Review

Distributed Intelligence in the Artificial Intelligence of Things: A Comprehensive Review of Architectures, Applications, and Challenges

Leandro Antonio Pazmiño Ortiz * , Alan Cuenca-Sánchez , Byron Loarte

Posted Date: 3 April 2026

doi: 10.20944/preprints202604.0261.v1

Keywords: artificial intelligence of things (AIoT); distributed intelligence; edge-fog-cloud (EFC) distributed intelligence model; edge AI; TinyML; federated learning; industrial IoT (IIoT); smart cities; connected healthcare (IoMT); smart agriculture; MLOps for the edge; security & privacy; interoperability & standards; explainable AI (XAI); human-in-the-loop AIoT; multimodal sensor fusion; energy-harvesting & self-sustaining devices; tactile internet

 doi:10.20944/preprints202604.0261.v1

Review

Distributed Intelligence in the Artificial Intelligence of Things: A Comprehensive Review of Architectures, Applications, and Challenges

Leandro Antonio Pazmiño Ortiz *, Alan Cuenca-Sánchez and Byron Loarte

Escuela de Formación de Tecnólogos, Escuela Politécnica Nacional, Quito 170143, Ecuador
* Correspondence: leandro.pazmino@epn.edu.ec

Abstract

Artificial Intelligence of Things (AIoT) applications increasingly exceed the limits of centralized cloud processing because they require low latency, privacy preservation, scalability, and operational resilience. This review synthesizes distributed intelligence across the edge–fog–cloud continuum through a structured integrative methodology comprising multi-stage literature search, two-stage filtering, and thematic synthesis of more than 100 sources. The analysis covers four representative domains—industrial IoT, smart cities, connected healthcare, and smart agriculture—to identify recurring architectural patterns and shared deployment challenges. The review organizes these challenges around power and computational constraints, data management, security and privacy, interoperability, and model lifecycle management. Building on this synthesis, the paper formalizes an Edge–Fog–Cloud distributed intelligence model and develops a workload-placement taxonomy based on latency, privacy, power, and model complexity. Comparative analysis shows that on-device TinyML is best suited to ultra-low-latency and privacy-sensitive inference, edge and fog layers provide an effective compromise for localized near-real-time intelligence, and cloud infrastructures remain essential for large-scale analytics and model training. Across domains, the evidence supports hybrid multi-layer architectures as the most robust strategy for advanced AIoT deployments. The review also identifies key future directions, including human-in-the-loop AIoT, multimodal sensor fusion, energy-harvesting devices, federated learning, and the Tactile Internet.

Keywords: artificial intelligence of things (AIoT); distributed intelligence; edge-fog-cloud (EFC) distributed intelligence model; edge AI; TinyML; federated learning; industrial IoT (IIoT); smart cities; connected healthcare (IoMT); smart agriculture; MLOps for the edge; security & privacy; interoperability & standards; explainable AI (XAI); human-in-the-loop AIoT; multimodal sensor fusion; energy-harvesting & self-sustaining devices; tactile internet

1. Introduction

Modern Internet of Things (IoT) deployments are producing an unprecedented and accelerating deluge of data from billions of connected devices situated at the network edge. The scale of this phenomenon is staggering. By 2026, it is projected that over 55 billion IoT devices will be in operation worldwide, collectively generating nearly 80 zettabytes (ZB) of data annually [1–3]. This explosive growth in both the number of endpoints and the volume of data they produce is driving a fundamental paradigm shift away from traditional, centralized computing models. The once-dominant approach of funneling all raw sensor readings to distant central cloud servers for processing is becoming technically and economically infeasible due to prohibitive bandwidth costs, inherent network latency, and increasing privacy concerns. In response, industry is pivoting toward edge computing—the practice of bringing computation and data storage closer to the sources of data. This strategic shift is underscored by market analyses indicating that global spending on edge computing will reach

 doi:10.20944/preprints202604.0261.v1

approximately $260 billion by 2026 [4–7], highlighting the urgent economic demand for Artificial Intelligence of Things (AIoT) solutions that intelligently distribute computation throughout IoT networks. The significance of this convergence is further reflected in a recent surge of comprehensive scholarly reviews that are mapping the progress, challenges, and future trajectory of AIoT, firmly establishing it as a critical field of study for the coming decade [8–11].

Traditional, siloed computing paradigms are proving profoundly inadequate for the real-time demands of modern IoT. First, cloud-centric architectures, despite their immense computational power, cannot alone meet the stringent real-time requirements of cyber-physical systems. Even with virtually unlimited cloud resources, the simple act of sending every sensor reading to a distant data center introduces network latency and potential points of failure that are unacceptable for mission-critical applications such as autonomous industrial control, real-time traffic management, or remote telemedicine [12–16]. Furthermore, the cloud-only approach inherently magnifies privacy and security risks, as sensitive data (e.g., high-definition video from public-facing city cameras or personal health metrics from wearable sensors) must leave the trusted local environment for processing, increasing its exposure.

