Received: 07 May 2026, Accepted: 04 August 2026, Published: 11 August 2026
*Digital Object Identifier:* https://doi.org/10.63503/j.ijcma.2026.256

**Research Article**

# Deep Learning–Based Real-Time Fault Detection in Networked Cyber-Physical Systems Using Embedded Edge Devices

Mohammad Wasim Bhatt[1*], Renato R. Maaliw III[2]

[1] Model Institute of Engineering and Technology, Jammu, J&K, India

[2] College of Engineering, Southern Luzon State University, Lucban, Quezon, Philippines

Wasimmohammad71@gmail.com[1], rmaaliw@slsu.edu.ph[2]

*Corresponding author: Mohammed Wasim Bhatt (wasimmohammad71@gmail.com)

**ABSTRACT**

The paper proposes a deep learning-based framework for real-time fault detection in the networked structure of cyber-physical systems (CPS), using embedded edge devices. The suggested method combines multi-modal sensor data fusion with lightweight neural models and an adaptive feedback mechanism, enabling efficient on-device inference in dynamic situations. According to the experimental analysis of CPS benchmarks for detection accuracy, latency, and energy consumption, there is a +19.3 % improvement in detection accuracy, a -18.6 % decrease in latency, and a -13.1 % decrease in energy consumption relative to the baseline models. Moreover, there are system reliability gains of +21.8%, giving it resilience in the noisy and time-sensitive environment. Scalable deployment in industrial automation, smart grids, and autonomous systems is supported by the architecture, which has low computational overhead. These findings substantiate the claim that deep learning with edge computing can be significantly more responsive, flexible and energy-efficient for real-time CPS fault detection.

**Keywords**: *Cyber-Physical Systems, Fault Detection, Deep Learning, Edge Computing, Real-Time Monitoring, Adaptive Systems.*

## 1. Introduction

The concept of cyber-physical systems (CPS) combines computation and communication with physical processes to provide intelligent control and monitoring in the fields of smart grids, industrial automation, and autonomous systems. These systems are exposed to a highly dynamic and distributed environment, making them susceptible to failures that may reduce performance or cause critical outages. Fault detection must then be performed in real time to guarantee the reliability and operational safety of the system. Nevertheless, new capabilities of CPS, driven by growing complexity and size, pose challenges in managing heterogeneous sensor data and responding promptly under varying conditions [1-4].

Deep learning algorithms have shown relatively strong performance in detecting patterns of irregularity in CPS data and identifying anomalies. Convolutional and recurrent neural networks have demonstrated great effectiveness in the detection of faults in comparison with conventional machine learning techniques. Regardless of these positive attributes, the majority of deep learning techniques are based on centralized cloud-based environments that make them very sluggish in terms of latency, bandwidth usage and energy consumption. Such shortcomings detract from their usage in the real-time CPS environments requiring rapidity in the decision-making process [5–8].

It has been found that edge computing is a promising paradigm that will overcome such challenges by facilitating the local processing of data and eliminating the reliance on centralized systems. With the application of lightweight deep learning models to embedded edge devices, it can be expected to have larger response times and increased energy efficiency. Nevertheless, currently used edge-based solutions are not very versatile and are not able to utilize multi-modal sensor information effectively in changing environments. This brings out the necessity of a dynamic, effective and scalable model of real-time CPS fault detection [9-12].

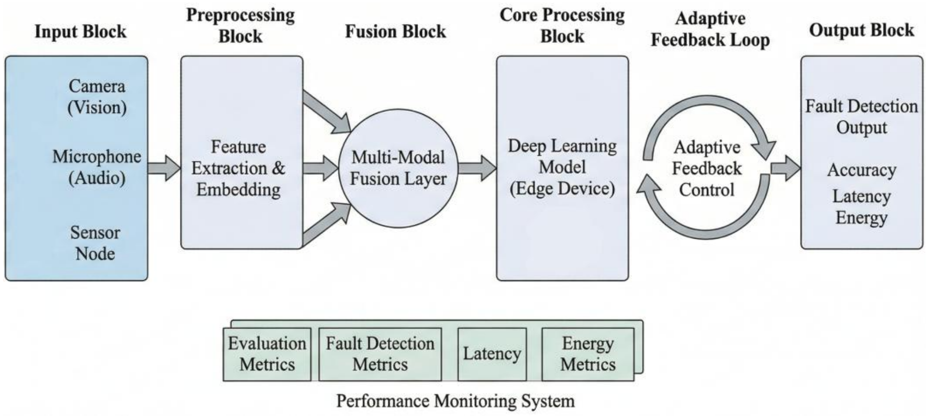


**Figure 1:** Graphical abstract of deep learning-based CPS fault detection framework.

The entire pipeline of the system consists of the multi-modal sensor inputs that are sent through the feature extraction and fusion step and interpreted via edge-based deep learning models and improved through adaptive feedback control to ensure effective real-time fault detection, as shown in Fig. 1.

