

Embedded Intelligent Systems for Advanced Engineering Applications

Dr. Sophie de Vries

Associate Professor, Amsterdam Institute of Advanced Technology, Netherlands

Abstract

The rapid development of embedded computing, artificial intelligence, edge processing, and intelligent sensing has transformed conventional embedded systems into sophisticated computational platforms capable of autonomous monitoring, decision-making, and control. Advanced engineering applications increasingly require embedded systems that can process complex data locally while operating under strict constraints involving power consumption, memory, computational resources, reliability, and real-time response. Embedded intelligent systems address these requirements by integrating machine learning, sensor fusion, adaptive algorithms, real-time operating environments, and specialized hardware accelerators into engineering devices. This paper examines the role of embedded intelligent systems in advanced engineering applications, focusing on intelligent manufacturing, robotics, transportation, energy systems, infrastructure monitoring, and autonomous engineering platforms. A conceptual and analytical research methodology is adopted to identify the principal architectural components and evaluate their contribution to intelligent engineering operations. The study develops a layered framework integrating sensing, embedded computation, artificial intelligence, communication, decision-making, and actuation.

The analysis indicates that embedded intelligence can reduce dependence on remote computing infrastructure by enabling real-time data processing directly at or near the point of acquisition. This capability can improve response speed, reduce communication requirements, enhance operational autonomy, and support predictive maintenance and intelligent control. However, embedded intelligent systems also face challenges associated with limited computational resources, energy consumption, model complexity, cybersecurity, thermal constraints, software-hardware integration, and long-term system reliability. The study emphasizes the importance of lightweight artificial intelligence models, hardware acceleration, edge intelligence, secure firmware, and adaptive resource management for practical implementation. The proposed framework demonstrates how embedded intelligence can serve as an enabling technology for advanced engineering systems that require fast, reliable, and context-aware decision-making. The study concludes that future engineering applications will increasingly depend on tightly integrated embedded intelligence capable of combining real-time sensing, local computation, secure communication, and autonomous decision support within resource-constrained environments.

Keywords: Embedded intelligent systems; Embedded artificial intelligence; Edge intelligence; Engineering applications; Intelligent control; Machine learning; Real-time systems; Embedded computing; Smart engineering; Autonomous systems

1. Introduction

Embedded systems have become fundamental components of modern engineering infrastructure, providing computational capabilities within industrial machines, vehicles, robotic platforms, energy systems, medical devices, and intelligent infrastructure. Conventional embedded systems generally perform predefined computational functions according to programmed rules. However, the increasing availability of machine-learning algorithms, advanced sensors, and efficient processors has enabled embedded platforms to evolve toward intelligent systems capable of learning from data, identifying patterns, and supporting autonomous decisions.

The integration of artificial intelligence into embedded platforms is particularly important for engineering applications requiring real-time responses. Traditional cloud-based architectures often transmit sensor data to remote computing platforms for processing. Although cloud computing provides substantial computational capacity, communication delays, network dependency, and data-transfer requirements can become problematic for time-sensitive engineering applications. Embedded intelligence allows selected computational tasks to be performed locally, reducing the need to continuously transfer raw data to remote systems.

Advanced engineering environments also generate increasingly complex data. Sensors may simultaneously capture temperature, vibration, pressure, acoustic signals, images, electrical characteristics, and motion information. Processing these data locally can enable embedded systems to identify abnormal conditions, estimate equipment states, optimize control parameters, and initiate corrective actions. Sensor fusion and machine learning can therefore transform embedded devices from passive data collectors into active decision-making components.

The development of specialized hardware has further expanded the capabilities of intelligent embedded systems. Graphics processing units, neural processing units, field-programmable gate arrays, digital signal processors, and application-specific accelerators can execute machine-learning workloads more efficiently than conventional microcontrollers alone. These technologies make it increasingly practical to deploy sophisticated algorithms in resource-constrained engineering devices.

Despite these opportunities, embedded intelligence introduces important engineering challenges. Intelligent models may require significant memory and computational resources, while embedded platforms often operate under strict power and thermal constraints. Security is also critical because compromised firmware, models, sensors, or communication interfaces can affect physical operations. This paper therefore examines embedded intelligent systems from an integrated engineering perspective and proposes a conceptual framework for their application across advanced engineering domains.