Second, passive IoT sensor networks devoid of local intelligence are equally insufficient. In legacy IoT deployments, so-called "dumb" sensors simply collect and transmit raw data streams for potential offline analysis. This approach not only overwhelms communication links but also represents a colossal missed opportunity for immediate, on-site decision-making. The vast streams of data generated by IoT devices often go tragically underutilized. Historically, it has been estimated that over 90% of all collected IoT data has been discarded, archived without analysis, or used for simple thresholding, failing to generate actionable, timely insight [9,18–20]. Clearly, neither a purely cloud-based model nor a non-intelligent device network can fulfill the real-time, scalable, and privacy-sensitive demands of emerging applications. The convergence of Artificial Intelligence (AI) with IoT—in a distributed, hierarchical manner—has thus become a technological necessity rather than a luxury.

Several survey articles have reviewed elements of this convergence, but a holistic perspective that navigates the intricate interplay across the device-edge-cloud continuum is still needed. For instance, foundational IoT surveys such as Al-Fuqaha et al. (2015) provided an excellent overview of enabling technologies, protocols, and applications but were primarily focused on networking and cloud-centric data processing, thus predating the recent paradigm shift toward on-device machine learning and distributed edge intelligence [21]. Seminal surveys on edge computing [12,13] successfully laid out the vision of latency reduction and bandwidth savings via local processing, yet they typically treat AI workloads as abstract computational tasks and do not delve into the domain-specific nuances of machine learning (ML) applications or the extreme resource constraints addressed by TinyML (tiny machine learning).

Domain-specific reviews offer valuable depth but often lack breadth. For example, excellent reviews on AI in connected health and the Internet of Medical Things (IoMT) [11,14] meticulously detail clinical applications but often consider only a binary choice between cloud analytics or single-hop edge processing. They seldom synthesize the cross-domain architectural challenges or account for the full computational continuum, from tiny on-device models performing real-time biosignal analysis to massive cloud-based models for epidemiological research. In short, prior surveys have tended to focus either on IoT without pervasive AI, on AI in a centralized context, or on a single application domain, generally neglecting the critical interplay and optimization opportunities across different system layers and application sectors.

For instance, while comprehensive surveys on *lightweight deep learning* provide crucial methodologies for creating efficient algorithms [22], and domain-specific reviews offer detailed application insights in areas such as cardiovascular health monitoring [23], these works often operate within their respective silos. Similarly, surveys exploring the convergence of embedded systems, edge computing, and machine learning conceptually frame the *Internet of Intelligent Things* [24]. However, there remains a distinct gap in the literature for a survey that systematically analyzes the *cross-level optimization*

 doi:10.20944/preprints202604.0261.v1

between the device, edge, and cloud layers while synthesizing architectural patterns, domain-specific challenges, and the critical co-design of algorithms, hardware, and communication protocols. While recent works have begun to chart this territory [25], a holistic, forward-looking synthesis that can guide the development of next-generation, scalable, and trustworthy AIoT systems remains an open and pressing need.

This survey addresses these gaps by providing the first holistic review of distributed intelligence in AIoT, systematically spanning architectures, application domains, and operational challenges. We synthesize knowledge across four pivotal and diverse domains—Industrial IoT (IIoT), Smart Cities, Connected Healthcare, and Smart Agriculture—which have traditionally been treated in separate literature streams, to identify common architectural patterns and recurring challenges in deploying AI across the edge–fog–cloud continuum.

The key novel contributions of this work are fourfold:

1. **A Systematic Framework for AI Workload Partitioning:** Grounded in the established Edge–Fog–Cloud (EFC) paradigm, we develop a systematic framework for partitioning AI workloads in AIoT systems. This contribution provides a novel and practical taxonomy that guides the strategic placement of AI tasks—from lightweight on-device inference to complex cloud-based training—based on specific performance and operational metrics such as latency, bandwidth, privacy, and model complexity.
2. **A Cross-Domain Taxonomy of Challenges:** We conduct a comprehensive cross-domain analysis of challenges—from power constraints and data management to security and model lifecycle—distilling a detailed taxonomy of problems that is applicable to virtually any AIoT deployment, thereby providing a unified lens through which to view system design.
3. **An In-depth Comparative Evaluation of AIoT Architectures:** We deliver a detailed comparative evaluation of different AIoT architectural patterns (on-device vs. edge vs. cloud vs. hybrid) using clearly defined performance metrics, reinforced by qualitative analysis in tables and quantitative results from benchmark studies visualized in new figures.
4. **Actionable Guidelines and Frameworks for Practitioners:** We provide practical guidelines for AIoT development, including a structured decision framework for workload placement and a review of state-of-the-art hardware and software platforms that enable the design and deployment of real-world AIoT solutions.