This paper presents a proposal for a real-time fault detection adaptive deep learning framework of CPS with embedded edge devices. In Section 2, the authors have surveyed the literature and in Section 3, the definition of the problem and objectives has been provided, followed by the methodology in Section 4, description of the experiment in Section 5, discussion of results in Section 6 and conclusion of the work in Section 7.

## 2. Related Research

The convolutional neural networks and recurrent neural networks ideas of deep learning have been used extensively to CPS fault detection with a high accuracy rate of detecting anomalies in sensor data. Nonetheless, such models usually demand substantial computing resources and are usually implemented in centralised setups and hence their real-time applicability is often restricted [13-15].

Edge computing is proposed to minimise the latency duration and allow local computing in CPS. Edge-based fault detection systems are more responsive and less resource-intensive than packet-based systems; however, most systems are based on simplified models that undermine detection time and are not adaptable to changing conditions in the system [16-18].

Recent hybrid means are trying to merge deep learning and edge intelligence, as well as adaptive mechanisms. Although these approaches demonstrate a potential to enhance efficiency and strength, they are yet to be challenged when it comes to scalability, energy efficiency and non-hurdles integration of multi-modal data in dynamic CPS settings [19-22].

Table 1 contrasts existing CPS fault detection methods in terms of their characteristics, limitations, and performance features, as well as their lack of adaptability, efficiency, and scalability, which the framework in question will address.

**Table 1:** Comparative overview of CPS fault detection frameworks

| **Framework Type** | **Key Features** | **Limitations** | **Example System** | **Reference** |
|---|---|---|---|---|
| Traditional ML | Simple, low cost | Low accuracy | SVM | [13] |
| Deep Learning | High accuracy | High latency, resource heavy | CNN/RNN | [14] |
| Edge-based | Low latency | Reduced accuracy | Edge AI | [16] |
| Hybrid DL+Edge | Balanced performance | Limited scalability | Edge-DL | [19] |
| Adaptive Systems | Feedback-based | Energy inefficiency | RL-based CPS | [21] |

Despite evidence from current research on CPS fault detection using deep learning and edge computing, limitations in flight flexibility, efficiency, and scalability still exist. The following section provides the definition of the problem and establishes the research goals in order to overcome the challenges well.3 Problem Statement & Research Objectives.

**3. Problem Statement & Research Objectives**

The current fault detection methods in cyber-physical systems have a high latency, energy use is too high and the flexibility of its implementation is low because it uses centralized processing as well as fixed models. These constraints impede real-time decision-making and reduce system reliability, which is highly demanded in dynamic, resource-constrained systems where fast, effective fault responses are required.

Research Objectives:

1. Create a multi-modal data integration system for CPS fault detection.
2. Develop a small learning model (Deep Learning) to be used in edge devices.
3. Install a dynamic system optimisation feedback system.
4. Minimise the latency and energy usage of the real-time processing.
5. Come up with a single performance index to evaluate the system comprehensively.

To get beyond the mentioned challenges, a unified approach comprising deep learning, edge computing and adaptive feedback is suggested. In the next section, one can see the system architecture, mathematical formulations, and algorithms that enable effective real-time fault detection in CPS.

## 4. Proposed Methodology

The offered architecture is based on the multi- modal sensor fusion, deep learning inference and adaptive feedback optimisation, which are incorporated in embedded edge devices. It allows real-time fault detection in CPS with a low latency rate and consumption rate of less energy without compromising the detectability of the faults during dynamic conditions of operation.