2. Objectives

2.1 Enhancing Local Intelligence in Engineering Systems

The first objective is to examine how embedded artificial intelligence can enhance local decision-making in advanced engineering applications. Embedded systems can process sensor information directly at the device level, allowing abnormal conditions, equipment states, and operational patterns to be identified without continuously transmitting raw information to centralized platforms. This capability is particularly valuable for applications requiring rapid responses and reliable operation under variable network conditions.

The objective also considers the role of lightweight machine-learning algorithms and specialized hardware in enabling local intelligence. Engineering embedded systems frequently operate with limited

memory, processing capacity, and energy availability. Therefore, intelligent models must be optimized for resource-constrained environments. Model compression, quantization, efficient architectures, and hardware acceleration can enable useful AI capabilities while maintaining acceptable computational and energy performance.

2.2 Developing Reliable Embedded Intelligence Architectures

The second objective is to develop a conceptual architecture for integrating sensing, embedded computation, artificial intelligence, communication, and actuation. The architecture is intended to support engineering applications where intelligent decisions must be generated close to physical processes. It emphasizes real-time processing, secure communication, adaptive resource management, and reliable interaction between computational and physical components.

The objective further addresses system reliability and security. Embedded intelligent systems may operate continuously in industrial and infrastructure environments where failures can have significant consequences. Secure boot mechanisms, firmware protection, authenticated updates, model integrity, fault detection, and cybersecurity monitoring should therefore be incorporated into the system architecture. The proposed approach treats intelligence, security, and reliability as interconnected engineering requirements.

3. Literature Review

3.1 Evolution of Embedded Systems

Embedded computing has evolved from simple microcontroller-based systems toward heterogeneous platforms capable of supporting complex software and artificial intelligence. Early embedded systems primarily executed deterministic control functions, while modern platforms increasingly incorporate operating systems, communication interfaces, advanced sensors, and machine-learning capabilities.

This evolution has enabled embedded devices to participate in larger cyber-physical ecosystems. Instead of operating independently, embedded systems can exchange information with other machines, edge nodes, cloud services, and engineering control platforms. Such connectivity creates opportunities for coordinated intelligence but also introduces additional requirements for communication security, interoperability, and resource management.

3.2 Embedded Artificial Intelligence

Embedded AI refers to the deployment of machine-learning capabilities directly on embedded hardware. The approach allows data to be processed locally rather than requiring continuous transmission to centralized computing platforms. Local inference can reduce latency, decrease network traffic, and improve privacy for applications where sensitive data are generated.

However, embedded AI requires algorithms to be optimized according to hardware limitations. Large neural networks may exceed the memory or computational capacity of embedded devices. Techniques such as pruning, quantization, knowledge distillation, and hardware-specific optimization can reduce model size and computational requirements.

3.3 Edge Intelligence and Engineering Systems

Edge computing extends embedded intelligence by distributing computation closer to data sources. Embedded devices can perform initial processing, while edge nodes execute more computationally

demanding tasks. This hierarchical approach enables engineering systems to balance local responsiveness with access to greater computational resources.

Edge intelligence is particularly relevant to manufacturing, autonomous vehicles, infrastructure monitoring, and industrial IoT systems. Local devices can detect immediate events, while edge platforms can aggregate information across multiple devices and perform more complex analysis. This arrangement can improve scalability while reducing unnecessary communication with centralized cloud systems.

3.4 Intelligent Control and Cyber-Physical Systems

Embedded intelligent systems are increasingly integrated into cyber-physical systems in which computational decisions directly influence physical processes. Intelligent controllers can adapt operating parameters based on sensor information, equipment conditions, or environmental changes. Such systems can support predictive maintenance, adaptive manufacturing, autonomous navigation, and energy optimization.

However, intelligent control introduces additional validation requirements. Machine-learning decisions may be less predictable than deterministic control logic, particularly when operating conditions differ from training data. Engineering systems therefore require robust validation, safety mechanisms, fallback controls, and human oversight where appropriate.

4. Methodology

4.1 Research Design

This study uses a conceptual and analytical research design to examine embedded intelligent systems for advanced engineering applications. The methodology identifies the major components required to transform conventional embedded platforms into intelligent engineering systems. These components include sensing, data preprocessing, local computation, machine-learning inference, communication, decision-making, and actuation.

The analysis evaluates these components according to engineering requirements such as real-time performance, reliability, energy efficiency, computational capability, scalability, security, and adaptability. The study adopts a system-level perspective because embedded intelligence cannot be evaluated solely according to machine-learning accuracy. Practical effectiveness depends on the interaction between algorithms, hardware, sensors, software, communication networks, and physical processes.

