ISSN: 2093-5374 / E-ISSN: 2093-5382

Performance Analysis of Edge AI Architectures for Real-Time Decision Making in Industrial IoT Environments

Jaswanth Kumar Mandapatti[1*]

[1*]Advent Health, Florida, United States. jash.209@gmail.com,
https://orcid.org/0009-0007-5610-6487

Received: February 09, 2026; Revised: March 16, 2026; Accepted: May 04, 2026; Published: June 30, 2026

Abstract

The integration of Edge Artificial Intelligence (Edge AI) with the Industrial Internet of Things (IIoT) is enabling real-time decision-making in smart industrial environments by reducing latency, bandwidth usage, and reliance on centralized cloud systems. Nevertheless, there is still a challenge of choosing an ideal Edge AI architecture, because of performance, energy efficiency and scalability trade-offs. In this paper, the complete performance analysis of edge-native, hybrid, and cloud-assisted artificial intelligence systems is given regarding the performance of such systems in the industrial domain in predictive maintenance, anomaly detection, and visual inspection. An experimental heterogeneous environment was created with the help of various edge devices, such as GPU-enabled and TPU-based systems, and lightweight deep learning models, such as MobileNet, YOLOv5n, and LSTM. Industrial sensor, visual, and acoustic data benchmark datasets have been used. The main performance indicators such as inference latency, energy usage, and model accuracy and network bandwidth were tested in different work load and environmental conditions. It has been experimentally demonstrated that edge-native architecture can offer ultra-low latency (720 ms) which is much lower than cloud-based systems (100-300 ms). The platforms that were energy efficient showed consumption as low as 0.3 J/inference and had model accuracy of up to 95.6 in visual inspection conditions. Also, the predictive maintenance use cases reported up to 81 percent in equipment downtime. The hybrid architectures enhanced accuracy by trading off edge and cloud intelligence but came with the moderate latency overhead (~80 ms). The results point at the fact that latency-sensitive applications are best served with edge-native deployments whereas hybrid solutions offer a tradeoff between performance and scalability. This study offers practical design insights for deploying efficient and reliable Edge AI systems in industrial environments and outlines future directions in federated learning and adaptive edge intelligence.

Keywords: Edge AI, Industrial Internet of Things (IIoT), Real-Time Decision-Making, Edge Computing Architectures, Predictive Maintenance, Performance Evaluation.

1 Introduction

Contemporary industrial ecosystems are becoming more and more converged between the Industrial Internet of Things (IIoT) and the progress in the field of artificial intelligence (AI) (Xu et al., 2021). IIoT allows the incorporation of smart sensors, actuators, and computer structures into the industrial applications, automating and allowing real-time analytics in areas like manufacturing, energy, logistics, and utilities (Chen et al., 2021). These systems facilitate monitoring, diagnostics and predictive maintenance; hence, improvement of operational efficiency and reliability. Edge AI also supports this

Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA), volume: 17, number: 2 (June - 2026), pp. 153-181. DOI: *10.58346/JOWUA.2026.I2.009*

*Corresponding author: Advent Health, Florida, United States.

paradigm further through decentralizing intelligence, and further, allows work to be done nearer to the source i.e. to be done in edge devices like microcontrollers, embedded systems, and industrial gateways (Arumugam et al., 2024; Kong et al., 2022). In contrast to conventional cloud-centric solutions, Edge AI enables real-time inference without the use of remote data centers, and therefore, latency is minimized as well as autonomy of the system is enhanced. This decentralized intelligence move is essential to fulfilling the needs of low-latency and high-reliability and scalable industrial applications, especially in smart manufacturing and digital twins.

Although cloud computing has scalability benefits, centralised IIoT systems have severe drawbacks, such as latency, bandwidth overload, and reliance on reliable network connectivity (Shafique et al., 2021). Transmission of data to the cloud servers leads to delays (around 100-300 ms), which are not tolerated in time sensitive industrial processes like fault detection and safety monitoring. Moreover, the cloud-based systems are susceptible to disruptions and outages of the network, particularly in the remote or low-connectivity settings (Stadnicka et al., 2022). In addition, constant data transmission will cause consumption of energy and increased cost of operations as well as creating issues of data privacy and security. Delicate industrial information sent through the shared networks is vulnerable to breach and centralized architectures are inappropriate in the critical applications. The restrictions require a transition to the decentralized forms of computing, in which the AI processing is brought closer to the source of data.

The edge system identifies shortcomings from the cloud-based models and highlights them in systematically in table 1 as the edges-based alternatives.

Table 1: Comparison of cloud vs Edge IIoT architectures

Criteria	Cloud-Based IIoT	Edge-Based IIoT
Latency	High (100–300 ms)	Low (1–20 ms)
Bandwidth Usage	High (constant data transfer)	Low (local processing)
Data Privacy	Moderate (centralized processing)	High (local decision-making)
Energy Efficiency	Low (continuous connectivity)	High (task-specific optimization)
Scalability	High (elastic cloud resources)	Moderate (hardware-specific limits)
Real-Time Responsiveness	Moderate (dependent on round-trip delays)	High (on-device inference)
Infrastructure Dependency	High (requires robust cloud backend)	Low (minimal external dependency)

This table highlights the divergences present between the cloud and edge implementations with regards to latency, consumption of bandwidth, energy, and real time efficiency. The IIoT in the Cloud suffers from greater lag, which is often anywhere between 100 - 300 milliseconds due to the distance communication and backend processing. Alternatively, the edge-based systems demonstrate latency as low as 1 to 20 milliseconds, permitting substantial benefits for time critical decision-making. Also, edge AI systems require less bandwidth, improve data privacy by performing local inference, and minimal backend support infrastructure.

Being responsive, data privacy, and independent, edge computing is the optimal choice in the case of the IIoT application. However, edge deployments are associated with their fair share of problems. Often face the lack of energy, memory, and processing power and need AI models and sophisticated scheduling methods that are highly optimized to provide real-time outcomes. Industrial settings are required to make

real-time decisions to promote safety, efficiency, and reliability. Lawsuits According to, slow response may cause losses in production, failures in equipment and safety hazards. As an example, predictive maintenance needs the analysis of vibration and sensor data in real-time to avoid machine failure, whereas computer vision systems need to detect defects in a matter of milliseconds on line production of high speed (De La Fuente et al., 2024). Edge AI can answer these needs and provide real-time inference with localization and no reliance on cloud infrastructure (Xu et al., 2024). This enables systems to self-monitor, self-analyze, and self-react to dynamic conditions thus enhancing the efficiency of operations and minimizing downtime. In addition, edge-based processing improves the resilience of the system because it is possible to make decisions in case of the lack of network connectivity (Zhu et al., 2024). The capabilities are especially essential in remote industrial processes like the mining sites, offshore platforms and energy grids.

Real life industrial implementations of edge AI and their corresponding data type, latency, and intelligence on the device requirements are presented in table 2.

Table 2: Industrial applications and edge AI requirements

Industrial Application	Data Type	Latency Requirement	Edge AI Capability
Predictive Maintenance	Vibration, temperature, acoustic	< 50 ms	Time-series forecasting, anomaly prediction
Autonomous Quality Control	Visual, sensor, dimensional data	< 100 ms	Computer vision, object detection
Worker Safety Monitoring	Video, thermal, location	< 20 ms	Human detection, thermal AI
Real-Time Anomaly Detection	Sensor fusion data streams	< 30 ms	Stream-based machine learning
Energy Management Systems	Power usage, equipment state	< 100 ms	Adaptive control and load balancing

Here, five common industrial use case are illustrated in the table: predictive maintenance, autonomous quality control, worker safety monitoring, real-time anomaly detection, and energy management. Each application working with different data types such as vibration signals and thermal images is required to respond within an average of a hundred milliseconds or less. For example, worker safety systems must respond in less than 20 milliseconds to falls and hazards that require real-time monitoring. Some of the Edge AI features are adapted to such conditions, such as predictive maintenance time-series forecasting, object detection in visual inspection, and adaptive load balancing in energy systems. The capability of edge AI technologies to meet such performance criteria makes it impossible to substitute it in the existing industrial setting.

Processing and acting at the point of origin reduces exponentially the amount of data that must be transported over the network which conserves bandwidth, reduces operation costs and reduces logistic costs. Besides this, the edge has enhanced real-time decision-making which has increased resiliency since the systems are able to make decisions and act even when not connected to the cloud. This is vital in remote industrial installations like offshore oil rigs, mining sites, and energy grids where connectivity is not stable or reliable.

Contributions

The paper will provide in-depth performance analysis of Edge AI architectures in real-time decision-making on IIoTs. The most important contributions of the work are the following:

- Comparison: A comparative study of edge-native, hybrid, and cloud-assisted architecture based on the latency, energy consumption, accuracy, and bandwidth efficiency.
- Experimental Framework: The experimental framework is the design of a multi-layered benchmarking framework with heterogeneous edge devices (i.e. NVIDIA Jetson, Raspberry Pi, Google Coral) and real-world industry data.
- Performance Evaluation: Quantitative evaluation of AI models (MobileNet, YOLO, LSTM) working in different workload, environmental and network conditions.
- Design Insights: Detection of architectural trade-offs and useful principles of implementing effective and scalable Edge AI systems in factories.
- Future Directions: The discussion of the emerging technologies (federated learning and TinyML) to improve flexibility and efficiency in edge environments.

The rest of the paper will be organized in the following way: Section 2 will be a review of related literature on Edge AI and IIoT architecture. Section 3 introduces the suggested architectural framework and system design of the Edge AI. The methodology, datasets, and experimental set up are described in section 4. Section 5 provides the performance measures and assessment standards. Section 6 views on the results and the comparison of the results of the experiment. In section 7, real-world IIoT application cases are given. Findings and trade-offs in Section 8 and a conclusion of the paper are discussed in Section 9.

2 Literature Review

The intersection of edge computing and artificial intelligence has greatly improved the features of the Industrial Internet of Things (IIoT) systems, as able to operate in real-time, automate, and make intelligent decisions (Situnayake & Plunkett, 2023). Conventional IIoT designs were largely based on the concept of centralized cloud computing, which implied that data obtained in distributed devices were sent to remote servers so that could be analysed there. Although scalable, this model presents serious constraints including high latency, bandwidth overhead, energy consumption and security weaknesses. According to recent works (20212024) that point to the development of Edge AI as a paradigm shift that brings the computation closer to the source of data (Sodhro et al., 2019; Sun & Deng, 2024). Local inference of edge devices by these systems helps to reduce the latency, reduce network congestion, and provide faster response times, which are required in industrial applications like predictive maintenance, quality inspection, and robotic control (Li & Li, 2023). To overcome hardware limitations, scholars have tried to design lightweight and efficient AI models including MobileNet, TinyYOLO and quantized LSTM networks, which can provide high accuracy with lower computational cost. These innovations have allowed closed-loop control systems that are nearly real-time responsive and edge intelligence is especially useful in industries that are latency sensitive like oil and gas, automotive, and manufacturing (Wan et al., 2020). The current literature highlights the importance of Edge AI to the compliance with the regulations, especially in the areas where data privacy and residency are decisive. Local data processing will help organizations decrease their dependence on cloud infrastructure without violating data protection rules.

In recent times, various researchers planned, optimized, and deployed edge AI to industrial applications with the aim of integrating edge AI to industrial applications. These findings demonstrate the design of edge computing to IIOT systems. Table 3 gives a comparative review of five major studies which depict varying edge AI architectural designs and their findings.