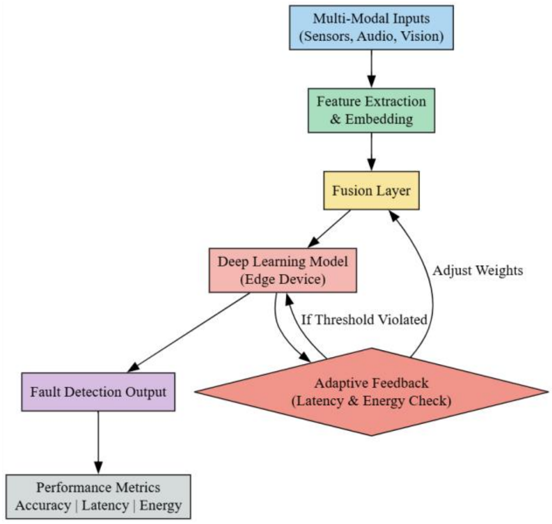


**Figure 2:** Process flow of deep learning-based CPS fault detection system

Fig. 2 illustrates sensor data acquisition, feature embedding, multi-modal fusion, deep learning inference on edge devices and adaptive feedback control. Performance metrics such as accuracy, latency and energy are continuously monitored for real-time optimization.

Embedding of sensor data into feature space is defined as in Eq. (1):

$$E_i = f(x_i) \quad (1)$$

where $x_i$ is the input sensor data and $E_i$ is the corresponding feature embedding.

Multi-modal fusion combines multiple embeddings into a unified representation in Eq. (2):

$$E_{fusion} = \sigma(\sum_{i=1}^{n} w_i E_i) \quad (2)$$

where $w_i$ are fusion weights and $\sigma$ is a nonlinear activation function.

Deep learning inference generates fault prediction output as Eq. (3):

$$Y_t = F(E_{fusion}, \theta) \quad (3)$$

where $F$ is the model and $\theta$ represents parameters.

Fault probability estimation is computed as in Eq. (4):

$$P_t = \frac{1}{1 + e^{-Y_t}} \quad (4)$$

where $P_t$ is the probability of fault occurrence.

Training loss function for optimization is defined as in Eq.(5):

$$L = -\sum_{t} [y_t log(P_t) + (1 - y_t) log(1 - P_t)] \quad (5)$$

where $y_t$ is the true label.

Average inference latency is calculated as in Eq.(6):

$$L_t = \frac{1}{N}\sum_{i=1}^{N} T_i \quad (6)$$

where $T_i$ is processing time per sample.

Energy consumption during inference is expressed as Eq.(7):

$$E_t = \frac{E_{used}}{E_{total}} \quad (7)$$

where $E_{used}$ is consumed energy and $E_{total}$ is available energy.

Dataset logging for adaptive learning is defined as in Eq.(8):

$$D_t = (x_t, E_t, L_t, P_t) \quad (8)$$

where $x_t$ is input data at time $t$.

Adaptive feedback reward function is given in Eq.(9):

$$R_t = \alpha A_t + \beta D_t - \gamma L_t - \delta E_t \quad (9)$$

where $A_t$ is accuracy, $D_t$ adaptability score and $\alpha, \beta, \gamma, \delta$ are weights.

System Performance Index (SPI) is computed as Eq.(10):

$$SPI = \frac{A_t + D_t}{L_t + E_t} \quad (10)$$

where higher SPI indicates better system performance.

The algorithm performs real-time fault detection by processing multi-modal sensor data, generating embeddings, applying deep learning inference and optimizing performance using adaptive feedback. It continuously monitors latency and energy, dynamically updating model parameters to maintain accuracy and efficiency under varying CPS conditions.

**Algorithm 1: Real-Time Adaptive Fault Detection in CPS Using Edge Deep Learning**

Input: Multi-modal CPS sensor data

Output: Fault detection decision and system performance index

**Steps:**

1. Initialize model parameters $\theta$, thresholds
2. Acquire sensor data $x_t$
3. Compute embeddings using Eq.(1)
4. Perform fusion using Eq.(2)
5. Generate prediction using Eq.(3–4)
6. Compute loss using Eq.(5)
7. Measure latency & energy using Eq.(6–7)

8. Update dataset using Eq.(8)
9. Optimize using reward Eq.(9–10)

## 5. Experimental set-up

Simulated and real-life CPS data are used to evaluate the proposed system until the manager is confident in the system's performance under dynamic conditions. Embedded edge hardware is an embedded system comprising sensors and communication modules that is being experimented on. Accuracy, energy consumption, latency and system reliability are used to measure the performance. The baseline models are compared with this to confirm that real-time detection and efficiency have improved across various CPS scenarios.