4.2 Intelligent Processing Architecture

The second methodological component evaluates how intelligence can be distributed between embedded devices, edge nodes, and cloud platforms. Local processing is prioritized for time-sensitive tasks, while computationally intensive or long-term analytical workloads can be transferred to higher-level computing resources. This hierarchical approach allows engineering systems to balance response time, computational requirements, and communication costs.

The methodology also considers data preprocessing and sensor fusion. Raw sensor information may contain noise, redundancy, or inconsistent measurements. Embedded preprocessing can reduce data volume and improve the quality of information supplied to machine-learning models. Sensor fusion can combine multiple data sources to provide more reliable representations of physical system states.

4.3 Embedded Intelligence Requirements

The following key requirements guide the proposed framework:

- **Real-time processing:** Decisions should be generated within application-specific response limits.
- **Computational efficiency:** Algorithms should operate within available processor and memory resources.
- **Energy efficiency:** Intelligent processing should minimize unnecessary power consumption.
- **Reliability:** Embedded systems should continue operating under variable conditions.
- **Security:** Firmware, models, data, and communication interfaces require protection.
- **Adaptability:** Systems should accommodate changing operational conditions.
- **Hardware compatibility:** Algorithms should match available processing architectures.
- **Communication efficiency:** Only necessary information should be transmitted.
- **Model integrity:** AI models should be protected against unauthorized modification.
- **Fault tolerance:** Systems should provide safe fallback mechanisms during failures.

These requirements establish a practical foundation for evaluating embedded intelligence. They demonstrate that an accurate AI model may not be suitable for embedded deployment if it requires excessive memory, processing time, or energy. Successful implementation requires joint optimization of algorithmic and hardware characteristics.

4.4 Framework Development

The final methodological stage develops a layered framework consisting of sensing, preprocessing, embedded intelligence, communication, edge coordination, decision-making, and actuation. Security and reliability mechanisms operate across all layers. The framework supports a hierarchical distribution of computation, allowing immediate decisions to remain local while more complex analysis is transferred to edge or cloud infrastructure.

The proposed architecture also incorporates feedback mechanisms. System performance and model behavior are continuously monitored to identify degradation, sensor faults, unexpected operating conditions, or changes in the engineering environment. Model updates can be securely delivered when required. This adaptive architecture allows embedded intelligent systems to remain useful throughout changing operational conditions while maintaining appropriate control over software and model modifications.

5. Analysis

5.1 Architecture of Embedded Intelligent Systems

An embedded intelligent engineering system can be understood as an integrated platform consisting of sensing, processing, intelligence, communication, and actuation components. Sensors provide information about physical conditions, while preprocessing mechanisms remove noise and reduce unnecessary data. The embedded processor then executes machine-learning models or intelligent control algorithms to generate decisions.

Communication mechanisms connect embedded devices with other systems and edge platforms. However, not all data must be transmitted continuously. Local processing can identify important events and transmit only relevant information, thereby reducing network traffic. This architecture is particularly beneficial in industrial environments containing large numbers of sensors.

Table 1: Core Components of Embedded Intelligent Engineering Systems

Component	Primary Function	Engineering Contribution
Sensors	Capture physical information	Provides real-time system observations
Data Preprocessing	Filters and transforms data	Improves data quality and efficiency
Embedded Processor	Executes computational tasks	Enables local processing
AI/ML Model	Performs intelligent inference	Supports prediction and classification
Hardware Accelerator	Speeds intensive workloads	Improves AI execution efficiency
Communication Interface	Exchanges information	Enables system connectivity
Edge Node	Provides additional computation	Supports distributed intelligence
Actuator	Executes control decisions	Connects intelligence to physical processes
Security Module	Protects system resources	Improves cybersecurity
Memory and Storage	Maintains programs and data	Supports reliable system operation

Interpretation: The table demonstrates that embedded intelligence depends on the coordinated operation of multiple components. Sensors provide the data foundation, while preprocessing and embedded computation transform raw information into useful intelligence.

AI models provide decision-making capabilities, and actuators translate decisions into physical actions. Hardware accelerators can improve computational performance, whereas security modules protect software, models, and communication interfaces. The integration of these components enables embedded systems to operate as intelligent engineering platforms rather than isolated computational devices.

5.2 Engineering Applications

Embedded intelligent systems have broad applications across modern engineering. In manufacturing, intelligent embedded controllers can monitor equipment conditions, detect production anomalies, and adjust operating parameters. In robotics, embedded AI enables perception, localization, motion planning, and adaptive control. Automotive systems can use embedded intelligence for driver assistance, object detection, and vehicle monitoring.