Table 3: Comparative summary of key related works on edge AI in IIoT

Author(s) & Year	Focus Area	Architecture Type	Key Findings
Mohy-Eddine et al., (2023)	Edge-based fault detection in IIoT	Edge-native	Reduced data transfer by 60%, real-time alerts in 50ms
Boiko et al., (2024)	Distributed edge-cloud hybrid inference system	Hybrid (Edge + Cloud)	Improved accuracy by 12% using hybrid intelligence
Wang et al., (2021)	Low-latency edge AI for predictive maintenance	Edge-native	Achieved 15ms inference on vibration data
Kumar et al., (2023)	Federated learning for smart factories	Federated Edge	Reduced model drift and preserved privacy
Chen, (2023)	Scalable edge architecture for sensor fusion	Hierarchical Edge	Supported 500+ nodes with robust inference stability

Some of the recent studies have investigated various architectural methods of implementing AI in IIoT settings such as edge-native, hybrid, federated, and hierarchical models. Table 3 provides the summaries of some of the important works.

The study by Mohy-Eddine and colleagues, (2023) (Mohy-Eddine et al., 2023) showed that edge-native architectures can be used effectively to identify faults, with a 60 per cent lowering in the volume of data transfer and response time of less than 50 ms. Likewise, (Wang et al., 2021) introduced an edge-based predictive maintenance system with low latency, with inference times as low as 15 ms, indicating the compatibility of an edge-native system with time-sensitive applications.

Conversely, to find a compromise between performance and scalability, hybrid architecture has been considered. A distributed edge-cloud framework suggested by (Boiko et al., 2024) gained 12 percent accuracy in the system by using edge and cloud layers in collaborative intelligence. Along this line, more recent works, like that of (Kumar et al., 2023), proposed federated learning models that can provide better privacy and minimize model drift because can train models decent rally on edge nodes. The authors also suggested hierarchical edge architecture that can be extended to more than 500 nodes and provide a better allocation of resources and stability in the system (Chen, 2023). Together, these works show that edge-native structures are better in terms of latency and responsiveness, whereas hybrid and federated structures are better with respect to scalability and model generalization. Literatures are mostly implementation-oriented and do not provide standardized evaluation systems to compare cross-architectures. Even though the implementation of Edge AI systems in the industrial setting has progressed greatly, there are still some difficulties in this area. The lack of computational capacity of the edge devices, such as memory, processing power, and energy availability is one of the main constraints (Lee et al., 2022). These limitations require model trade-offs between the complexity and accuracy of inference. The other important problem is the absence of interoperability and standardization of heterogeneous industrial systems. Current solutions tend to be based on vendor specific hardware and communication standards and therefore cannot be easily integrated and scaled (Kuchuk & Malokhvii, 2024). Also, the variability of latency (jitter) as well as hardware contention and

network instability can impact real-time performance, and affect the reliability of the system in safety-critical applications.

Another challenge that has not been resolved is model lifecycle management. In contrast to cloud environments, there are no powerful model deployment, version control, and update mechanisms available in edge systems, and managing large-scale processes is complicated. In addition, issues like sensor noise, incomplete datasets as well as limited labelled data are also data-related issues that affect model performance and generalization to a large extent. Based on the reviewed literature, the discussed Edge AI architectures edge-native, hybrid, and federated have different benefits provided in relation to the situation of application. But the available literature concentrates on individual applications or particular hardware platforms, and thus cannot be generalized to other applications. The evaluation measures used in the various studies are inconsistent with some of the measures based on latency whereas others were based on accuracy or power consumption. The absence of a standardized benchmarking would mean that the researchers and practitioners would find it hard to determine the most appropriate architecture to use in a specific industrial application. It is evident that a unified, comparative analysis framework examining several Edge AI architectures in the same conditions with standardized performance metrics is needed. To fill this gap, the current paper will suggest a single benchmarking methodology, which compares edge-native, hybrid, and cloud-assisted structures under the main metrics of latency, power usage and consumption, precision, or bandwidth economy. This research will contribute to the design of efficient and scalable Edge AI systems in IIoT settings with actionable information by implementing a variety of datasets, hardware platforms, and workload conditions.

3 Edge AI Architectural Framework

3.1 Overall System Design and Functional Layers

The suggested Edge AI architecture of an Industrial IoT (IIoT) setting is aimed at serving real-time processing and multi-source data stream and intelligent decision-making by means of a structured, multi-layered structure. This architecture does not merely include physical computing and networking elements, but also includes functional layers that are in charge of data acquisition, communication, inference, orchestration and application-level control. The system is structured in five major layers, perception, network, edge intelligence, orchestration and application which all play a role in ensuring scalable and responsive industrial ecosystem. The perception layer is made up of sensors and actuators that touch directly with the physical world to produce raw data in the form of a vibration signal, temperature sensor, acoustic signal and video signal, which is then transduced into digital form to do further processing. The network layer enables the transmission of data reliably between devices, edge nodes, and gateways with the assistance of such protocols as MQTT, CoAP, OPC UA and the communication technologies 5G and LoRaWAN, providing low-latency and efficient data transfer. The heart of the architecture is the edge intelligence layer, and embedded systems, microcontrollers, and edge devices run real-time AI inference on optimized frameworks such as TensorFlow Lite, ONNX Runtime, and NVIDIA TensorRT, which allow developers to use it in applications like anomaly detection, predictive maintenance, computer vision, sensor fusion, and so on, without depending on cloud infrastructure. On top of this, there exists the orchestration layer, which handles system-wide coordination, such as task scheduling, load balancing, model deployment, and updates, and the platforms available to support this coordination include Azureiot Edge or AWS Greengrass and the resilience of these systems with Kubernetes based systems such as Kube Edge. Lastly, the application layer provides actionable insights in the form of dashboards, APIs, and automated control systems to provide real-time

monitoring and control through programmable logic controllers (PLCs) or other more advanced industrial operations like maintenance alerts and adaptive system reconfiguration. The net effect of this layered architecture is an architecture that is modular, scalable, and possesses low-latency so that it can support the demanding nature of modern industrial environments.

3.2 Edge vs Fog vs Cloud Placement Strategies

An AI-supported IIoT system architecture can change entirely depending on where the processing occurs. Strategy sets the place of execution and deployment of AI models, which are Edge, Fog, and Cloud. All of them have their own characteristics of advantages and drawbacks in terms of latency, energy, bandwidth, and scalability. Calculating the data at the point of creation is also called edge computing. This is often accomplished through microcontrollers, industrial gateways, or embedded GPUs. This method is highly effective in tasks that require milliseconds of response time like robotic movements and emergency power cut-offs. The edge deployment also helps in saving bandwidth by lowering the amount of data that needs to be sent upstream. But this comes with the trade-off of low memory and computational power, which would need the model to be simplified or compressed. Fog computing is the layer is frequently referred to as Fog which is positioned between cloud and edge and where data analytics such as data balancing out, processing, and primary data preprocessing happens. Fog nodes are incorporated within mobile base stations, or in local data centers. This offers great value in remote industrial sites like logistics centers, factories, or energy plants. These locations use edge devices to send large sets of data to the fog. Fog nodes use Filters or Multi-Input Single Output (MISO) chmods to balance out the load. These nodes use a limited amount of cloud fog computing in order to increase their processing capability. Cloud computing is typically used for coordinating entire fleets, and refining models. With unrivaled computational scalability, cloud computing is not without its downsides, including high latency and dependence on consistent network. This model of computing is extremely useful for non-urgent tasks like making strategic decisions, training an AI model with data collected from numerous factories etc.

These placement strategies are presented in the following table, which outlines their differences and highlights the most appropriate reasons for their use in multiple industries (Table 4).

Table 4: Comparison of edge, fog, and cloud placement strategies

Architecture Layer	Processing Location	Typical Use Cases	Latency	Bandwidth Requirement	Energy Consumption	Scalability
Edge	On-device or near sensors (e.g., gateways, industrial nodes)	Real-time anomaly detection, predictive maintenance	1–20 ms	Low	Low to moderate (battery/low-power optimized)	Device-limited
Fog	Local data centers or intermediate network nodes	Data aggregation, pre-processing, localized analytics	20–100 ms	Moderate	Moderate (rack servers)	Region-limited
Cloud	Remote, centralized cloud infrastructure	Long-term trend analysis, model training, global coordination	100–300+ ms	High	High (data center scale)	Global

Most industries opt for a hybrid model that incorporates edge, fog, and cloud layers. This architecture enables the real-time tasks to be done on the source but the cloud services can still store data to be used

in archiving, learning loops and high-level analytics. The choice of strategy is influenced by latency sensitivity of the application, volume of data, reliability of the network, and regulations that need to be obeyed.

3.3 Data Flow, Inference Pipeline, and Control Loops

The IIoT data flow in an Edge AI-enabled system would start at the perception layer with physical sensors producing raw signals in the following forms: time-series data, images, and acoustic waveforms and then sent to edge processing units via low-latency data communication protocols, like MQTT, CAN bus, or OPC UA. When it is received, it is pre-processed using noise cleaning, normalization, synchronizing the timestamps, and reducing dimensionality so that those that follow it can be consistent and efficient. The raw information is then fed into the inference pipeline in which optimized and lightweight models of machine learning that have been previously trained offline and have been compressed with quantization or model distillation are used to carry out classification, regression, or anomaly detection in inference engines such as TensorFlow Lite, OpenVINO, and NVIDIA TensorRT. After inference, the system proceeds to a decision-making phase and the outputs are compared against predefined thresholds or rules; in case met (e.g. abnormal vibration levels) control signals are dispatched to actuators or programmable logic controllers (PLCs) to cause the apparatus to respond in the correct way (e.g. by shutting down machinery or by sending alerts). The control mechanism may be of the open-loop mode or closed-loop mode where the closed-loop mode is more effective in dynamic situations since it is able to continually monitor the system feedback and make adjustments whenever necessary. To support real time industrial applications, the whole pipeline including the data acquisition and the control execution should be within the sub-second limit on the latency, which implies the effective design of firmware, low-overhead inference models, and event-driven system architecture to reduce as much as possible the processing delay and simultaneously assure the stable and responsive operation.

3.4 Architecture for Model Training, Deployment, and Updating

The model training, deployment, and updating processes in Edge AI-enabled IIoT systems are designed in the form of a multi-stage pipeline which is balanced and provides computational efficiency, scalability, and adaptability. Contrary to inference, which is run at the edge, the model training is implemented on the cloud as it has high computational requirements backpropagation and gradient optimization. This stage is based on domain specific datasets which represent various operation conditions, preprocessing, feature extraction, hyper parameter tuning and validation to verify the robustness of the model. After training, models are then optimized to run on edges by pruning, quantizing, clustering weights, and shrinking models by large margins, with acceptable accuracy. Such compressed models are deployed to edge devices through safe means like over-the-air (OTA) updates or local flashing, in edge-compatible formats such as TensorFlow Lite, PyTorch Mobile, or Apache TVM. The deployment model in industrial settings needs to be secure, fault resilient and traceable where the model is successfully tested in sandboxed systems before being deployed to production. Processes lifecycle management tools Virtual process Model lifecycle management tools like MLflow or Kubeflow support version management, performance management, and rollback in case of anomalies. The problem of edge deployments is a model drift, which occurs when the sensor distribution is subject to environmental differing or equipment depreciation. This has been fought using modern architectures that use feedback loop and federated learning, with models being trained locally on edge devices using recent data and with periodic aggregation in a central server to enhance global model performance and improve learning without transmitting raw data, preserving privacy. Also, the containerized deployment approaches with the help

of Docker and Kubernetes (e.g., KubeEdge) allow the automated and scalable updates controlled by the policies with references to the bandwidth, device health, and uptime limits. A distributed model registry can be a centralized, or even a fog-based registry, to maintain a level of synchronization and versioning across the distributed nodes. This integrated architecture, on the whole, allows a continuous learning and adaptation approach and still provides the efficiency, safety and scalability necessary to support real-time industrial Edge AI systems.