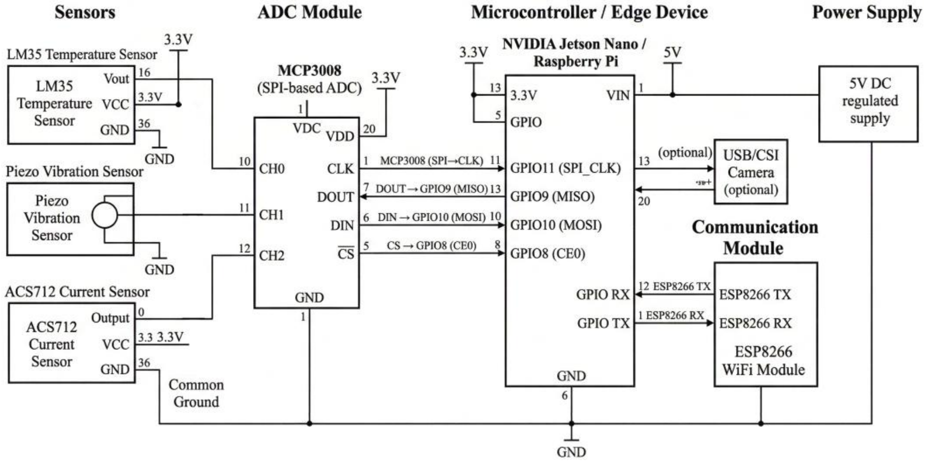


**Figure 3:** Wiring diagram of CPS fault detection system with embedded edge components

Fig. 3 illustrates relationships among sensors, embedded processors, memory modules, communication interfaces as well as power modules. Sensors send data to the edge device to be processed and feedback information controls the system behaviour, which provides efficient and real-time fault detection.

Table 2 will present the summarised datasets and they include size, modality and application domain as well as the numerical properties that can be used to validate the performance of the proposed fault detection framework.

**Table 2:** CPS datasets and characteristics used for experimental evaluation

| Dataset Name | Domain | Modalities | Samples | Fault Classes | Data Size (MB) | Avg. Sampling Rate (Hz) |
|---|---|---|---|---|---|---|
| Smart Grid Dataset | Power Systems | Voltage, Current Sensors | 50,000 | 5 | 120 | 60 |
| IIoT Fault Dataset | Industrial IoT | Temperature, Pressure, Vibration | 35,000 | 6 | 95 | 50 |

| | | | | | | |
|---|---|---|---|---|---|---|
| CPS Network Simulation | Network CPS | Sensor Logs, Traffic Data | 40,000 | 4 | 110 | 55 |

Table 3 provides the hardware and software settings employed in the experiments, which, in this case, reflect the computational processing capacities, edge device specifications, and structures that guarantee reproducibility and real-time processing.

**Table 3:** System configuration for edge-based fault detection

| **Component** | **Specification** |
|---|---|
| Edge Device | NVIDIA Jetson Nano |
| CPU | Quad-core ARM Cortex |
| GPU | 128-core Maxwell |
| RAM | 8 GB |
| Framework | PyTorch 2.2 |
| OS | Ubuntu Embedded |
| Tools | CUDA, TensorRT |

The test environment will create a holistic testing environment using a variety of datasets and edge hardware configurations. In the next section, detailed results and comparisons of major models are provided to demonstrate improvements in accuracy, latency, energy efficiency, and adaptability.

## 6. Results and Discussion

The proposed model (Model E) is compared to four baseline models, namely Model A (SVM), Model B (CNN), Model C (LSTM) and Model D (Edge-optimized DL). Findings indicate that Model E will always outperform all baselines across key measures. It achieves higher fault detection accuracy (90.5%) compared to Model A (71.2%), Model B (79.5%), Model C (83.1%) and Model D (85.7%). Also, it is much lower in latency and energy usage and offers a much better adaptability, showing its suitability in real-time CPS fault detection on edge devices.

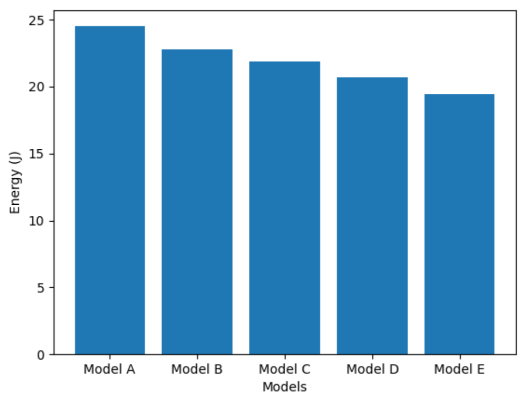


**Figure 4:** Energy consumption comparison across models

Fig. 4 compares the power usage in all the models, indicating that the proposed model uses the minimum amount of energy per inference cycle. This underscores its effectiveness and capability to run on the resource-constrained edge devices.