Energy systems can also benefit from embedded intelligence. Local devices can monitor voltage, current, temperature, and equipment conditions while identifying abnormal behavior. Infrastructure monitoring systems can use embedded processors to analyze vibration, structural conditions, and environmental measurements. These applications demonstrate the importance of local intelligence where rapid decisions and continuous operation are required.

5.3 Performance and Resource Optimization

Embedded intelligent systems must balance computational capability against resource limitations. Increasing model complexity may improve prediction accuracy but can also increase memory use, execution time, energy consumption, and thermal output. Engineering designers must therefore select models according to application requirements rather than simply maximizing algorithmic complexity.

Table 2: Performance Factors for Embedded Intelligent Systems

Factor	Primary Concern	Optimization Approach
Processing Speed	Real-time response	Hardware acceleration and efficient models
Memory	Limited embedded resources	Model compression and quantization
Energy	Battery and thermal constraints	Low-power inference
Accuracy	Decision reliability	Model optimization and validation
Latency	Immediate control requirements	Local inference
Security	Protection from attacks	Secure boot and authenticated updates
Reliability	Continuous operation	Fault tolerance and redundancy
Adaptability	Changing conditions	Continual model monitoring
Communication	Bandwidth consumption	Local preprocessing
Thermal Performance	Heat generation	Efficient hardware and workload management

Interpretation: The table shows that embedded AI performance involves multiple competing factors. High model accuracy alone does not guarantee engineering suitability. Systems must also meet processing, memory, energy, latency, security, reliability, and thermal requirements.

Local inference can reduce communication latency, while model compression and hardware acceleration can improve computational efficiency. A balanced optimization strategy is therefore necessary to achieve reliable intelligent performance within the physical limitations of embedded platforms.

6. Discussion

6.1 Benefits of Embedded Intelligence

Embedded intelligence can significantly improve the responsiveness of engineering systems by moving computational decision-making closer to physical processes. Local inference eliminates or reduces dependence on remote servers for immediate decisions. This is particularly important in applications such as robotic control, equipment protection, autonomous navigation, and real-time industrial monitoring.

Another benefit is reduced communication requirements. Instead of transmitting continuous raw sensor streams, embedded systems can process information locally and transmit selected events, features, or decisions. This can reduce bandwidth requirements and improve scalability in environments containing large numbers of connected devices.

6.2 Hardware and Algorithm Co-Design

The effectiveness of embedded intelligence depends strongly on the interaction between algorithms and hardware. A machine-learning model designed for high-performance computing may not be appropriate for a microcontroller or low-power embedded processor. Hardware-aware optimization is therefore essential for practical deployment.

Specialized accelerators can significantly improve inference performance for neural networks and signal-processing workloads. However, hardware acceleration may increase system complexity and development costs. Engineers must therefore evaluate performance benefits against power consumption, hardware availability, software compatibility, and long-term maintainability.

6.3 Reliability and Security

Embedded intelligent systems often operate within environments where failures can affect physical processes. Reliability mechanisms such as watchdog timers, redundancy, fault detection, safe fallback modes, and system-health monitoring are therefore important. Intelligent models should not become single points of failure for critical engineering functions.

Security is equally important. Attackers may attempt to modify firmware, manipulate sensor data, replace AI models, or compromise communication interfaces. Secure boot, firmware authentication, encrypted communication, access control, model integrity verification, and secure update mechanisms can reduce these risks. Security should be incorporated during system design rather than added after deployment.

7. Challenges

One major challenge is the limited computational resources of embedded platforms. Advanced AI models can require substantial memory, processing capacity, and energy. Deploying such models on small devices may require extensive optimization. Model compression, quantization, pruning, efficient neural architectures, and hardware acceleration can reduce these requirements, but optimization may involve trade-offs in accuracy and flexibility.

Energy consumption is another important challenge. Battery-powered embedded systems may need to operate for extended periods without physical maintenance. Continuous AI inference can increase energy consumption significantly. Designers must therefore develop energy-aware inference strategies, event-driven processing, low-power hardware configurations, and adaptive computational workloads.

Cybersecurity presents additional concerns because embedded systems are often deployed in physically accessible and operationally critical environments. Compromised firmware or manipulated sensor data can affect intelligent decisions and physical processes. Secure hardware, trusted execution mechanisms, authenticated software updates, encrypted communication, and continuous security monitoring are required to protect intelligent embedded platforms.