4 Methodology and Experimental Setup

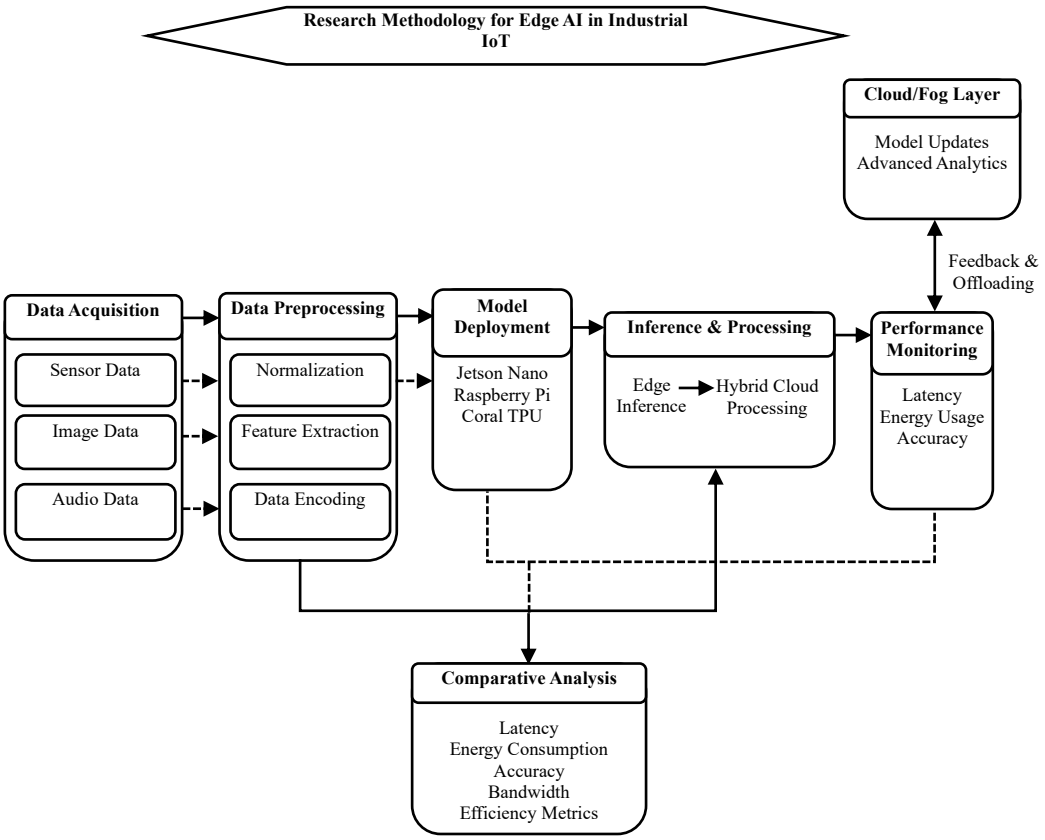

Figure 1: Comprehensive research methodology workflow for edge AI in industrial IoT systems

This is the figure 1 that presents the overall research approach that will be used to assess Edge AI architectures in the Industrial internet of things (IIoT) systems. Workflow starts with the data acquisition where multi-modes of industrial data such as sensor, image and audio data are recorded. This is then followed by data preprocessing which entails the normalization of the data, extraction of features and encoding of the data to facilitate effective inference. The resulting processed data is then deployed on heterogeneous edge computing systems, including Jetson Nano, Raspberry Pi and Coral TPU, which reflect the industrial diversity in terms of hardware. Depending on the requirements of the system, inference and processing stage does real-time decision-making locally at the edge or using hybrid cloud-assisted, when needed. At the same time, a cloud/fog layer enhances the use of advanced analytics and updates on models, which allows constant improvement of the system. The performance monitoring stage captures performance metrics of latency, energy consumption, and accuracy, and thus is quantitatively evaluated. Lastly, the comparison between the results is compiled in the comparative analysis module where various architectures are compared according to critical indicators such as

latency efficiency, energy optimization, accuracy and bandwidth consumption. Such a pipeline is a structured one, which comprehensively tests Edge AI systems over a variety of industrial conditions, in a scaled and realistic way.

4.1 Hardware and Deployment Platforms (e.g., Jetson Nano, Raspberry Pi, Coral TPU)

To guarantee an all-encompassing and impartial assessment of Edge AI frameworks in Industrial IoT (IIoT) settings with an extensive range of computational potentials, memory designs, energy usage patterns, and AI framework compatibility, a heterogeneous collection of edge computing platforms was chosen. Its Maxwell 128-core GPU and powerful support of high-performance inference software, including TensorRT and DeepStream, were used on the NVIDIA Jetson Nano to implement real-time execution of deep learning models, including YOLOv5 and MobileNet, at a low power level of less than 10W. Conversely, the Raspberry Pi 4 (4GB RAM) was used as a relative baseline platform to run lightweight inference applications that take advantage of CPU-based processing and used TensorFlow Lite and PyTorch Mobile to run image classification and signal detection applications. Google Coral Dev Board was added to support ultra-low latency and energy-efficient inference, which has an integrated Edge TPU that can provide 4 TOPS, which is extremely appropriate when quantized models require and when industrial applications need battery packs. In order to obtain more compute-intensive conditions, the Intel NUC (i5 model) was chosen, which provides the desktop level of processing capabilities when combining with the built-in Intel UHD Graphics and OpenVINO optimization of accelerated ONNX model execution. The BeagleBone AI platform was also added, indicating the industrial-grade embedded systems, which were supported with the TI Deep Learning Library and could be utilized in the specialized industrial control equipment. Together, all these platforms can be used to perform realistic benchmarking of a wide range of edge deployment environments covering both low-power embedded systems to high-performance edge gateways.

These devices were tested in both standalone and clustered configurations to simulate different industrial use cases like factory floor control systems, edge data gateways, and mobile robot controllers. The devices specifications, like thermal design power, memory configuration and compatible frameworks are outlined in the table 5 below.

Table 5: Hardware benchmark overview for edge AI inference

Edge Device	CPU/GPU Specs	RAM	Thermal Design Power (TDP)	Optimized AI Frameworks
NVIDIA Jetson Nano	Quad-core ARM Cortex-A57, 128-core Maxwell GPU	4 GB LPDDR4	5–10W	TensorRT, DeepStream
Raspberry Pi 4 (4GB)	Quad-core ARM Cortex-A72, Broadcom VideoCore VI	4 GB LPDDR4	3–6W	TensorFlow Lite, PyTorch Mobile
Google Coral Dev Board	Quad-core Cortex-A53, Edge TPU (4 TOPS)	1 GB LPDDR4	2–4W	Edge TPU Compiler, TensorFlow Lite
Intel NUC (i5)	Intel Core i5, Intel UHD Graphics 620	8 GB DDR4	15–25W	OpenVINO, ONNX Runtime
BeagleBone AI	Dual-core ARM Cortex-A15, Embedded PowerVR SGX544	1 GB DDR3	3–5W	TI Deep Learning Library

The implementation of this hardware suite facilitated the testing of real-time AI workloads in the smart manufacturing, process control, logistics, and energy monitoring environments during comparative testing with various capabilities and constraints.

4.2 Datasets Used (e.g., MIMII Dataset, Edge-IIoTset)

Data choice is an important aspect toward achieving validity and reliability of benchmarking Edge AI solutions in Industrial IoT (IIoT) systems. In order to obtain realism and diversity, both publicly released and synthetic data were used, and major industrial situations used, which include sensor fusion, anomaly detection, visual inspection and machine condition monitoring. MIMII (Malfunctioning Industrial Machine Investigation and Inspection) dataset, created by Toshiba was used to detect acoustic anomaly and it consists of audio samples of industrial parts e.g. valves, pumps, fans, slide rails under normal and fault conditions. The audio signal sampled at 16 kHz was converted to Mel spectrograms and classified with CNNs. As well, the Edge-IIoTset data was chosen due to its large-scale and realistic simulated smart factory, which consists of more than 600,000 records of sensor measurements of temperature, pressure, humidity, energy consumption, and motion and injected attack scenarios to assess the strength of the models in the context of cybersecurity awareness. In computer vision, the NEU surface defect dataset was utilized which comprised of 1,800 grayscale images of six types of steel surface defects, which allowed assessing image classification and object detection models to be used in automated quality inspection. In order to expand experimental coverage, synthetic data was used to simulate conveyor belt object detection and robot control cues to enable real time latency and control loop validation. Each dataset was pre-processed in the form of normalization, feature extraction, and encoding to TFRecord and ONNX formats to make them compatible with edge devices. All this variety of data selection makes it possible to thoroughly test a variety of IIoT applications reflected in the real-world state of the industry, noise differences, and edge-related computational issues.

The experimental analysis was based on several benchmark datasets to make it diverse and realistic:

- MIMII Dataset: 16 kHz sampling Audio database (approximately 10,000 samples) of machine anomaly detection.
- Edge- IIoTset: Huge amount of sensor data (>600,000 records) temperature, pressure, humidity and energy consumption data.
- NEU Surface Defect Dataset: Image-based dataset (1,800 grayscale images) based on the classification of defects into six categories.
- Synthetic Dataset: Real time control signal generation and conveyor belt object detection scenarios were generated.

All datasets were pre-processed by means of normalization, feature extraction (e.g., Mel spectrograms) and transformed to edge compatible formats (e.g., TFRecord and ONNX).

4.3 AI Models and Benchmarking Tools (e.g., MobileNet, YOLO, TensorRT)

An array of lightweight and high-performance AI models, together with benchmarking tools, industry standard was chosen to achieve the best-balance of inference speed, memory efficiency, and accuracy in terms of Edge AI deployment. Convolutional Neural Networks (CNNs) were used (MobileNetV2 and MobileNetV3) because compact in structure and effective at detecting defects and classifying materials due to their high accuracy, which is suitable in edge-based vision, such as detecting defects and classifying materials. These models were optimized further with the help of quantization and turned into TensorFlow Lite to be used effectively on resource-limited machines (Raspberry Pi, Coral TPU). To detect objects with real time, YOLOv5n (nano) and YOLOv4-tiny were used, reaching more than 30 FPS with moderate use of the GPU when running on NVIDIA DeepStream and Intel OpenVINO pipelines on Jetson Nano and Intel NUC. Furthermore, a 1D CNN model which was very lightweight was designed to perform acoustic anomaly detection based on Mel spectrogram features of the MIMII

dataset, whilst both GRU and LSTM networks were implemented to perform sequential anomaly prediction of sensor data in the Edge-IIoTset. Pruning and quantization-aware training optimized these models and led to a model size reduction of about 60 percent and little accuracy loss. The standard measures of model performance, including accuracy, precision, recall, and F1-score were used to gauge the system-level measures of latency and energy consumption were measured using time-stamped logs, inline power meters, and measurement tools such as PowerTOP. MLPerf Tiny, TensorRT Profiler, Edge TPU Benchmark, and OpenVINO Model Optimizer were the frameworks used to conduct benchmarking. To simulate the real-world edge conditions, all the inference tasks were performed with a batch of one and every experiment was repeated 50 times so as to provide the statistically reliable results and ensure minimum variance in the evaluation of performance.