To the best of our knowledge, this is the first survey to integrate these critical elements into a single, unified treatment of distributed intelligence in AIoT. The remainder of the paper is structured as follows. Section 2 describes the review methodology, including the literature search strategy, filtering criteria, and thematic-synthesis approach. Section 3 characterizes the reviewed literature corpus. Section 4 provides the critical state-of-the-art review for the four target domains. Section 5 introduces the foundational concepts and formalizes the Edge–Fog–Cloud (EFC) distributed intelligence model that underpins our analysis. Section 6 examines the major cross-domain challenges in deploying AIoT systems, including power and computation limits, data deluge management, interoperability, security and privacy, and model-lifecycle management. Section 7 compares the principal AIoT architectural patterns using defined performance and operational metrics. Section 8 analyzes the software ecosystem for distributed AIoT, spanning runtimes, platforms, middleware, and orchestration. Section 9 translates these insights into practice through a workload-placement framework, hardware survey, and case studies. Section 10 discusses open challenges and future research directions. Section 11 acknowledges the scope and limitations of the survey and highlights priorities for further inquiry. Section 12 concludes the paper.

2. Methodology

To capture the rapidly evolving and inherently multidisciplinary landscape of AIoT, we adopted a structured, systematic search and integrative review methodology. The breadth of the topic—spanning hardware engineering, software development, networking protocols, and multiple distinct application

 doi:10.20944/preprints202604.0261.v1

domains—precludes the use of a rigid, narrow-protocol systematic review (e.g., PRISMA, which is better suited for highly specific clinical or empirical questions with uniform outcome measures). Instead, our approach was designed to be transparent, comprehensive, and reproducible, with the explicit goal of synthesizing high-impact scholarly research, influential industry reports, and foundational technical documentation into a cohesive and critical analysis of the field.

Our methodology followed a rigorous three-phase process: (1) a systematic and multi-staged literature search and collection protocol, (2) a meticulous two-stage filtering and selection process based on predefined criteria, and (3) an inductive thematic synthesis of the final corpus to identify core themes and insights.

2.1. Phase 1: Literature Search and Collection

We conducted an iterative and comprehensive search across several key academic and technical databases to form our initial literature corpus, ensuring coverage of both peer-reviewed research and cutting-edge industry developments.

- **Databases Searched:**
  - **Academic Databases:** IEEE Xplore, ACM Digital Library, Scopus, and ScienceDirect were chosen for their comprehensive coverage of engineering, computer science, and applied sciences literature.
  - **Pre-print and Broad Search:** arXiv was systematically searched to include cutting-edge, pre-peer-reviewed research that often precedes formal publication. Google Scholar was used as a supplementary tool to capture a wider range of sources, including highly cited industry white papers, influential blog posts from research labs, and reports from industry consortiums.
- **Search Strategy and Query Formulation:** Our search was conducted in three strategic stages to ensure both breadth in capturing foundational concepts and depth in specific application areas.

1. **Architectural and Foundational Search:** This initial stage focused on core technologies, conceptual models, and overarching challenges. We used broad keywords combined with terms like 'survey' and 'review' to identify foundational and summary literature that could provide a structural backbone for our review.

   ("Artificial Intelligence of Things" OR "AIoT" OR "Edge AI" OR "TinyML" OR "Embedded AI" OR "Federated Learning") AND (survey OR review OR architecture OR framework OR challenges)

2. **Domain-Specific Search:** The second stage drilled down into specific use cases and implementations by combining architectural keywords with application domain terms. This allowed us to find literature detailing the practical application of AIoT in our chosen sectors.

   ("Edge Computing" OR "On-Device AI") AND ("Industrial IoT" OR "IIoT" OR "Smart Factory") AND ("predictive maintenance" OR "anomaly detection")

   ("AIoT" OR "Edge AI") AND ("Smart Cities") AND ("traffic management" OR "public safety" OR "urban computing")

   ("Internet of Medical Things" OR "IoMT" OR "Smart Healthcare") AND ("remote patient monitoring" OR "wearable sensors" OR "biosignal processing")

3. **Technology and Platform Search:** The final stage targeted specific hardware, software, and development frameworks to gather practical implementation details, performance data, and real-world deployment insights.

   "TensorFlow Lite Micro" AND (performance OR benchmark OR microcontroller)

   ("AWS Greengrass" OR "Azure IoT Edge") AND ("ML deployment" OR "MLOps" OR "case study")

 doi:10.20944/preprints202604.0261.v1

2.2. Phase 2: Filtering and Selection

From an initial pool of several hundred documents, we applied a structured two-stage filtering process to arrive at the final corpus of over 100 highly relevant and high-quality sources.

- **Inclusion Criteria:**
  - **Publication Type:** Peer-reviewed journal articles, highly-selective conference papers (e.g., from top-tier IEEE/ACM conferences), and comprehensive survey/review articles. Authoritative industry white papers and technical reports from recognized technology leaders (e.g., NVIDIA, ARM, Google) and influential industry consortiums (e.g., OpenFog/Industrial Internet Consortium, LF Edge) were also included to ensure practical relevance.
  - **Time Frame:** We primarily focused on sources published between 2016 and 2026 to capture the most recent advancements in this fast-moving field. Foundational, highly cited papers published prior to 2018 (e.g., seminal works on Fog Computing) were retained for essential historical context.
  - **Relevance:** Sources must explicitly and substantially address the intersection of Artificial Intelligence/Machine Learning and the Internet of Things, particularly in a distributed context.
  - **Language:** English.
- **Exclusion Criteria:**
  - Sources focusing purely on IoT networking protocols or cloud infrastructure without a substantial AI/ML component.
  - Sources focusing purely on abstract or cloud-based AI algorithms without a clear and practical connection to edge, fog, or IoT data sources and constraints.
  - Short abstracts, posters, editorials, presentation slides, and non-technical marketing materials.
  - Sources from non-reputable, predatory, or unverified publishers.