Table 4 presents a descriptive summary of the energy consumption measure for each of the models compared; it shows a considerable decrease in the energy consumption measure of the proposed framework compared with baseline approaches.

**Table 4:** Energy performance comparison

| Model | Avg Energy (J) | Peak Energy (J) | Efficiency (%) |
|---|---|---|---|
| A | 24.5 | 27.2 | 60.1 |
| B | 22.8 | 25.4 | 64.3 |
| C | 21.9 | 24.1 | 66.8 |
| D | 20.7 | 22.8 | 70.5 |
| E | 19.4 | 21.6 | 74.2 |

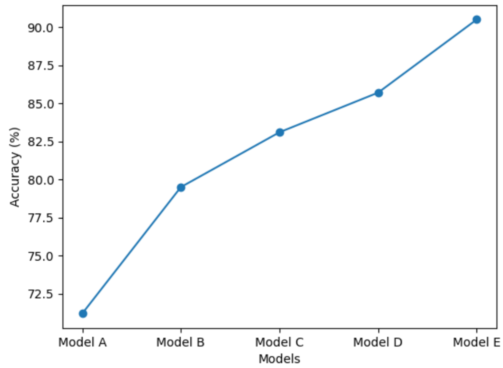


**Figure 5:** Fault detection accuracy comparison across models

Fig. 5 shows the accuracy of fault detection across CPS datasets. The proposed model is the most accurate, as it has better learning ability and greater resilience compared with traditional and deep learning baseline models.

Table 5 presents the accuracy rates of the models, indicating that the proposed approach is the most successful in fault detection across varied CPS conditions.

**Table 5:** Accuracy comparison

| Model | Accuracy (%) |
|---|---|
| A | 71.2 |
| B | 79.5 |

| | |
|---|---|
| C | 83.1 |
| D | 85.7 |
| E | 90.5 |

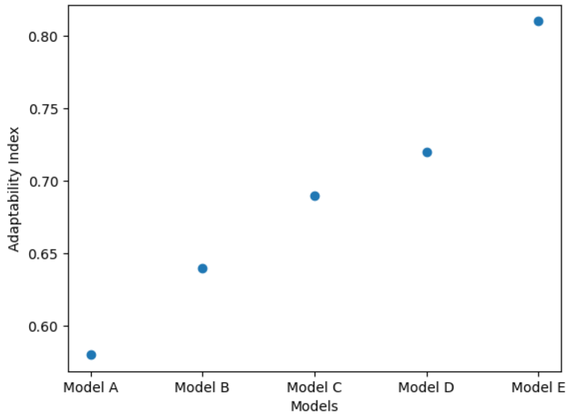


**Figure 6:** Adaptability comparison under dynamic conditions.

Fig. 6 shows adaptability index under dynamic CPS conditions. The proposed model demonstrates the highest adaptability, effectively adjusting to varying inputs and environmental changes compared to baseline models.

Table 6 provides adaptability metrics, confirming improved robustness and dynamic response capability of the proposed model compared to other approaches.

**Table 6:** Adaptability index comparison

| **Model** | **Adaptability Index** |
|---|---|
| A | 0.58 |
| B | 0.64 |
| C | 0.69 |
| D | 0.72 |
| E | 0.81 |

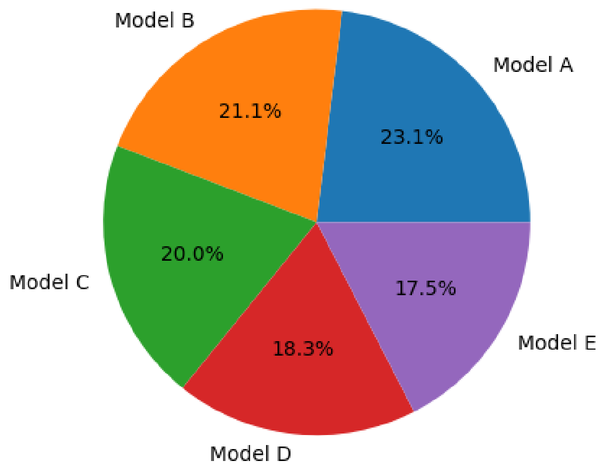


**Figure 7:** Latency comparison under real-time conditions.

Fig. 7 compares response latency across models, showing that the proposed model achieves the lowest inference time, ensuring faster decision-making suitable for real-time CPS applications.