Model reliability and explainability also remain challenging. Machine-learning systems may produce incorrect decisions when exposed to conditions that differ substantially from their training data. Engineering applications require predictable and safe behavior, especially when AI decisions influence physical equipment. Robust validation, uncertainty estimation, anomaly detection, and fallback control mechanisms can help reduce the consequences of incorrect predictions.

8. Future Research

Future research should focus on developing highly efficient embedded AI architectures capable of performing increasingly complex tasks within strict energy and computational constraints. Research into tiny machine-learning models, neuromorphic computing, specialized AI accelerators, model quantization, and adaptive inference could improve the feasibility of intelligent processing on low-power devices. Benchmarking should evaluate not only model accuracy but also energy consumption, thermal performance, memory requirements, latency, and reliability under realistic engineering workloads.

Another important direction is the development of collaborative embedded-edge intelligence. Future systems could dynamically distribute workloads among embedded devices, edge nodes, and cloud platforms according to latency, computational demand, network availability, and energy conditions. Such architectures could allow embedded devices to make immediate decisions while using edge infrastructure for more complex analysis. Research should investigate intelligent task allocation, federated learning, model synchronization, and secure distributed inference for large-scale engineering environments.

Future research should also address trustworthy embedded AI. Engineering applications require systems that can explain decisions, identify uncertainty, detect model degradation, and safely recover from abnormal conditions. Research should explore explainable AI, robust learning, adversarial resilience, continual learning, and formal verification techniques for embedded intelligence. Integration of these approaches with secure hardware and safety-oriented engineering practices could provide stronger foundations for autonomous and intelligent engineering systems.

9. Limitations

This study presents a conceptual framework and does not include experimental implementation of the proposed embedded intelligent architecture on specific hardware platforms. Consequently, actual performance differences among processors, accelerators, memory architectures, and AI models were not experimentally measured. Future empirical research is required to validate processing latency, energy consumption, thermal behavior, and model accuracy under realistic engineering conditions.

Another limitation concerns the diversity of embedded engineering applications. Requirements vary significantly among robotics, manufacturing, transportation, energy, infrastructure, and autonomous systems. An architecture suitable for one application may require substantial modification for another. Therefore, the proposed framework should be adapted according to application-specific safety, performance, environmental, and computational requirements.

The study also recognizes that embedded intelligence introduces dependencies on software and model quality. Incorrect sensor inputs, model degradation, software vulnerabilities, and unexpected environmental conditions can affect system decisions. Consequently, intelligent embedded systems should not be evaluated solely according to machine-learning performance but through comprehensive assessment of hardware reliability, cybersecurity, safety, and operational behavior.

10. Conclusion

Embedded intelligent systems are becoming increasingly important in advanced engineering applications because they combine sensing, computation, artificial intelligence, communication, and control within compact and resource-constrained platforms. The integration of machine learning into embedded devices enables local interpretation of engineering data and supports rapid decisions without requiring continuous dependence on remote computing infrastructure. This capability can improve response time, reduce communication requirements, and enable greater autonomy in manufacturing, robotics, transportation, energy, and infrastructure systems.

The analysis demonstrates that successful embedded intelligence requires careful coordination between algorithms and hardware. AI models must be optimized according to processing capacity, memory availability, energy consumption, thermal conditions, and latency requirements. Techniques such as model compression, quantization, efficient architectures, and hardware acceleration can help overcome resource limitations. At the same time, secure firmware, authenticated updates, model integrity protection, and fault-tolerant mechanisms are essential for reliable deployment in engineering environments.

Overall, embedded intelligent systems provide a strong technological foundation for the development of responsive and autonomous engineering platforms. Future architectures are likely to combine embedded intelligence with edge computing, distributed learning, specialized accelerators, secure hardware, and advanced sensing technologies. However, practical deployment requires more than algorithmic innovation. It requires integrated consideration of performance, safety, security, reliability, energy efficiency, and maintainability. With appropriate hardware-software co-design and trustworthy AI practices, embedded intelligent systems can play a central role in advancing the capabilities of next-generation engineering applications.