The title and abstract of each identified paper were initially screened for relevance by at least two authors. Full-text articles of the shortlisted papers were then reviewed against the inclusion/exclusion criteria, with a focus on selecting seminal surveys that define a sub-field, highly cited research that introduced key concepts, and papers that were representative of specific domains, technologies, or challenges.

2.3. Phase 3: Thematic Synthesis

To structure our review and extract key insights in a systematic manner, we employed an inductive coding and thematic synthesis process. This bottom-up approach allows themes to emerge organically from the literature itself rather than being imposed pre-emptively.

1. **Familiarization and Open Coding:** Each selected paper was read and reviewed in detail. During this stage, we assigned descriptive codes to key concepts, methodologies, identified challenges, proposed solutions, and recurring ideas. Examples of initial codes included 'power constraints on MCUs,' 'privacy concerns in IoMT,' 'model compression techniques,' 'real-time latency requirement in IIoT,' and 'interoperability standards.'
2. **Theme Development (Axial Coding):** We then grouped these initial, fine-grained codes into higher-level, coherent thematic clusters. For example, codes such as 'on-device inference,' 'TinyML models,' 'quantization,' 'pruning,' and 'microcontroller AI' were consolidated into the broader theme of 'On-Device AI and TinyML Architectures.' Similarly, codes related to HIPAA compliance, data encryption, model inversion attacks, and federated learning were grouped under the major theme of 'Security & Privacy in AIoT Systems.' This iterative process of grouping and refining codes directly led to the thematic categorization presented in the next section (Table 1).
3. **Thematic Synthesis and Narrative Construction:** Finally, we synthesized the findings within and across these developed themes to construct the narrative of this review. We performed a cross-case analysis, comparing how challenges and solutions were discussed across different

 doi:10.20944/preprints202604.0261.v1

application domains (e.g., how power constraints manifest differently and are solved differently in resource-scarce Smart Agriculture versus power-available Smart Factories) to draw out the cross-domain challenges that form the basis of Section 6. This structured, evidence-grounded approach ensures that our analysis is robustly rooted in the literature and provides a solid foundation for our conclusions and recommendations.

3. Characterization of Reviewed Literature

Following the systematic methodology outlined in Section 2, our final corpus provides a comprehensive and contemporary snapshot of the AIoT research landscape. This section characterizes the selected literature by visualizing its distribution across key domains and over time (Figure 1) and by summarizing the key thematic categories that emerged from our inductive analysis (Table 1).

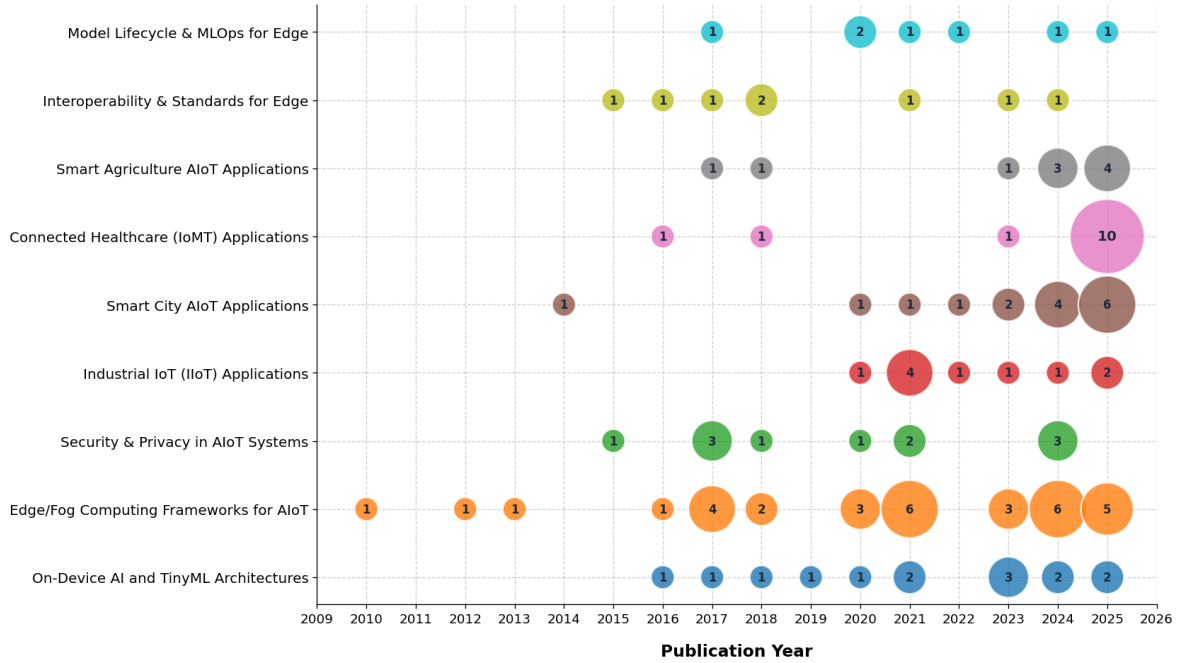

Figure 1. Visualization of the literature corpus in a bubble chart by publication year and domain emphasis. The chart illustrates the growing volume and diversification of AIoT research, with significant recent acceleration in specialized domains like Healthcare (IoMT) and Industrial IoT (IIoT).