Table 7 presents the latency results for all models, showing a substantial decrease in response time with the proposed framework.

**Table 7:** Latency performance comparison

| Model | Latency (ms) |
|---|---|
| A | 340 |
| B | 310 |
| C | 295 |
| D | 270 |
| E | 257 |

These findings indicate that the suggested model obtains equal improvements in accuracy, latency, energy efficiency and adaptability. The next section concludes the study and suggests future research directions to improve CPS fault detection systems.

### 7. Conclusion and Future Work

In this paper, a real-time fault detection system for cyber-physical systems with embedded edge devices is proposed, implemented using deep learning. The proposed model (Model E) achieves 90.5% accuracy, outperforming Model A (71.2%), Model B (79.5%), Model C (83.1%) and Model D (85.7%). It has a latency of 257 ms, down from 340ms in Model A and uses 19.4 J, which is more than 20 times less efficient. The adaptability index goes up to 0.81 and this is very dynamic in dynamic environments. These findings confirm the usefulness of edge computing with deep learning. Federated learning, multi-agent CPS coordination, and improving security-conscious fault detection to enable scalable and resilient deployments will become the focus of future work.

### Conflict of Interest

The authors have no conflict of Interest.

## References


[1] Zhang, K., Shi, Y., Karnouskos, S., Sauter, T., Fang, H., & Colombo, A. W. (2022). Advancements in industrial cyber-physical systems: An overview and perspectives. IEEE Transactions on Industrial Informatics, 19(1), 716-729. https://doi.org/10.1109/TII.2022.3199481

[2] Boi-Ukeme, J., Ruiz-Martin, C., & Wainer, G. (2020, December). Real-time fault detection and diagnosis of CPS faults in DEVS. In 2020 IEEE 6th International Conference on Dependability in Sensor, Cloud and Big Data Systems and Application (DependSys) (pp. 57-64). IEEE. https://doi.org/10.1109/DependSys51298.2020.00017

[3] Kathiravelu, P., Van Roy, P., & Veiga, L. (2019). SD-CPS: software-defined cyber-physical systems. Taming the challenges of CPS with workflows at the edge. Cluster Computing, 22(3), 661-677.doi: 10.1007/s10586-018-2874-8

[4] Zhou, X., Yang, Z., Ni, M., Lin, H., Li, M., & Tang, Y. (2020). Analysis of the impact of combined information-physical-failure on distribution network CPS. IEEE Access, 8, 44140-44152. https://doi.org/10.1109/ACCESS.2020.2978113

[5] Zhao, H., Liu, H., Hu, W., & Yan, X. (2018). Anomaly detection and fault analysis of wind turbine components based on deep learning network. Renewable energy, 127, 825-834. https://doi.org/10.1016/j.renene.2018.05.024

[6] Sheela, K. S., Pavithra, A., Jeevanandham, S., Sundarrajan, M., & Choudhry, M. D. (2025, December). Multi-Sensor Data Fusion with Explainable AI for Anomaly Detection in Industrial CPS. In 2025 First International Conference of Advances in Engineering and Computing Technologies for Sustainable Development (AECTSD) (pp. 1-6). IEEE. https://doi.org/10.1109/AECTSD65988.2025.11410609

[7] Zhang, J., Yang, S., Yang, Y., Yuan, Y., & Wang, X. (2026). A Hybrid Forecasting Framework with Adaptive Parameter Optimization and Multi-Scale Feature Fusion for Non-Stationary Power Grid CPS Time Series. Journal of Electrical Engineering & Technology, 1-22.doi: 10.1007/s42835-025-02580-0

[8] Zhang, N. (2021). A cloud-based platform for big data-driven cps modeling of robots. IEEE Access, 9, 34667-34680. https://doi.org/10.1109/ACCESS.2021.3061477

[9] Funchal, G., Pedrosa, T., De la Prieta, F., & Leitão, P. (2026). A Cloud-Driven Support Layer for Enhancing Distributed IDS in IoT Networks. SN Computer Science, 7(2), 186.doi: 10.1007/s42979-026-04750-x