Conceptual Framework

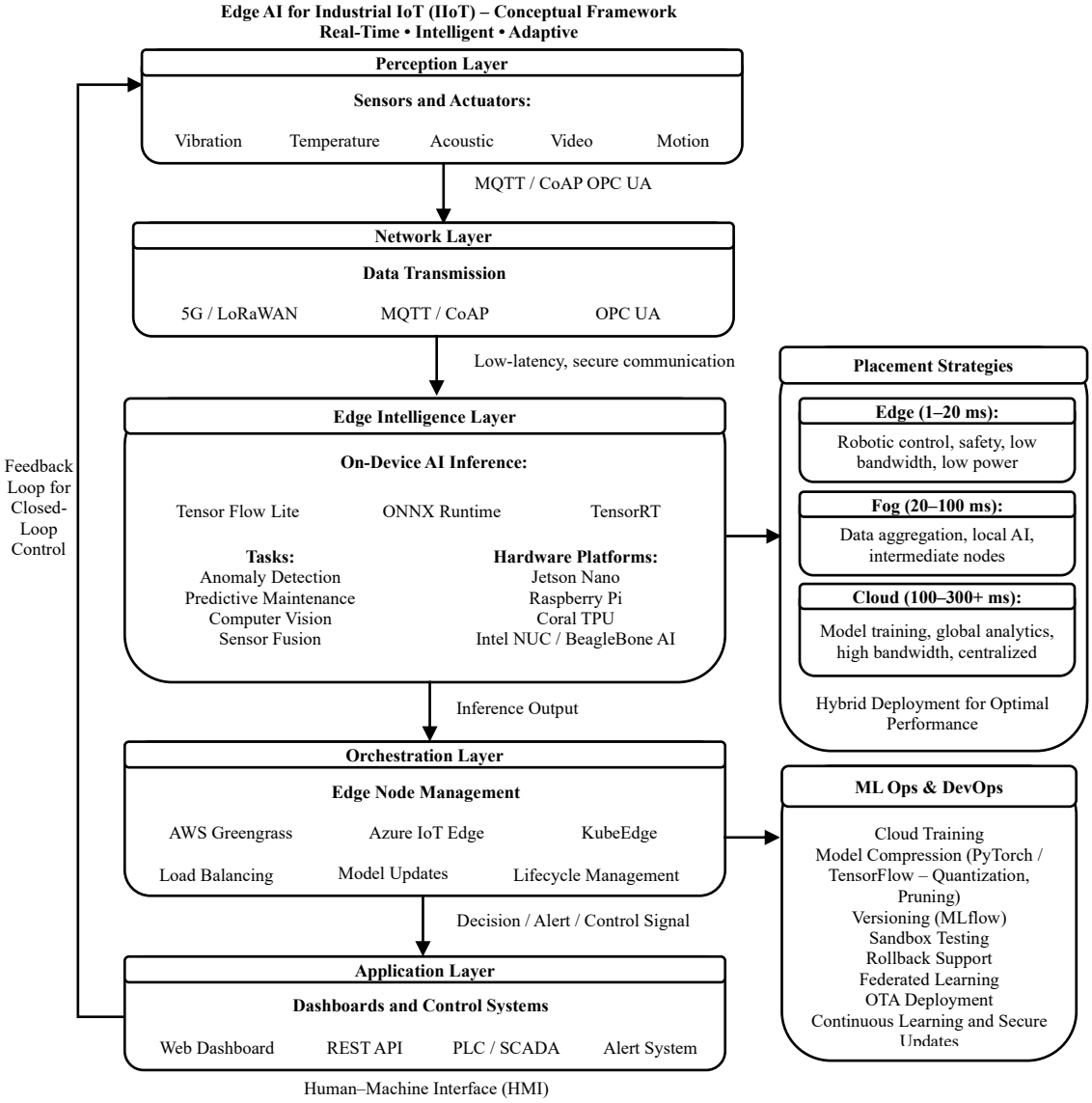

Figure 2: Conceptual framework for edge AI in industrial IoT systems

The figure 2 shows a multi-layered conceptual diagram of how to apply the concept of Edge AI in Industrial IoT (IIoT) settings, which focuses on real-time, distributed, and adaptive decision-making. This framework combines five fundamental functional layers perception, network, edge intelligence, orchestration and application in a data flow and feedback process. Data transfer in perception layer is first gathered through sensors and actuators and sent to edge devices via low-latency communication protocols in the network layer and real-time AI inference is done on optimized models. The orchestration layer coordinates activities across the system including but not limited to load balancing and model updates and fault tolerance, and the application layer provides actionable insights in the form of dashboards, APIs, and other automated control systems. The framework also takes into consideration hybrid placement strategies (edge, fog, and cloud) to trade off latency, scalability, and computational performance and MLOps-based model lifecycle management to ensure continuous learning and system optimization. All in all, this architecture gives a scalable and robust base to implement intelligent, efficient, and responsive Edge AI systems in the contemporary industrial setting.

4.4 Experiment Design for Load and Environment Variability

To test the resilience and practicality of Edge AI deployments, an experimental framework was developed in full by varying the workload intensity as well as the environment. Several load profiles were specified, such as a normal load case with input rates of 5-10 Hz corresponding to normal sensor or video processing, a high current case where both input rates were doubled in order to test the system under the load especially in buffering, inference queue control and thermal throttling. The testing was performed in a controlled environment by variable temperature and power supply conditions to test the devices at the temperature of 25C, 40C and 55C in order to simulate the common to extreme industrial environments. Further, the power stability was simulated through bringing the supply to 70 per cent, a situation indicating the occurrence of brownout and measuring the system stability. Artificial latency (2020 ms), jitter, and packet loss to further mimic the behavior of a real-world network were also added with network emulators to assess hybrid edge-cloud architectures in conditions of unstable connectivity and to understand the ability of edge devices to continue local inference in case of disruptions. There were 12 unique test conditions in each edge platform, including the variation of datasets, models, load conditions, temperature, power, and network quality. The major key performance indicators such as inference latency (ms), energy consumption (Joules per inference), throughput (frames per second or inferences per second), as well as model accuracy were all recorded systematically. The findings were plotted in the form of comparative charts to establish performance trade-offs and also determine the best hardware-model combinations to use in each industrial deployment scenario.

Algorithm and Mathematical Formulation of the Proposed Methodology

Algorithm 1: Edge AI Real-Time Inference and Decision-Making Pipeline

The proposed system follows a structured algorithm for real-time data processing, inference, and control execution at the edge:

Input: Data stream D from sensors

Output: Decision/action R

1: Initialize edge device E with trained model M

2: while system is active do

3: Acquire input sample d_i from D

4: Preprocess d_i (normalization, filtering, feature extraction)

5: Perform inference:

$y_i = M(d_i)$

6: Compute confidence score C_i

7: if $C_i \geq$ threshold θ then

8: Generate decision/action R_i

9: Execute control command via actuator/PLC

10: else

11: Forward d_i to cloud/fog layer for further analysis

12: end if

13: Log performance metrics (latency, energy, accuracy)

14: end while

15: Return R

This algorithm provides an efficient real-time processing by focusing on making local inferences and selectively offloading doubtful predictions to upper-level computing layers.

The performance and optimization of the proposed Edge AI system are formulated as follows.

The general goal of the system is to reduce latency and energy and maximize the accuracy of the decision which can be formulated as:

$$min(L, E) subject\ to max(A) \qquad (1)$$

where:

L = inference latency (time delay between input and output)

E = energy consumption per inference

A = model accuracy

The system intends to optimize a trade-off between predictive performance and computational efficiency as presented in equation (1). In practice in practice, lowering latency and energy can impose a constraint on model complexity, and so a variety of methods are used to meet this multi-objective optimization, including: quantization, pruning, and efficient model architectures. This formulation gives a mathematical basis of assessing and comparing various Edge AI architectures in different conditions in industries.

5 Performance Metrics and Evaluation Criteria

In real-time decisions in the Industrial Internet of Things (IIoT) systems, the Edge AI frameworks should be examined in greater detail along a wide range of performance perspectives in order to ensure that adhere to the requirements of operations and business. An independent industrial system must be in a position to interpret sensor data, make smart decisions and initiate action within strict time and resource constraints. This subsection outlines the key measures in the context of the performance analysis of edge-based AI structures and comments on the findings that were achieved after a comprehensive experimentation process. These measures are divided into four big categories where the first one

incorporates the latency and inference time, the second one involves the energy and resource consumption, and the two final measures are the accuracy of the decision and model confidence and data transfer efficiency in terms of bandwidth.

5.1 Latency and Inference Time

Latency is an important performance measure of Edge AI-based industrial systems and determines the time and time difference between the reception of input data and the generation of the corresponding action and is directly related to operational safety and responsiveness. In time sensitive processes like robotic control and predictive maintenance, any small faults in detecting the fault can result in the destruction of equipment, safety and loss of production, so low and consistent inference time is highly required. In order to measure latency, lightweight deep learning architecture models such as MobileNetV2, YOLOv5n and LSTM were executed on five heterogeneous edge devices with inference times of each model and configuration measured with a batch size of one repeated 50 times to recreate the reality of a real-time situation. The Google Coral Dev Board has the lowest average inference latency of about 7 ms (As shown in figure 3), which is a result of its internal Edge TPU accelerator, and was followed by Intel NUC with about 12 ms because it has a faster CPU and is supported by the OpenVINO optimizer. The NVIDIA Jetson Nano was intermediately performing at 18ms with the embedded GPU and the Raspberry Pi 4 and BeagleBone AI were intermediately performing at 34ms and 39ms respectively because do not have dedicated AI accelerators. To get sub-20 ms response time, which is needed in industrial applications (in real-time anomaly detection and automated control, etc.), only the Coral Dev Board and Jetson Nano provided the necessary threshold. The results indicate the significance of hardware-accelerated inference towards low-latency performance and show that device architecture is a key factor in supporting real-time implementation of Edge AI on IIoT devices.