Figure 1 visualizes our literature corpus using a bubble chart that maps publications by year against their primary domain focus. The x-axis represents the publication year (from 2015 to 2026), the y-axis categorizes sources by primary emphasis (IIoT, Smart City, Healthcare, Agriculture, or Cross-Domain/Foundational), and the size of each bubble indicates the relative volume of publications or the prominence of that theme in that particular year. As the chart clearly illustrates, the volume of AIoT-related publications shows a strong and consistent upward trend over the last decade across all domains. We observe notable surges around the 2020–2021 period, which correlates with the maturation and mainstream research adoption of key enabling technologies like TinyML and more accessible edge AI hardware.

The healthcare and IIoT domains exhibit consistently large 'bubbles' in recent years, reflecting intense research and commercial interest. This is driven by high-value applications, such as the development of federated learning for privacy-preserving analysis of sensitive health data, and the deployment of AI-based predictive maintenance in industrial settings to reduce costly downtime. While publications in Smart Agriculture are fewer in number, the steady growth of this bubble indicates a nascent but rapidly expanding research community focused on addressing the unique challenges of that domain, such as connectivity and power constraints. The chart also underscores the interdisciplinary nature of AIoT, with relevant research being published in a wide array of venues,

 doi:10.20944/preprints202604.0261.v1

from specialized journals on sensors and electronics to top-tier AI conferences and domain-specific application journals.

Beyond this thematic distribution, the literature reveals a clear evolutionary trajectory in the focus of research. Early foundational papers, such as Aouedi et al. (2024) [26], were concerned with defining the initial convergence of AI and IoT and outlining the potential of the paradigm. This was followed by a wave of domain-specific explorations and architectural proposals between 2020 and 2022, where researchers focused on demonstrating feasibility in specific contexts [8,9]. More recently, as evidenced by the latest surveys and research papers, the focus has matured to tackle more complex, second-order challenges. This includes developing sophisticated frameworks for Explainable AI (XAI) in IoT to build trust and accountability in autonomous systems [27], leveraging technologies like blockchain for secure and decentralized data sharing and model provenance [28], and creating holistic surveys that synthesize years of progress into a cohesive, forward-looking vision [10,29]. This trajectory from foundational concepts to practical, trust-oriented solutions underscores the rapid maturation of the AIoT field.

Table 1 provides a thematic summary of the primary literature categories identified during our synthesis. It lists the number of key sources we identified in each category and provides a representative citation that exemplifies the work in that area. This table illustrates how the corpus comprehensively spans both technology-centric themes (e.g., architectures, security, MLOps) and application-centric themes (e.g., smart cities, connected healthcare). For instance, we identified a substantial body of work focusing on On-Device TinyML techniques (over 15 primary sources, exemplified by the comprehensive survey by Capogrosso et al. [30]), and a similarly large and growing set of sources on AIoT security and privacy frameworks (around 12 core sources, represented by foundational surveys like Yang et al. [31]). For each of the major application domains, we categorized approximately 10–15 significant sources that were representative of key applications and challenges. This thematic classification directly guided the organization of our critical analysis in the subsequent sections of this paper.

Table 1. Thematic Summary of Reviewed Literature: Key categories of AIoT research literature, the approximate number of primary sources identified in each category, and an example citation representing seminal or comprehensive work in that category.

Theme / Category	Approx. # of Key Sources	Representative Work
On-Device AI and TinyML Architectures	14	Capogrosso et al. (2023)[30]
Edge/Fog Computing Frameworks for AIoT	33	Shi et al. (2016) [32]
Security & Privacy in AIoT Systems	11	Sicari et al. (2015) [33]
Industrial IoT (IIoT) Applications	10	Wang et al. (2021) [34]
Smart City AIoT Applications	10	Zanella et al. (2014) [35]
Connected Healthcare (IoMT) Applications	13	Tariq et al. (2020) [36]
Smart Agriculture AIoT Applications	10	Wolfert et al. (2017) [37]
Interoperability & Standards for Edge	8	OpenFog Consortium (2017) [38]
Model Lifecycle & MLOps for Edge	7	Verma and Santhanam (2024) [39]

By employing this structured methodology, we have ensured that our review is comprehensive, up-to-date, and balanced, providing a solid foundation for a critical examination of the state-of-the-art across different domains, which is presented in the next section.