[10] Cao, K., Hu, S., Shi, Y., Colombo, A. W., Karnouskos, S., & Li, X. (2021). A survey on edge and edge-cloud computing assisted cyber-physical systems. IEEE Transactions on Industrial Informatics, 17(11), 7806-7819. https://doi.org/10.1109/TII.2021.3073066

[11] Selvi, G. V., & Kumari, V. S. (2026). Edge AI: Transforming Real-Time Decision-Making in the Internet of Energy. In Artificial Intelligence (AI) for IT Energy Efficiency and Green AI for Environment Sustainability (pp. 539-564). Cham: Springer Nature Switzerland.doi: 10.1007/978-3-031-89420-6_26

[12] Elsayed, R., Ismail, M. I., Ashour, A. F., Sakr, H. A., Ibrahem, M. I., & Fouda, M. M. (2025, September). A Review of Smart Building Management Systems: CPS Applications for Energy-Efficient Monitoring and Control. In 2025 3rd International Conference on Artificial

Intelligence, Blockchain and Internet of Things (AIBThings) (pp. 1-6). IEEE. https://doi.org/10.1109/AIBThings66987.2025.11296174

[13] Fezai, R., Dhibi, K., Mansouri, M., Trabelsi, M., Hajji, M., Bouzrara, K., ... & Nounou, M. (2020). Effective random forest-based fault detection and diagnosis for wind energy conversion systems. IEEE Sensors Journal, 21(5), 6914-6921. https://doi.org/10.1109/JSEN.2020.3037237

[14] Ruan, H., Dorneanu, B., Arellano-Garcia, H., Xiao, P., & Zhang, L. (2022). Deep learning-based fault prediction in wireless sensor network embedded cyber-physical systems for industrial processes. Ieee Access, 10, 10867-10879. https://doi.org/10.1109/ACCESS.2022.3144333

[15] Cui, J., Long, J., Min, E., Liu, Q., & Li, Q. (2018, June). Comparative study of CNN and RNN for deep learning based intrusion detection system. In International Conference on Cloud Computing and Security (pp. 159-170). Cham: Springer International Publishing.doi: 10.1007/978-3-030-00018-9_15

[16] Qiu, T., Chi, J., Zhou, X., Ning, Z., Atiquzzaman, M., & Wu, D. O. (2020). Edge computing in the industrial internet of things: Architecture, advances and challenges. IEEE communications surveys & tutorials, 22(4), 2462-2488. https://doi.org/10.1109/COMST.2020.3009103

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

[18] Malaraju, S. K., & Madishetty, S. K. (2025, November). AI-Augmented Compiler Optimization for Energy Efficient Software Execution on Embedded Systems. In 2025 14th International Conference on System Modeling & Advancement in Research Trends (SMART) (pp. 1-6). IEEE. https://doi.org/10.1109/SMART66937.2025.11389313

[19] Lilhore, U. K., Simaiya, S., Sharma, Y. K., Rai, A. K., Padmaja, S. M., Nabilal, K. V., ... & Alsufyani, H. (2025). Cloud-edge hybrid deep learning framework for scalable IoT resource optimization. Journal of Cloud Computing, 14(1), 5.doi: 10.1186/s13677-025-00729-w

[20] Olowononi, F. O., Rawat, D. B., & Liu, C. (2020). Resilient machine learning for networked cyber physical systems: A survey for machine learning security to securing machine learning for CPS. IEEE Communications Surveys & Tutorials, 23(1), 524-552.doi: https://doi.org/10.1109/COMST.2020.3036778

[21] Tumkur, S. D., Iyer, J., & Kurra, S. B. (2025, November). Explainable Multi-Modal Deep Learning Framework for Enhancing Cyber-Physical Systems Security. In 2025 IEEE 12th International Conference on Cyber Security and Cloud Computing (CSCloud) (pp. 48-55). IEEE. https://doi.org/10.1109/CSCloud66326.2025.00015

[22] Marino, R., Wisultschew, C., Otero, A., Lanza-Gutierrez, J. M., Portilla, J., & De la Torre, E. (2020). A machine-learning-based distributed system for fault diagnosis with scalable detection quality in industrial IoT. IEEE Internet of Things Journal, 8(6), 4339-4352. https://doi.org/10.1109/JIOT.2020.3026211