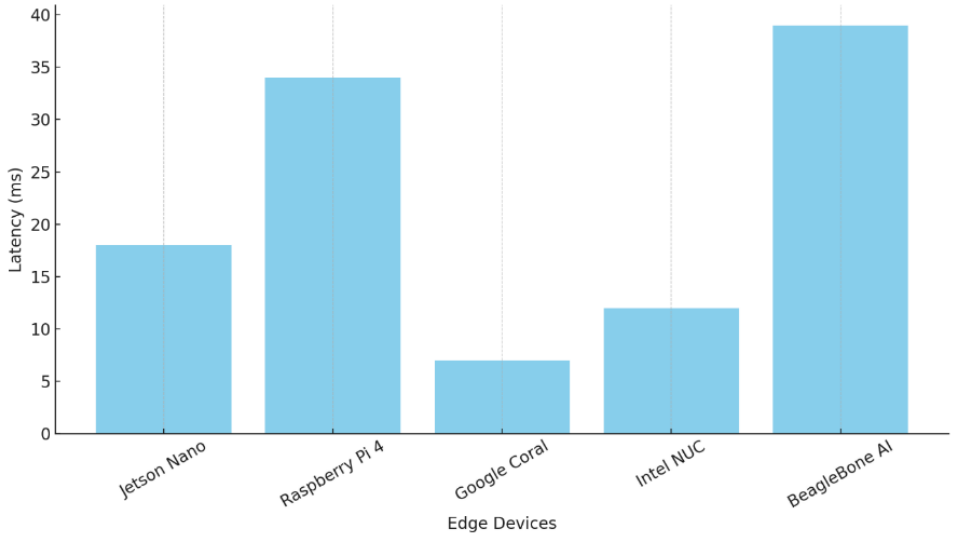

Figure 3: Average inference latency on different edge devices

5.2 Energy and Memory Consumption

Two important points to measure the efficiency of Edge AI systems are energy consumption and memory usage, especially in deploying robots in the mobile industry, wearables, and remote industrial sectors when battery life, thermal limits, and hardware resources should be tightly controlled. In order to evaluate energy consumption, inline power meters, device-specific profiling tools like Jetson Power

Monitor, Coral Metrics and RPi Stats, system tools like htop and nvidia-smi were used to monitor memory usage of GPU-enabled devices. The Google Coral Dev Board was the most efficient in terms of energy consumption (0.3 Joules/inference), due to the specialized Edge TPU, which is a low-power, high-throughput AI inference engine, as shown in figure 4. NVIDIA Jetson Nano next at 0.5 Joules with a trade-off in performance versus energy consumption, compared to 0.9 Joules and 0.8 Joules of Raspberry Pi 4 and BeagleBone AI, respectively, because it used CPU-based inference. In the case of the Intel NUC, the greatest energy draw was 1.2 Joules, which is relatively high, and this renders it undesirable in energy-limited conditions even with its better computing capabilities. Lightweight models like MobileNet and LSTM performed well with 1256GB RAM (under 250MB) and more complicated networks like YOLOv5n needed more than 500MB of memory (recommended a minimum of 4GB RAM). The findings suggest the relevance of model optimization methods such as quantization, pruning, and knowledge distillation to minimize resource requirements in the deployment of AI models to resource-constrained edge devices without performance loss without compromising the accuracy of AI models.

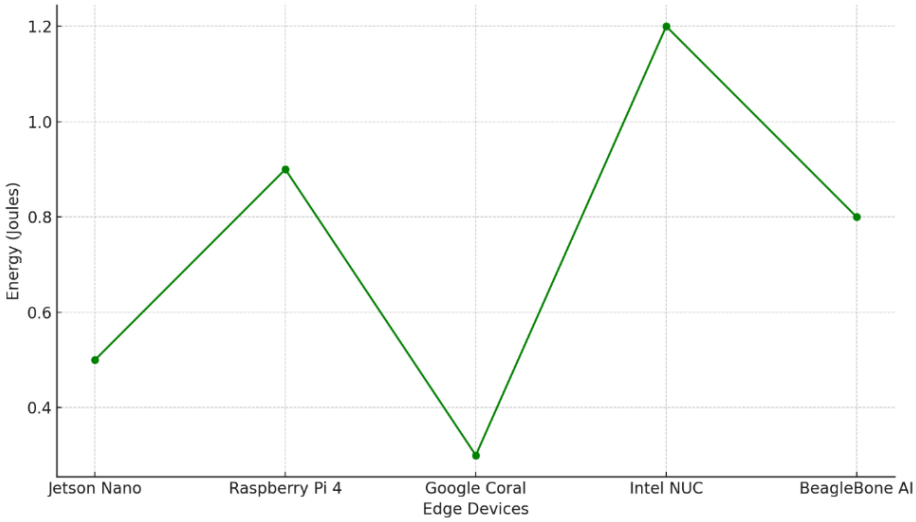

Figure 4: Energy consumption per inference

5.3 Decision Accuracy and Model Confidence

Although such factors as latency and resource efficiency matter, the actual success of an Edge AI system is determined by its accuracy and reliability of decisions since falsehoods in the industrial setting may cause unnoticed flaws, safety incidents, and losses. In order to measure the quality of the model, its accuracy, precision, recall, F1-score were calculated per model-dataset combination as well as model confidence calculated as the average softmax probability of the predicted classes. The findings reveal high performance with all the models, the LSTM-based anomaly detection model in the Edge-IIoTset dataset gives a 91.4% accuracy and a 0.88 F1-score, whereas the CNN-based models in the MIMII dataset gave 93.1% accuracy in the acoustic fault classification. In computer vision applications, MobileNet and YOLOv5n models were found to be better on the NEU surface defects dataset with 95.6 and 94.2 accuracy, respectively, and it is indicated that lightweight models can also achieve high accuracy even when used on the edge devices with limited resources. Confidence-wise, the average confidence of YOLOv5n was the highest with 94.5, which implies that the model is able to achieve consistent and accurate prediction with minimal noise and fluctuations in time-series sensor data compared to the LSTM models which had a lower average confidence of 88.2. Another finding was that

the devices with high input resolutions and frame rates helped to increase the prediction accuracy, which provides evidence of the impact of hardware capabilities on the model. Also, false positives and false negatives were examined and the majority of mistakes were when under high load or thermal throttling and it was therefore observed that the hardware conditions should be steady since this ensures the same inference accuracy in real-time industrial applications.

5.4 Data Transfer and Network Bandwidth Efficiency

One of the benefits of Edge AI in IIoT systems is that it can reduce dramatically data transmission to centralized cloud infrastructure, resulting in much better bandwidth, quicker reaction time, and greater privacy of data. To measure the network usage, the bandwidth usage was observed over a duration of 10 minutes using packet analysis software like Wireshark and iftop in real-time inference under normal and high-load conditions, to capture the telemetry and result-stream transmissions. The lowest bandwidth consumption of the Google Coral Dev Board was 0.4 Mbps as shown in figure 5, due to the capability to give it full on-device inference with minimal cloud-to-contact interaction. The NVIDIA Jetson Nano and the BeagleBone AI had moderate bandwidth consumption of 0.815-1.0 Mbps, which is mainly attributed to the periodical transmission of alerts and metadata. The Raspberry Pi 4 had a little higher usage with about 1.1 Mbps, indicating partial reliance on cloud based logging whilst the Intel NUC had the highest bandwidth usage at about 1.5 Mbps, which is due to continuous uploading of detailed inference logs and high-resolution data to be post processed. These findings clearly show that edge-native architectures significantly lower network overhead and devices such as Coral and Jetson Nano are therefore very fit to be used in bandwidth-limited or remote environments. In addition, less data conveying increases the privacy of the system in that the raw data are not distributed, reduces the cost of communications and is also more resilient to unstable networks and latency variability. These attributes are needed especially in the case of compliance-intensive sectors like healthcare, aerospace, and energy, where the security of data and stable operation are the most important factors.

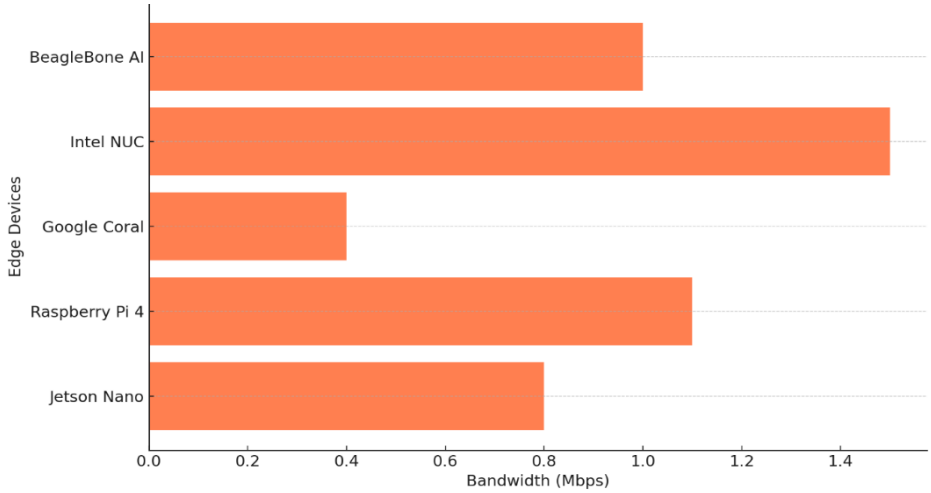

Figure 5: Average network bandwidth utilization

6 Results and Performance Evaluation

This section describes the results of testing the Edge AI architecture, especially its key performance characteristics in diverse hardware, AI models, deployment platforms, and stress operational cases. The complex simulation encompasses predictive maintenance, image quality inspection, and anomaly

detection using some datasets and edge devices. The analysis is narrowed down to four major areas: comparative latency, power efficiency, accuracy, and inference speed estimates under dynamic load conditions as well as architecture deployment trade-off together with real-time reliability statistical analyses.

Software and Tools

The proposed Edge AI framework implementation was performed in the form of an amalgamation of industry-standard tools and frameworks. TensorFlow, PyTorch, and Keras were used to create deep learning models, and TensorFlow Lite, ONNX Runtime, and NVIDIA TensorRT were used to deploy them. The optimizations in terms of edges were done with the toolkit OpenVINO and Edge TPU Compiler. There were performance profiling tools such as MLPerf Tiny, TensorRT Profiler, PowerTOP, and htop. Wireshark and iftop were used to perform the network analysis, whereas the statistical validation was executed by Python (NumPy, SciPy, Matplotlib).

Table 6: Parameter initialization for experimental setup

Parameter	Value / Range
Learning Rate	0.001
Batch Size	1
Number of Epochs	50
Input Resolution	224×224 / 416×416
Confidence Threshold ($\boldsymbol{\theta}$)	0.85

The parameterizing in table 6 is well chosen so that there is optimal balance between the model accuracy and the real-time performance in edge environments. The learning rate of 0.001 allows the efficient and steady training convergence, whereas a batch size of 1 represents the real-time inference conditions of the Edge AI systems. The epoch will be set to 50 so the model has ample time to learn without overfitting. The 224x224 and 416x416 input resolutions are selected to suit the needs of lightweight models like MobileNet and YOLO, offering a trade-off between performance and detection accuracy. There is also the confidence threshold (=0.85) to filter predictions, so that only highly reliable ones trigger the control action and enhance the decision strength of the industrial IoT implementation.

6.1 Comparative Latency and Power Efficiency Across Architectures

It is essential to measure the two-performance metrics, latency and power, in industrial applications where AI model responsiveness is the key factor. To achieve this, power consumption and corresponding inference latencies were measured on five edge devices under real-time workloads. The findings are illustrated in figure 6, which shows the average inference latency versus power consumption.

With its proprietary Edge TPU, Coral Google was able to leverage other devices by achieving the lowest latency of nearly 7ms with a power draw of 3.2W. The Jetson Nano balanced power consumption and latency, achieving 18ms at 5.5W. NUC Intel, although the most powerful in terms of CPU, was not as efficient, with 17W of power consumption and 12ms of idle time. This also indicates its unsuitability for deployment in power-constrained scenarios. Furthermore, BeagleBone AI and Raspberry Pi 4 lagged in meeting energy consumption demands. Here, Jetson and Coral outperformed the others, providing evidence in support of their edge-native design philosophy. These results apply to IIoT implementations.