4. State-of-the-Art and Critical Literature Review

In this section, we critically review the application of AI techniques in four major IoT domains—Industrial IoT (IIoT), Smart Cities, Connected Healthcare (IoMT), and Smart Agriculture.

 doi:10.20944/preprints202604.0261.v1

For each domain, we acknowledge representative successes from prior work to establish the current state-of-the-art, and then we critically identify the persistent limitations and unmet needs. These identified gaps directly inform and motivate the focus of later sections; we explicitly note how our survey's analyses of architectures, challenges, and frameworks (in Sections 5–9) are designed to address the specific shortcomings identified in each domain.

4.1. Industrial IoT (IIoT)

The industrial sector was among the earliest adopters of IoT and is now at the forefront of integrating AI to create intelligent, autonomous, and efficient operations, a paradigm often referred to as Industry 4.0. Recent comprehensive surveys and systematic reviews on industrial IoT [40,41] have focused specifically on the opportunities and challenges in industrial AIoT, highlighting its transformative role in enabling predictive maintenance, enhancing automation, and optimizing complex supply chains.

Key Contributions and Successes: A primary success story in IIoT is the development of AI-driven predictive maintenance systems. These systems monitor continuous streams of sensor data (e.g., vibration, acoustic, thermal) from critical machinery to predict failures before they occur, as demonstrated by platforms like TIP4.0 [42]. For example, numerous studies have demonstrated that machine learning models, particularly deep learning models like LSTMs or CNNs, can accurately predict equipment malfunctions days or even weeks in advance by analyzing subtle patterns in high-frequency sensor data from IIoT networks in real-time [43]. Such systems have achieved significant and verifiable cost savings, with some reporting up to a 30% reduction in maintenance costs and a 70% reduction in downtime in factories, refineries, and energy plants. The evolution continues with sophisticated data-driven approaches that integrate edge intelligence for holistic smart manufacturing, optimizing not just individual machines but entire production lines [44].

Critical Gaps and Unmet Needs: Despite these successes, a critical gap in the state-of-the-art IIoT lies in the scalable and robust deployment of these AI models on the factory floor, especially on resource-constrained embedded devices at the "point of failure." While many pilot studies and academic papers assume that sensor data is continuously streamed to a powerful on-site server or the cloud for analysis, this is often not practical in real-world industrial environments. Many settings involve battery-powered sensors attached to rotating machinery, sensors on mobile assets like autonomous guided vehicles (AGVs), or monitoring devices on remote infrastructure like pipelines, where constant, high-bandwidth connectivity is not guaranteed and power is at a premium.

Existing solutions often lack a clear framework for embedding intelligence directly onto these constrained sensors (e.g., running a TinyML model for anomaly detection directly on a vibration sensor node). This capability is crucial for enabling predictions to be made in real time, with minimal latency, even during intermittent network connectivity. In other words, the IIoT field currently excels at using AI to analyze data but struggles with the strategic question of where that analysis should occur to achieve maximum reliability, minimal latency, and optimal resource usage.

How This Survey Addresses the Gap: This survey directly addresses this critical deployment gap by providing a deep examination of distributed AIoT architectures (Section 6)—including on-device and edge intelligence—that can bring predictive maintenance algorithms closer to the machines they monitor. We also provide a detailed discussion of model compression and optimization techniques (Sections 5 and 7) that enable complex AI models to be pruned, quantized, and optimized to fit on the constrained embedded devices commonly found in IIoT, thereby directly tackling the practical deployment challenge in industrial settings.

4.2. Smart Cities

The application of AIoT in smart cities continues to be a major driver of innovation, with recent work focusing on improving urban efficiency, public safety, and environmental sustainability through intelligent transportation, energy management, and surveillance systems [45]. Smart city initiatives deploy extensive IoT sensor networks—including cameras, environmental sensors, traffic monitors,

 doi:10.20944/preprints202604.0261.v1

and smart meters—throughout urban environments, and are increasingly leveraging AI to manage city infrastructure and services proactively.

Key Contributions and Successes: Notable contributions include the development of AI-based adaptive traffic management systems. In these systems, computer vision algorithms analyze real-time video feeds from intersection cameras to dynamically optimize traffic light timing, resulting in measured reductions in congestion and vehicle emissions [46,47]. Cities have also successfully piloted smart lighting and energy grids that use AI-powered predictive models to adjust energy distribution based on real-time usage patterns and anticipated demand, leading to significant energy savings [48]. In the realm of public safety, AI-powered systems can detect anomalies in public spaces, such as gunshots from acoustic sensors or unusual crowd movements from video analytics, enabling faster emergency response [49]. Emerging approaches are further enhancing urban sustainability through advanced techniques like explainable federated learning for optimizing energy consumption in vehicular networks [50]. These successes clearly demonstrate the immense potential of AIoT to improve the quality of urban life, with integrated AI-IoT technologies driving comprehensive smart city transformation [51].

