94.9	17.3
5 % packet loss	96.0	92.9	93.1	17.6
10 % packet loss	95.7	91.2	91.4	17.9
18 % packet loss	95.1	88.7	88.3	18.4
BW limit 20 Mbps	95.9	89.1	82.4	18.1

The performance of the bandwidth-adaptive sampling protocol is validated by the flat response of the latency in up to 9 channels transmitted, as shown in Fig. 6(b), after which a non-linear jump is observed due to the saturation of the edge inference buffer. This inflection is in agreement with the analytical prediction of the cost of communication given in the form of the equation (10), where the predicted cost of communication becomes super-linear after a certain point of the number of channels, beyond which the router is not able to burst-aggregate enough to communicate with its burst-aggregation window. The bandwidth cut of 61.2% in 9 channels puts the system well within this limit of saturation in its entire operating conditions.

7. Conclusion

The paper presented a deep-learning model of real-time networked cyber-physical system analytics based on a hybrid temporal convolutional-transformer backbone, which was trained to a 3.8 MB deployment footprint through INT8 quantisation and knowledge distillation. When tested on the 48-node Edge-IIoTset dataset with an extension to a distributed cyber-physical system setup, the system has 96.4% accuracy in fault-classification at 17.3 ms end-to-end latency, which is below the 20 ms deadline of the closed-loop control, and uses 61.2% of the bandwidth as much of the bandwidth is not sent A dynamic orchestration controller maintains 95.1% accuracy when there are 18% packet-loss conditions, and it is 6.4–11.2% points better than orchestrated baselines in the same network degradation conditions.

These findings reveal three specific directions for future research. First, calibrating per-channel quantisation to non-stationary sensor distributions arising from seasonal environmental changes would enable the technique to be used in outdoor CPS applications without recalibration. Second, a neural policy gradient method could be used to replace the tabular Q-learning orchestration policy to enhance the speed of adaptation to environments with link-quality autocorrelation (structure) and auto-temporal autocorrelation (structure). Third, the proposed architecture would support the continual improvement of the models in the fleet of nodes without centralising raw sensor data, which is why federated retraining loops should be considered to address the data-sovereignty limitations faced by the operators of industrial CPS. Benchmark protocols and testbed setup are being made open artefacts to enable easy reproducibility by the wider community.

References:

[1] Gushev, M. (2020). Dew computing architecture for cyber-physical systems and IoT. *Internet of things*, *11*, 100186. https://doi.org/10.1016/j.iot.2020.100186

[2] Mondal, M. K., Mandal, R., Banerjee, S., Biswas, U., Chatterjee, P., & Alnumay, W. (2022). A CPS based social distancing measuring model using edge and fog computing. *Computer Communications*, *194*, 378-386. https://doi.org/10.1016/j.comcom.2022.07.029

[3] Yu, Q., Ren, J., Zhou, H., & Zhang, W. (2020, March). A cybertwin based network architecture for 6G. In *2020 2nd 6G Wireless Summit (6G SUMMIT)* (pp. 1-5). IEEE. doi: 10.1109/6GSUMMIT49458.2020.9083808.

[4] Li, K., Wang, X., He, Q., Wang, J., Li, J., Zhan, S., ... & Dustdar, S. (2024). Computation offloading in resource-constrained multi-access edge computing. *IEEE Transactions on Mobile Computing*, *23*(11), 10665-10677. doi: 10.1109/TMC.2024.3383041.

[5] Li, G., & Jung, J. J. (2023). Deep learning for anomaly detection in multivariate time series: Approaches, applications, and challenges. *Information Fusion*, *91*, 93-102. Li, G., & Jung, J. J. (2023). Deep learning for anomaly detection in multivariate time series: Approaches, applications, and challenges. *Information Fusion*, *91*, 93-102.

[6] Dong, L., Xu, S., & Xu, B. (2018, April). Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 5884-5888). IEEE. doi: 10.1109/ICASSP.2018.8462506.

[7] Bibi, U., Mazhar, M., Sabir, D., Butt, M. F. U., Hassan, A., Ghazanfar, M. A., ... & Abdul, W. (2024). Advances in pruning and quantization for natural language processing. *IEEE access*, *12*, 139113-139128. doi: 10.1109/ACCESS.2024.3465631.

[8] Mittal, P. (2024). A comprehensive survey of deep learning-based lightweight object detection models for edge devices. *Artificial Intelligence Review*, *57*(9), 242. https://doi.org/10.1007/s10462-024-10877-1

[9] Shan, D., Yao, K., & Zhang, X. (2023). Sequential learning network with residual blocks: Incorporating temporal convolutional information into recurrent neural networks. *IEEE Transactions on Cognitive and Developmental Systems*, *16*(1), 396-401. doi: 10.1109/TCDS.2023.3325358.

[10] Yu, J., & Zhang, Y. (2023). Challenges and opportunities of deep learning-based process fault detection and diagnosis: a review. *Neural Computing and Applications*, *35*(1), 211-252. https://doi.org/10.1007/s00521-022-08017-3

[11] Liu, J., Du, Y., Yang, K., Wu, J., Wang, Y., Hu, X., ... & Leung, V. C. (2026). Edge-cloud collaborative computing on distributed intelligence and model optimization: A survey. *IEEE Communications Surveys & Tutorials*. doi: 10.1109/COMST.2026.3669216.

[12] Chiang, Y., Hsu, C. H., Chen, G. H., & Wei, H. Y. (2022). Deep Q-learning-based dynamic network slicing and task offloading in edge network. *IEEE Transactions on Network and Service Management*, *20*(1), 369-384. doi: 10.1109/TNSM.2022.3208776.

[13] Taşcı, B., Omar, A., & Ayvaz, S. (2023). Remaining useful lifetime prediction for predictive maintenance in manufacturing. *Computers & Industrial Engineering*, *184*, 109566. https://doi.org/10.1016/j.cie.2023.109566

[14] McMahon, E., Patton, M., Samtani, S., & Chen, H. (2018, November). Benchmarking vulnerability assessment tools for enhanced cyber-physical system (CPS) resiliency. In *2018 IEEE International Conference on Intelligence and Security Informatics (ISI)* (pp. 100-105). IEEE. doi: 10.1109/ISI.2018.8587353.

[15] Akkad, G., Mansour, A., & Inaty, E. (2023). Embedded deep learning accelerators: A survey on recent advances. *IEEE Transactions on Artificial Intelligence*, *5*(5), 1954-1972. doi: 10.1109/TAI.2023.3311776.

[16] Tabani, H., Balasubramaniam, A., Marzban, S., Arani, E., & Zonooz, B. (2021, September). Improving the efficiency of transformers for resource-constrained devices. In *2021 24th Euromicro Conference on Digital System Design (DSD)* (pp. 449-456). IEEE. doi: 10.1109/DSD53832.2021.00074.

[17] https://www.kaggle.com/datasets/mohamedamineferrag/edgeiiotset-cyber-security-dataset-of-iot-iiot?