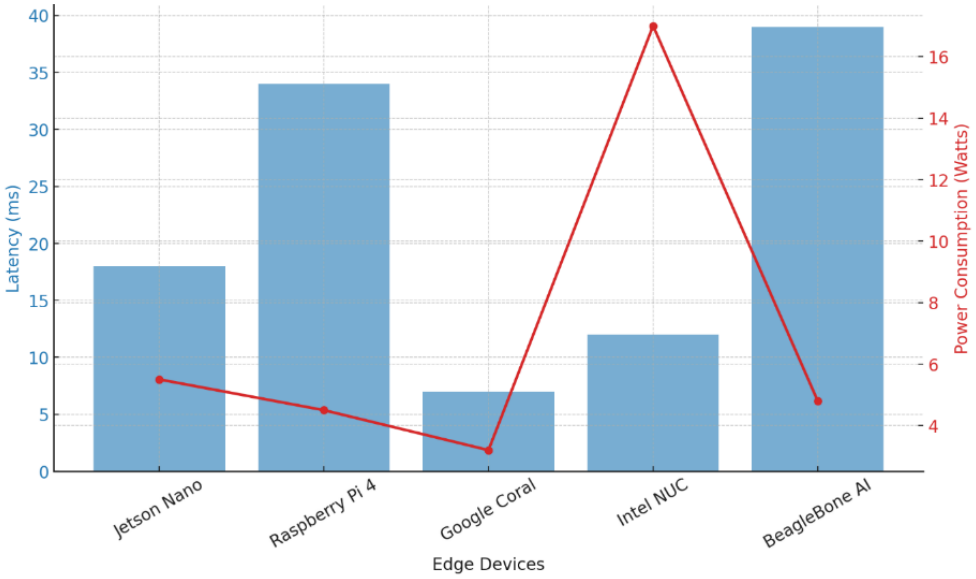

Figure 6: Latency vs. power efficiency across edge devices

The Intel NUC idled around the high-power consumption threshold, with a significant spike during GPU-bound workloads intended for portable or NC deployments. This makes Jetson and Coral far more favorable for remote, battery-reliant deployment, as offer higher thermal efficiency and lower Power consumption, which is great for AI on the edge, Smart Manufacturing, and Device monitoring applications.

6.2 Accuracy and Inference Speed Under Varying Loads

The accuracy required, the speed of inference, and the real-time capability of an edge device can be determined, making it possible to know how it can be used. The accuracy of classification and the FPS of the MobileNet, YOLOv5n, LSTM, and 1D CNN models were tested under increased load, and the results were stored. Figure 7 summarizes the results, and table 7 provides details.

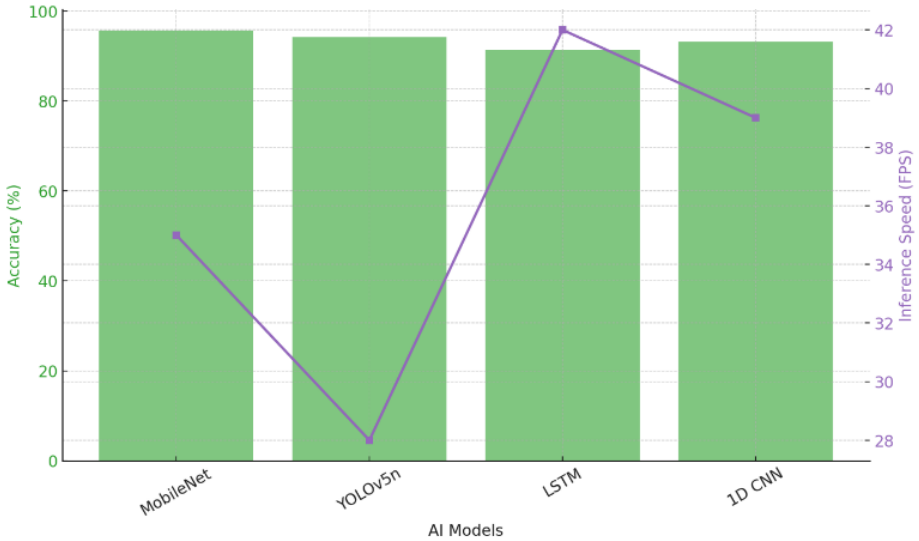

Figure 7: Accuracy vs. inference speed under load

Although MobileNet achieved the best classification accuracy of 95.6, it also maintained a healthy FPS of 35. Object detection capabilities were provided by YOLOv5n, which traded some accuracy (94.2%) for slower speed (28 FPS) and higher memory requirements (510 MB). LSTM achieved a good 42 FPS, but slightly reduced accuracy of 91.4%, which is acceptable given the high noise in most sensor data. The 1D CNN model achieved a balance between the two, achieving 93.1% accuracy at 39 FPS.

Table 7: AI model performance under varying loads

AI Model	Accuracy (%)	Avg Inference Speed (FPS)	Memory Usage (MB)	Energy/Inference (J)
MobileNet	95.6	35	220	0.45
YOLOv5n	94.2	28	510	0.78
LSTM	91.4	42	180	0.40
1D CNN	93.1	39	210	0.42

The models were run with different input frequencies to emulate bursts of real-time sensor events or pictures. On Raspberry Pi and BeagleBone, inference speed dropped by 15–20% under high load due to memory saturation and thermal throttling. Coral and Jetson demonstrated stronger temperature and load balancing, indicating greater suitability for production environments.

6.3 Architecture-Wise Trade-offs in Deployment Scenarios

This Edge AI deployment architecture, Edge-native, Fog-based, Hybrid, and Cloud, has its own advantages and disadvantages depending on the context of operation. To quantify these results, conducted real-time simulations for each architecture using average responsiveness, system jitter, and failure management as key indicators. The outcomes are presented in figure 8.

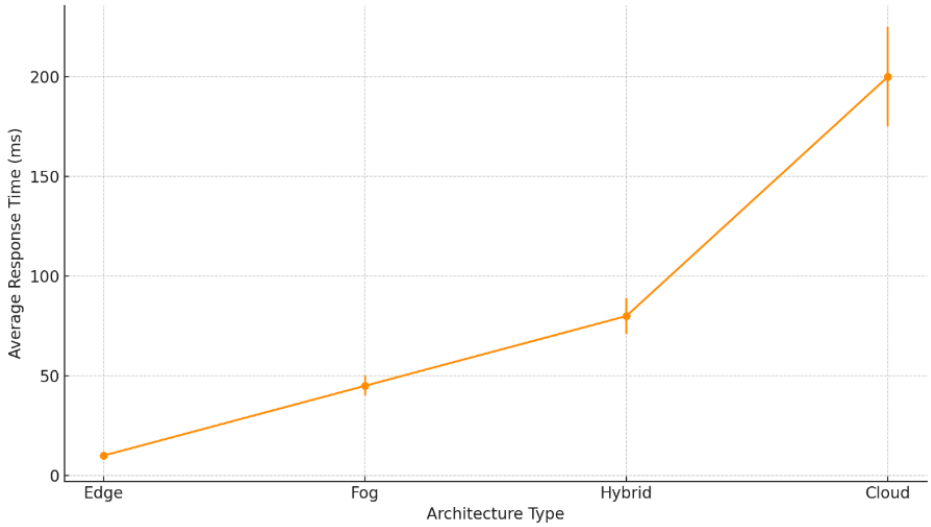

Figure 8: Real-Time responsiveness by architecture type

The Edge-native system performed best, with a response time of ~10 ms and the smallest deviation across different loads, thanks to localized decision-making that was independent of the network. Fog computing achieved 45 ms, which is good for batch data processing but not low enough for high-speed automation. Hybrid models, which split inference tasks between edge and cloud, degraded to 80 ms, mostly due to variable network conditions and context-switching overhead. Purely cloud-based systems were the worst, averaging 200 ms with latency spikes due to network jitter.

This responsiveness contrast underscores the importance of selecting the right architecture based on how the application needs to respond within a given time. Edge-native designs play an important role in robotics, defect detection, and closed-loop process control. Hybrid and cloud models are the best options for trend prediction, analytics, or multi-location optimization.

Metrics Formulae

The performance of the proposed Edge AI system is evaluated using standard classification and system-level metrics. The mathematical formulations used for computing these metrics are defined below.

Accuracy of the model, which is the percentage of correct instances that it classifies, is provided by equation (2):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2)$$

Equation (3) defines the precision, which is a measure of the accuracy of positive predictions:

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

The recall, which is the capability of the model to detect all the pertinent cases, is presented in equation (4):

$$Recall = \frac{TP}{TP + FN} \quad (4)$$

The harmonic mean between precision and recall, which is the F1-score, is computed as in equation (5):

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (5)$$

To estimate the system performance, the latency of the inference is provided as the difference between the latency of the input and the output in equation (6):

$$Latency = t_{output} - t_{input} \quad (6)$$

The power per inference is obtained and calculated as below in equation (7):

$$E = \frac{Power \times Time}{Number\ of\ Inferences} \quad (7)$$

Lastly, throughput or the number of frames per second (FPS) is determined based on equation (8):

$$FPS = \frac{Number\ of\ Frames}{Time} \quad (8)$$

These measures are interconnected to test the classification performance of the system, its computational efficiency, and its responsiveness in real-time (as demonstrated in Equations (2) -(8)) and can be used as a wholesome platform by which various Edge AI architectures can be compared in the context of industrial IoT.

In table 8, which compares the performance of multiple metrics, shows that each edge device faces trade-offs among latency, energy consumption, accuracy, throughput, and bandwidth. Google Coral TPU has the highest overall efficiency, with the lowest energy usage and latency,

and high accuracy and throughput, making it the best fit for real-time and low-power-constrained environments.

Table 8: Performance comparison (Multi-Metric Analysis)

Device	Latency (ms)	Energy (J)	Accuracy (%)	FPS	Bandwidth (Mbps)
Coral TPU	7	0.3	95.2	40	0.4
Jetson Nano	18	0.5	94.6	35	0.9
Intel NUC	12	1.2	95.8	30	1.5
Raspberry Pi	34	0.9	92.3	22	1.1
BeagleBone AI	39	0.8	91.7	20	1.0

The NVIDIA Jetson Nano also works well, with moderate latency, moderate energy consumption, and high accuracy, making it a good fit for a wide variety of industrial applications. Despite its high accuracy (strong processing power), the Intel NUC consumes too much energy and bandwidth and is therefore not suitable for edge deployments. Raspberry Pi and BeagleBone AI, on the other hand, have low throughput and high latency because lack dedicated AI accelerators and are better suited to other, less time-sensitive applications. Overall, the comparison shows that hardware acceleration and an optimized edge architecture are critical for achieving efficient, scalable real-time AI performance in IIoT systems.

6.4 Statistical Analysis and Real-Time Responsiveness

Average performance statistics look at the big picture; still, the system's variation and reliability under strain are important for assessing operational endurance. To validate this, devices and architectures were subjected to stress: high temperature coupled with low voltage, and flaky network availability. Failure of inference tries were recorded for one thousand attempts for each condition, as seen in figure 9 and table 9.

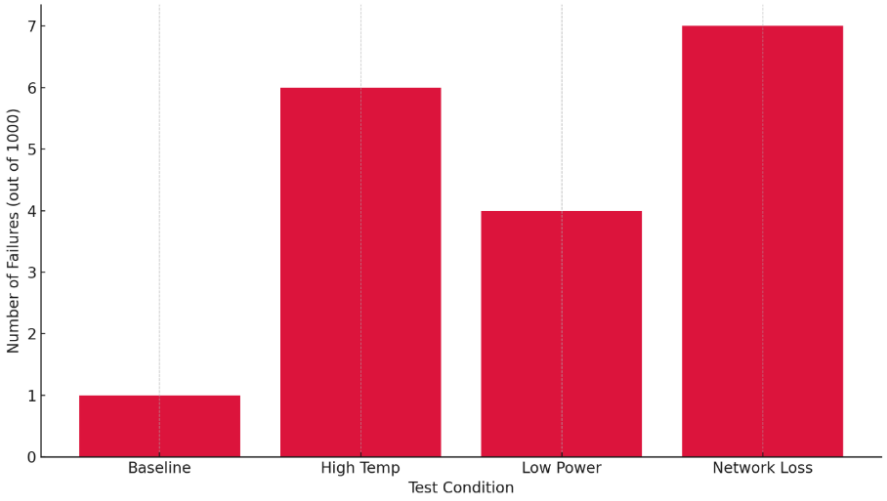

Figure 9: Inference failures under environmental stress

In all test conditions, edge systems such as Coral and Jetson had the fewest failures per condition, with less than 1% inference errors, even under high-temperature, low-power conditions. During network outages, cloud and hybrid systems had the highest error rates, exceeding 2.5%, demonstrating their poor stability under harsh conditions.