Critical Gaps and Unmet Needs: Despite numerous successful prototypes, the current landscape of smart city AIoT deployments is often fragmented and faces significant scalability and interoperability challenges. A core identified gap is the lack of a unified, scalable architecture that can integrate diverse citywide AIoT applications and manage millions of heterogeneous devices [52]. Many projects remain as vertical silos—for example, a traffic management AI system and a public safety surveillance AI system might operate completely independently, each sending large, redundant data streams to separate cloud backends [53]. This fragmentation leads to duplicated infrastructure, exorbitant backhaul bandwidth usage, and data isolation that prevents holistic, city-wide insights [54].

Moreover, real-time city applications, such as autonomous intersection management or immediate emergency response, suffer when relying on distant cloud servers; network latency or outages can have direct and severe public safety implications. While edge computing is frequently proposed as the solution, current deployments often use a few powerful edge servers per application, which can become localized bottlenecks themselves as the number of connected devices grows [55]. Interoperability is another critical and persistent challenge: IoT devices and AI models from different vendors often lack standardized data formats, APIs, or communication protocols, preventing the creation of an integrated, city-wide intelligence layer [56]. While recent approaches propose sophisticated eco-city architectures and digital twin systems to address these integration challenges, their widespread adoption remains limited due to complexity and cost [57].

How This Survey Addresses the Gap: Our survey addresses these smart city gaps by highlighting distributed frameworks that allow AI models to be deployed across a hierarchy of city infrastructure—from intelligent sensors on street poles, to local edge gateways in neighborhoods, to municipal cloud data centers. This tiered approach directly improves scalability and resiliency. In Section 6, we specifically discuss interoperability and standardization as a key challenge, reviewing initiatives like open data platforms and common IoT standards that aim to break down data silos. Furthermore, the Edge–Fog–Cloud (EFC) model we introduce in Section 5 provides a concrete blueprint for cities to decide which AI tasks should be performed on-site for low latency (e.g., immediate hazard detection by a roadside unit) and which should be sent to central servers for deeper, non-urgent analytics (e.g., long-term city-wide traffic pattern analysis).

4.3. Connected Healthcare (IoMT)

The field of the Internet of Medical Things (IoMT) is advancing rapidly, with a strong focus on enabling remote patient monitoring, early disease detection, and personalized diagnostics. Comprehensive surveys and systematic reviews [58,59] provide a deep dive into the specific advancements and opportunities in AIoT for smarter healthcare, reinforcing the critical need for robust privacy, security, and real-time responsiveness that our proposed framework addresses.

 doi:10.20944/preprints202604.0261.v1

Key Contributions and Successes: In the healthcare domain, AI is increasingly applied to the continuous stream of data from IoMT devices—comprising wearable sensors, smart medical devices, and connected home health monitoring systems—to enable proactive and personalized care [60]. Recent research has focused on developing secure and user-centric frameworks for these IoMT ecosystems that address both technical implementation and human adoption factors [61,62]. Significant accomplishments include wearable ECG and blood pressure monitors that use embedded AI to detect cardiovascular anomalies like arrhythmias or hypertensive events in real time and alert clinicians or caregivers [63,64]. Similarly, smart insulin pumps and medication dispensers can leverage AI predictions to dynamically adjust dosages based on a patient's real-time needs. Within hospital settings, AI-powered IoT systems track patient vital signs and movements to predict and prevent adverse events like falls or the onset of sepsis, leading to improved patient outcomes [65]. These integrated patient monitoring systems leverage smart sensor technology and IoT connectivity to create comprehensive, data-driven healthcare ecosystems [66].

Critical Gaps and Unmet Needs: Despite these advances, formidable gaps related to privacy, safety, and regulatory compliance remain major barriers to widespread IoMT deployment. Healthcare data is highly sensitive and personal, and traditional IoT architectures that stream raw patient data to third-party cloud servers raise significant concerns under stringent privacy laws (e.g., HIPAA in the US, GDPR in the EU) and can erode patient trust. While AI analysis of this data is immensely valuable, many existing solutions are cloud-dependent, meaning a patient's continuous vital signs or a video feed from a home monitoring camera may be constantly leaving the local device for analysis. This not only poses significant privacy risks but also makes the system critically reliant on network connectivity—a dropped connection could delay or prevent a life-saving health alert.

Moreover, the extreme heterogeneity of IoMT devices—from smartwatches to implanted sensors—means that device capabilities vary widely; yet, few frameworks exist to manage distributed AI across such a diverse medical device fleet. Another critical gap is the effective integration of a human-in-the-loop: healthcare AIoT must often involve clinicians and patients in the decision-making process, but current systems seldom provide intuitive mechanisms for users to provide feedback, ask for explanations, or correct the AI's actions in real time.

How This Survey Addresses the Gap: Our survey addresses these IoMT gaps in multiple ways. In Section 5, we devote significant attention to Security, Privacy, and Trust, discussing techniques such as on-device encryption, fine-grained access control, and federated learning, which can keep personal health data localized while still enabling the training of powerful AI models—directly speaking to the privacy concerns in IoMT. We also highlight human-in-the-loop AIoT as a key future direction (Section 9), outlining how AI systems can be designed to collaborate with and augment healthcare professionals rather than operating as opaque "black boxes." From an architectural standpoint, the EFC model (Section 4) and our proposed workload placement framework (Section 8) offer concrete guidance on deploying healthcare AI tasks on-device (e.g., in a wearable for immediate arrhythmia detection) or at an edge gateway (e.g., a home health hub that securely aggregates data from multiple sensors) to ensure both timely responses and data sovereignty.