References

1. Mendez, J., Bierzynski, K., Cuéllar, M. P., & Morales, D. P. (2022). Edge intelligence: Concepts, architectures, applications, and future directions. *ACM Transactions on Embedded Computing Systems*, 21(5), Article 48. <https://doi.org/10.1145/3486674>
2. Seng, K. P., & Ang, L.-M. (2022). Embedded intelligence: State-of-the-art and research challenges. *IEEE Access*, 10, 59236–59258. <https://doi.org/10.1109/ACCESS.2022.3175574>
3. Singh, R., & Gill, S. S. (2023). Edge AI: A survey. *Internet of Things and Cyber-Physical Systems*, 3, 71–92. <https://doi.org/10.1016/j.iotcps.2023.02.004>
4. Zhu, R., Anjum, A., Li, H., & Ma, M. (2023). Edge intelligence-enabled cyber-physical systems. *Concurrency and Computation: Practice and Experience*, 35(13), Article e7500. <https://doi.org/10.1002/cpe.7500>
5. Marwedel, P. (2018). *Embedded system design: Embedded systems foundations of cyber-physical systems, and the Internet of Things* (3rd ed.). Springer. <https://doi.org/10.1007/978-3-319-56045-8>
6. Seshia, S. A., Hu, S., Li, W., & Zhu, Q. (2017). Design automation of cyber-physical systems: Challenges, advances, and opportunities. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 36(9), 1421–1434. <https://doi.org/10.1109/TCAD.2016.2633961>
7. Lee, E. A., & Seshia, S. A. (2017). *Introduction to embedded systems: A cyber-physical systems approach* (2nd ed.). MIT Press. <https://LeeSeshia.org>

8. Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of Things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys & Tutorials*, 17(4), 2347–2376. <https://doi.org/10.1109/COMST.2015.2444095>
9. Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, 3(5), 637–646. <https://doi.org/10.1109/JIOT.2016.2579198>
10. Satyanarayanan, M. (2017). The emergence of edge computing. *Computer*, 50(1), 30–39. <https://doi.org/10.1109/MC.2017.9>
11. Chen, J., & Ran, X. (2019). Deep learning with edge computing: A review. *Proceedings of the IEEE*, 107(8), 1655–1674. <https://doi.org/10.1109/JPROC.2019.2921977>
12. Zhou, Z., Chen, X., Li, E., Zeng, L., Luo, K., & Zhang, J. (2019). Edge intelligence: Paving the last mile of artificial intelligence with edge computing. *Proceedings of the IEEE*, 107(8), 1738–1762. <https://doi.org/10.1109/JPROC.2019.2918951>
13. Wu, Q., Chen, X., Zhou, Z., & Zhang, J. (2021). A survey of recent advances in edge-computing-powered artificial intelligence of things. *IEEE Internet of Things Journal*, 8(18), 13849–13875. <https://doi.org/10.1109/JIOT.2021.3088875>
14. Seng, K. P., Ang, L.-M., & Ngharamike, E. (2022). Pervasive AI for IoT applications: A survey on resource-efficient distributed artificial intelligence. *IEEE Communications Surveys & Tutorials*, 24(4), 2366–2418. <https://doi.org/10.1109/COMST.2022.3200740>
15. Lu, Y. (2017). Industry 4.0: A survey on technologies, applications and open research issues. *Journal of Industrial Information Integration*, 6, 1–10. <https://doi.org/10.1016/j.jii.2017.04.005>
16. Monostori, L. (2014). Cyber-physical production systems: Roots, expectations and R&D challenges. *Procedia CIRP*, 17, 9–13. <https://doi.org/10.1016/j.procir.2014.03.115>
17. Sisinni, E., Saifullah, A., Han, S., Jennehag, U., & Gidlund, M. (2018). Industrial Internet of Things: Challenges, opportunities, and directions. *IEEE Transactions on Industrial Informatics*, 14(11), 4724–4734. <https://doi.org/10.1109/TII.2018.2852491>
18. Cheng, J., Chen, W., Tao, F., & Lin, C.-L. (2018). Industrial IoT in 5G environment towards smart manufacturing. *Journal of Industrial Information Integration*, 10, 10–19. <https://doi.org/10.1016/j.jii.2018.04.001>
19. Internet of Intelligent Things: A convergence of embedded systems, edge computing and machine learning. (2024). *Internet of Things*, 26, Article 101153. <https://doi.org/10.1016/j.iot.2024.101153>
20. Baliyar Singh, R. K. H., & Reddy, K. H. K. (2025). Edge-AI empowered cyber-physical systems: A comprehensive review on performance analysis. *Computer Science Review*, 58, Article 100769. <https://doi.org/10.1016/j.cosrev.2025.100769>