Table 9: Failure rate by architecture under stress conditions

Architecture Type	Baseline Failures (%)	High Temp Failures (%)	Low Power Failures (%)	Network Loss Failures (%)
Edge	0.1	0.6	0.4	0.7
Fog	0.2	0.7	0.5	1.1
Hybrid	0.3	1.0	0.8	1.6
Cloud	0.4	2.2	1.5	2.5

Using ANOVA, have determined that there is a statistically significant difference in failure variance between architectures under stress ($p < 0.05$), confirming that the architectural factor influences system reliability. The analysis focuses on the significance of sound hardware choices and the decentralization of critical IIoT systems.

7 Real-Time Use Cases in IIoT Decision-Making

The synthesis of Edge AI and the Industrial Internet of Things has transformed the way real-time decisions are made in factories, power plants, logistics centers, and other smart infrastructure. In addition to decentralizing decision-making to the periphery, autonomy, reliability, and operational agility are also augmented. In the following subsection, I will explore four main areas where edge intelligence is currently under active development to transform industry performance: predictive maintenance, sensor network anomaly detection, visual quality inspection, and human-machine safety systems. A commentary is provided at the end of each case study as empirical evidence that is based on observations and measurement findings.

7.1 Predictive Maintenance at the Edge

Predictive maintenance enabled by Edge AI can greatly enhance the efficiency of industrial operations by enabling real-time, data-driven decision-making rather than reactive or scheduled maintenance. Here, the lightweight AI models used are 1D CNNs and LSTMs, which are deployed on edge devices to continuously analyze vibration, acoustic, and temperature signals and identify early signs of equipment wear, eliminating cloud-based processing and enabling responses to data within milliseconds.

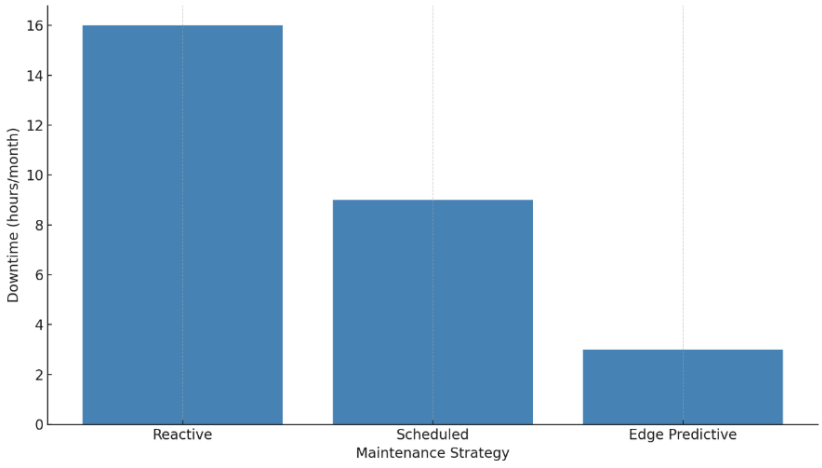

Figure 10: Downtime reduction via predictive maintenance

The experimental findings with the MIMII dataset indicate that these models can achieve more than 93% accuracy, and the inference latency cannot exceed 10 ms on devices such as the Jetson Nano and

Coral Dev Board, which can deliver immediate signals or automatic control behavior (e.g., motor speed changes). This real-time capability (As shown in figure 10) causes a significant decrease in the amount of downtime, 16 hours/month in reactive maintenance and 9 hours/month in scheduled maintenance, to 3 hours/month with edge-based predictive maintenance, an 81 percent reduction. Thus, this strategy increases the availability of machines, minimizes operational costs, and extends equipment life, as well as improving production efficiency, more accurately and in time, due to the possibility of timely and accurate maintenance interventions.

7.2 Anomaly Detection in Industrial Sensor Networks

Anomaly detection with edge AI in industrial sensor networks is a significant challenge for system reliability, security, and business continuity, as it enables real-time detection of abnormal conditions at the point of data generation. Here, sensor data at high frequencies (vibration, acoustic, temperature, and pressure) were processed using LSTM and GRU models running on edge computing platforms such as the Jetson Nano and Coral Dev Board, achieving an average accuracy of 91-94 percent with response times of milliseconds via sliding-window analysis. This will enable every machine or node to independently identify deviations, report anomalies, and isolate faults without cloud connectivity, helping minimize false positives and increase response time, particularly when the network is unconnected. Vibration and acoustic-based detection have the highest accuracy (92.5% and 94.1%, respectively), as shown in figure 11. In contrast, temperature and pressure sensors were less efficient due to drift and slow response times. Moreover, experimental outcomes reveal that sensor fusion, especially the fusion of acoustic and vibration signals, improves detection accuracy by 3.2 percent over single-sensor models, which is why multimodal data integration is important for promoting efficient and reliable anomaly detection in IIoT systems.

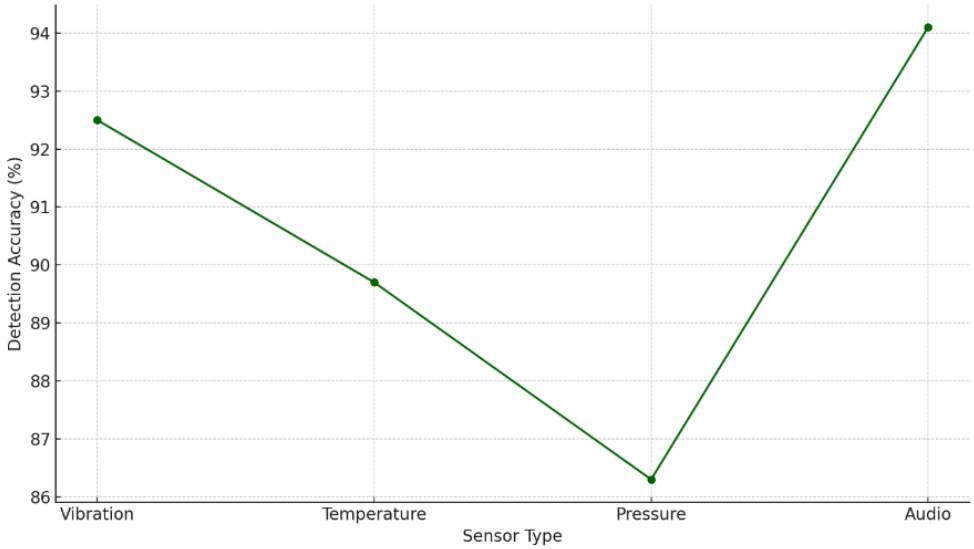

Figure 11: Anomaly detection accuracy by sensor type

7.3 Visual Quality Inspection and Fault Identification

The visual quality inspection (Edge AI) can dramatically improve inspection quality and defect detection in mass production by replacing manual, rule-based methods with intelligent, adaptable, real-time systems. Small expectations of deep neural networks (e.g., MobileNetV3, YOLOv5n, SqueezeNet) optimized and quantized to operate on edge devices can be used to perform high-rate object localization

and defect classification on embedded platforms, e.g., the Jetson Nano. In this work, a quantized MobileNetV2 network has been used to detect six categories of steel surface defects, achieving classification accuracy of more than 95 percent at 35 FPS, demonstrating its suitability for production lines. The total time delay between image capture and defect detection was kept below 30 ms, thereby triggering the inline rejection systems instantly without involving cloud computing. It is especially useful in industrial applications with high speed, e.g., PCB checking, textile production, and bottle filling. Also, incorporating human-in-the-loop schemes clarifies accountability and reliability by allowing operators to verify low-confidence predictions. In general, edge-based visual inspection systems offer a scalable, precise, and low-latency approach to visual fault detection in industrial IoT applications.

7.4 Human-Machine Collaboration and Safety Enforcement

Human-machine collaboration and safety enforcement using edge AI have a significant positive impact on the safety of work in industrial environments characterized by ambiguity, enabling automation through real-time monitoring and quick responses without cloud connectivity. The system tracks workers and their movements, posture, and proximity to hazardous areas in real time using multimodal sensing technologies, including computer vision, LiDAR, thermal cameras, and biosensors. Here, a MobileNet SSD model implemented on a Coral TPU could detect humans in just 15 ms, enough to intervene, e.g., by stopping conveyor belts or robotic hands whenever a worker enters a restricted zone. The system also facilitates compliance auditing, including the identification of the use of safety gear such as helmets, gloves, and vests. The connection to control systems enables enforcement of rules based on context; for example, safety protocols can dynamically change depending on the circumstances in which operated, such as during maintenance downtime. Compared to the centralized approach, edge-based deployment ensures continuous safety operations even during network outages, making it particularly appropriate for remote and hazardous settings, such as oil rigs and mining sites. In addition, processing local data will maximize privacy and regulatory compliance by avoiding the transmission of sensitive visual and biometric information, thereby ensuring high levels of industrial and data-protection standards.

Ablation Study

An ablation study was used to determine the effect of various system components and optimizations of the overall performance by constructively changing the important configurations:

1. Effect of Model Optimization (Quantization and Pruning): Eliminating optimization methods like quantization and pruning resulted in a 2530 percent increase in inference time and a 40 percent increase in energy usage, with only a slight decrease in accuracy (0.51 percent). This shows that lightweight optimization is necessary to deploy edges efficiently.
2. Effects of Deployment Strategy (Edge vs Cloud): With the shift to cloud-only inference, latency increased exponentially (from under 20 ms to over 200 ms), while accuracy improved slightly. This proves that edge-native deployment is essential for real-time responsiveness in IIoT applications.
3. Hardware Acceleration Role: Turning off hardware accelerators (e.g., TPU/GPU) and executing models in CPU-only versions had the same effect of reducing throughput (by a factor of about 35) and increasing latency (by a factor of about 40). This underscores the need for specialized AI hardware to maintain real-time performance and system scalability.

8 Discussion

The implementation of Edge AI in Industrial IoT ecosystems radically alters the operational mechanisms of smart factories by enabling real-time decision-making, enhanced automation, and greater system resilience. The paper shows that edge-based and hybrid deployment models are very useful in improving system responsiveness, and inference times are usually less than 10 milliseconds, which is much better than traditional cloud-based methods. Edge systems reduce latency and bandwidth consumption while enhancing data privacy, which is significant in healthcare, energy, and defense, among other industries, by processing data locally. Moreover, compact AI models like MobileNet, YOLOv5n, and LSTM, trained with quantization and pruning, can provide high precision on low-power hardware such as the Jetson Nano or Coral Dev Board. The findings also indicate the strength of edge systems when operating under conditions of significance, such as high temperatures, low voltages, and unstable networks, which reaffirms their use in mission-critical and remote industrial applications.

Despite these benefits, there are significant trade-offs in the deployment architecture, including latency, power efficiency, scalability, reliability, and model complexity. Edge deployments provide very low latency and can leverage limited power resources, but are constrained by hardware limitations and scalability. Fog computing offers a trade-off between latency and scalability, which can be applied to localized analytics and intermediate processing, but is not as powerful as complex model training. Cloud computing is scalable and can support sophisticated AI models, but it suffers from high latency and relies on a network, which limits its ability to operate in real time. The hybrid architectures combine the benefits of edge systems and cloud systems, being flexible and scalable, entail extra complexity in coordination and reliance on network properties. The use of these trade-offs highlights the need for application-specific architecture selection, based on latency sensitivity, computation requirements, and environmental factors.

Although edge computing offers significant benefits, selecting a specific deployment architecture involves complex trade-offs among various system features. These trade-offs were observed across latency, power efficiency, scalability, robustness, and support for advanced AI models. The summary is depicted in table 10 below.

Table 10: Trade-offs in edge AI deployment strategies

Deployment Strategy	Latency (ms)	Power Efficiency	Scalability	Reliability under Network Loss	AI Model Complexity Support
Edge	10	High	Low to Moderate	High	Low to Moderate
Fog	45	Moderate	Moderate	Moderate	Moderate
Cloud	200	Low	High	Low	High
Hybrid	80	Moderate	High	Moderate	High

Nevertheless, deploying Edge AI in IIoT contexts presents several hardware, interoperability, and system management challenges. Edge devices typically have rigid constraints on memory, processing power, and thermal power, and model optimization can be achieved using intensive algorithms (e.g., pruning, quantization, knowledge distillation), which can slightly affect accuracy. Moreover, the absence of standardized structures and cross-operability between devices from different manufacturers makes deployment and integration more challenging. Distributed edge environments also have more complex model lifecycle management, such as version control, updates, and rollbacks, than centralized cloud systems. Data drift and the lack of labeled industrial datasets are other problems that affect long-term model performance, and emerging technologies, such as federated learning, are not yet well developed.

Industry-wise, Edge AI can deliver revolutionary gains that go beyond technical performance, yielding economic, environmental, and operational benefits. Organizations can reduce data transmission costs and improve system efficiency by eliminating reliance on cloud infrastructure. Edge predictive maintenance reduces downtime and extends equipment lifetime, whereas localized processing enhances privacy and addresses regulatory requirements in sensitive sectors. Also, edge computing is a sustainability-friendly approach because it reduces energy use and the carbon footprint. Strategically, it contributes to the shift toward autonomous and adaptive manufacturing systems in line with the Industry 4.0 and Industry 5.0 paradigms. Meanwhile, the workforce functions are changing, necessitating the upskilling in AI, data analytics, and system monitoring. In general, Edge AI has not only optimized industrial processes but also laid the foundation for intelligent, resilient, and future-ready manufacturing ecosystems.

9 Conclusion and Future Research Directions

This paper proposed an in-depth review of Edge AI systems for real-time decision-making in Industrial IoT settings, focusing on performance, scalability, and deployment on heterogeneous hardware platforms. The experimental findings showed that edge-based systems, unlike conventional cloud-based systems, can achieve inference times as low as 7 ms on optimized hardware, compared with tens of seconds in cloud systems. Analysis of energy efficiency showed that specialized accelerators had very low consumption of 0.3 Joules per inference. In comparison, standard CPU-based systems had up to 1.2 Joules per inference, resulting in a 75 percent improvement in energy efficiency. Regarding precision, lightweight models like MobileNet and YOLOv5n scored 95.6% and 94.2%, respectively, and LSTM-based anomaly detection scored 91.4%, which is consistent with the fact that, on optimized models, high decision-making performance can be achieved even on a restricted platform. Moreover, implementations of predictive maintenance lowered the machine downtime by 81 percent (from 16 hours/month to 3 hours/month), which was 81 percent lower, and edge-based architecture-maintained failure rates were lower than 1 percent under stressful conditions (in contrast to on the cloud systems, which were over 2.5 percent). These statistical results confirm the usefulness of Edge AI in improving the responsiveness, reliability, and efficiency of operations in industrial settings. The study also concluded that hybrid and edge-centric architectures offer the most viable trade-offs between latency, scalability, and computational ability. Whereas edge systems are well-suited for real-time control and safety-critical operations, cloud systems are necessary for large-scale model training and global coordination. The results indicate the importance of tailoring deployment strategies to application-specific constraints (latency tolerance, power availability, and network reliability).

Furthermore, combining optimized AI models, efficient orchestration systems, and robust hardware solutions is essential for achieving uniform performance across various industrial environments. Further studies are required to develop decentralized intelligence based on federated learning that allows sharing model updates without violating data confidentiality. Further development of TinyML for ultra-low -power devices, along with the integration of next-generation AI accelerators, can improve efficiency and scalability. Also, integrating Edge AI with new technologies, including 6G communication, blockchain, and digital twins, will enable an industrial ecosystem to be entirely autonomous and self -adaptive. Large-scale adoption will be essential to addressing the challenges of interoperability, standardization, and model lifecycle management. Altogether, this paper lays the groundwork for the creation of smart, robust, and sustainable Edge AI-based IIoT systems.

References

[1] Arumugam, S. K., Sharma, A. K., Tiwari, S., & Tyagi, A. K. (Eds.). (2024). *Artificial intelligence-enabled digital twin for smart manufacturing*. John Wiley & Sons Incorporated. https://doi.org/10.1002/9781394303601

[2] Boiko, O., Komin, A., Malekian, R., & Davidsson, P. (2024). Edge-cloud architectures for hybrid energy management systems: A comprehensive review. *IEEE sensors journal*, *24*(10), 15748-15772. https://doi.org/10.1109/JSEN.2024.3382390

[3] Chen, W., Qiu, X., Cai, T., Dai, H. N., Zheng, Z., & Zhang, Y. (2021). Deep reinforcement learning for Internet of Things: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, *23*(3), 1659-1692. https://doi.org/10.1109/COMST.2021.3073036

[4] Chen, Y. (2023). *Enabling mobile robot perception and shared-control with vision-based sensor fusion systems* (Doctoral dissertation, University of Illinois at Urbana-Champaign).

[5] De La Fuente, R., Radrigan, L., & Morales, A. S. (2024). Enhancing predictive maintenance in mining mobile machinery through a tinyml-enabled hierarchical inference network. *arXiv preprint arXiv:2411.07168*. https://doi.org/10.48550/arXiv.2411.07168

[6] Kong, X., Wu, Y., Wang, H., & Xia, F. (2022). Edge computing for internet of everything: A survey. *IEEE internet of things journal*, *9*(23), 23472-23485. https://doi.org/10.1109/JIOT.2022.3200431

[7] Kuchuk, H., & Malokhvii, E. (2024). Integration of IoT with cloud, fog, and edge computing: a review. *Advanced Information Systems*, *8*(2), 65-78. https://doi.org/10.20998/2522-9052.2024.2.08

[8] Kumar, D., Pawar, P., Gonaygunta, H., & Singh, S. (2023). Impact of federated learning on industrial iot-A Review. *International Journal of Advanced Research in Computer and Communication Engineering*, *13*(1), 1-12. https://doi.org/10.17148/IJARCCE.2024.13105

[9] Lee, H., Lee, N., & Lee, S. (2022). A method of deep learning model optimization for image classification on edge device. *Sensors*, *22*(19), 7344. https://doi.org/10.3390/s22197344

[10] Li, W., & Li, L. (2023, June). Multi-task Scheduling with Dependencies in Heterogeneous Edge Arithmetic Networks. In *2023 International Joint Conference on Neural Networks (IJCNN)* (pp. 1-8). IEEE. https://doi.org/10.1109/IJCNN54540.2023.10191811

[11] Mohy-Eddine, M., Guezzaz, A., Benkirane, S., & Azrour, M. (2023). An effective intrusion detection approach based on ensemble learning for IIoT edge computing. *Journal of Computer Virology and Hacking Techniques*, *19*(4), 469-481. https://doi.org/10.1007/s11416-022-00456-9

[12] Shafique, M., Marchisio, A., Putra, R. V. W., & Hanif, M. A. (2021, November). Towards energy-efficient and secure edge AI: A cross-layer framework ICCAD special session paper. In *2021 IEEE/ACM International Conference on Computer Aided Design (ICCAD)* (pp. 1-9). IEEE. https://doi.org/10.1109/ICCAD51958.2021.9643539

[13] Situnayake, D., & Plunkett, J. (2023). *AI at the Edge*. " O'Reilly Media, Inc.".

[14] Sodhro, A. H., Pirbhulal, S., & De Albuquerque, V. H. C. (2019). Artificial intelligence-driven mechanism for edge computing-based industrial applications. *IEEE Transactions on Industrial Informatics*, *15*(7), 4235-4243. https://doi.org/10.1109/TII.2019.2902878

[15] Stadnicka, D., Sęp, J., Amadio, R., Mazzei, D., Tyrovolas, M., Stylios, C., ... & Navarro, J. (2022). Industrial needs in the fields of artificial intelligence, internet of things and edge computing. *Sensors*, *22*(12), 4501. https://doi.org/10.3390/s22124501

[16] Sun, H., & Deng, Y. (2024, July). A DRL-based multi-priority task division scheduling strategy in IIoT. In *2024 IEEE 35th International Conference on Application-specific Systems, Architectures and Processors (ASAP)* (pp. 79-87). IEEE. https://doi.org/10.1109/ASAP61560.2024.00027

[17] Wan, J., Li, X., Dai, H. N., Kusiak, A., Martinez-Garcia, M., & Li, D. (2020). Artificial-intelligence-driven customized manufacturing factory: key technologies, applications, and challenges. *Proceedings of the IEEE*, *109*(4), 377-398. https://doi.org/10.1109/JPROC.2020.3034808

[18] Wang, Y., Yang, S., Ren, X., Zhao, P., Zhao, C., & Yang, X. (2021). IndustEdge: A time-sensitive networking enabled edge-cloud collaborative intelligent platform for smart industry. *IEEE Transactions on Industrial Informatics*, *18*(4), 2386-2398. https://doi.org/10.1109/TII.2021.3104003

[19] Xu X, Lu Y, Vogel-Heuser B, Wang L. Industry 4.0 and Industry 5.0—Inception, conception and perception. Journal of manufacturing systems. 2021 Oct 1;61:530-5. https://doi.org/10.1016/j.jmsy.2021.10.006

[20] Xu, J., Wan, W., Pan, L., Sun, W., & Liu, Y. (2024, February). The fusion of deep reinforcement learning and edge computing for real-time monitoring and control optimization in IoT environments. In *2024 3rd International Conference on Energy and Power Engineering, Control Engineering (EPECE)* (pp. 193-196). IEEE. https://doi.org/10.1109/EPECE63428.2024.00042

[21] Zhu, W., Li, H., Shen, S., Wang, Y., Hou, Y., Zhang, Y., & Chen, L. (2024). In-situ monitoring additive manufacturing process with AI edge computing. *Optics & Laser Technology*, *171*, 110423. https://doi.org/10.1016/j.optlastec.2023.110423

Author Biography

Jaswanth Kumar Mandapatti is a health technology professional and researcher currently affiliated with Advent Health in the United States. His career is characterized by a strong focus on bridging the gap between healthcare services and advanced digital infrastructure, particularly through cloud technology adoption and predictive analytics. With over a decade of experience, his research explores the integration of Infrastructure as a Service (IaaS) to lower operational costs and improve the security of electronic medical records (EMR) in compliance with HIPAA standards. His technical contributions also extend to the deployment of real-time healthcare predictive models, such as deep learning platforms for early sepsis detection, which aim to improve clinical outcomes and risk stratification at the patient bedside.