4.4. Smart Agriculture

The unique challenges of deploying AIoT in agricultural settings have been the subject of recent, specialized reviews. Muhammed et al. (2024) provide a thorough review of architectures and technologies for smart agriculture, emphasizing applications like crop/pest monitoring and resource optimization, which are prime candidates for distributed, on-device intelligence that can operate in harsh and remote environments [67]. Smart agriculture applies IoT and AI to farming, livestock management, and environmental monitoring to increase yields, improve sustainability, and reduce resource usage, with comprehensive reviews documenting the integration of these emerging technologies [68,69].

Key Contributions and Successes: Achievements to date include sophisticated precision agriculture systems where networks of soil moisture sensors, local weather stations, and crop health cameras

 doi:10.20944/preprints202604.0261.v1

feed data into AI models that optimize irrigation schedules and fertilizer application with high spatial and temporal resolution [70]. Drones and autonomous robots equipped with AI-powered computer vision are now used for early pest detection and targeted, micro-dose pesticide spraying, significantly cutting chemical use while improving crop health, a feat made possible by powerful, edge-optimized devices [71]. AI-driven predictive models also help farmers forecast crop yields and detect the early onset of diseases by analyzing multi-spectral sensor and imaging data. These innovations demonstrate how AIoT can make a substantial contribution to global food security and more sustainable farming practices.

Critical Gaps and Unmet Needs: The agriculture domain exemplifies a unique and challenging set of constraints for AIoT that current deployments only partially address. Farms often span vast geographic areas with limited or non-existent connectivity and power infrastructure. Devices like in-field sensors and solar-powered cameras must be deployed in remote locations with only intermittent wireless coverage and extremely strict energy budgets. Cloud-based analytics become impractical when connectivity is sparse and unreliable; yet, most advanced AI models for agriculture (e.g., complex deep learning models for image-based crop disease classification) are developed and trained assuming ready access to powerful cloud GPU servers or at least well-connected farm office computers [11].

There is a significant gap in frameworks and technologies for robust, off-grid AIoT. This refers to systems that can perform significant AI computation on low-power, solar-powered edge devices or microcontrollers directly in the field, uploading only minimal data summaries or critical alerts when a connection becomes available [72,73]. Another major limitation is that agricultural IoT deployments are highly heterogeneous, often comprising a mix of legacy sensors, modern drones, and proprietary farm machinery, all lacking common standards, which severely complicates data integration for AI models. Furthermore, whereas an industrial or urban device might receive regular maintenance, remote farm devices must often operate autonomously and unattended for long periods, raising the importance of designing self-sustaining and fault-tolerant AIoT systems.

How This Survey Addresses the Gap: In response to these gaps, our survey highlights approaches to enable distributed intelligence in constrained and intermittently connected environments. We specifically examine power and computational constraints in depth in Section 5.1, discussing how techniques like model quantization and event-driven sensing can allow sophisticated AI algorithms to run on battery or solar-powered devices—a topic directly relevant to off-grid agriculture sensors. Section 9.3 on Energy-Harvesting and Self-Sustaining AI looks at emerging research on battery-less IoT devices that could operate perpetually on harvested solar or ambient energy [74]. Additionally, our review of hardware platforms (Section 8.1) covers the latest generation of low-power AI-capable microcontrollers and edge accelerators that are suitable for deployment on farms. By outlining these technologies and strategies, we provide a foundation for designing the next generation of smart agriculture AIoT systems that can overcome current connectivity and power limitations.

5. Foundational Concepts and a Model for Distributed AIoT

Before analyzing specific architectures and solutions, it is essential to establish a clear conceptual foundation. This section first provides a comprehensive definition of the Artificial Intelligence of Things (AIoT), distinguishing it from traditional IoT and standalone AI paradigms. We then introduce the paper's primary conceptual framework: the Edge–Fog–Cloud (EFC) Distributed Intelligence Model. This model provides a structured way to understand how AI workloads—from sensing and real-time inference to large-scale learning—can be strategically stratified across device, edge, fog, and cloud layers. Finally, we clarify related foundational concepts, such as the nuanced differences between edge and fog computing, and the principles of federated learning, which are central to our subsequent analysis.

5.1. Defining the Artificial Intelligence of Things (AIoT)

The Artificial Intelligence of Things (AIoT) represents the deep and synergistic integration of Artificial Intelligence (AI) capabilities with the Internet of Things (IoT) infrastructure. This convergence

 doi:10.20944/preprints202604.0261.v1

creates systems where connected devices can not only sense their environment and communicate data but can also perceive, reason, learn, and act intelligently and often autonomously. In its essence, AIoT = AI + IoT: it leverages the massive, real-world data streams from IoT sensors and employs a wide range of AI techniques (including machine learning, deep learning, and reasoning algorithms) to extract meaningful insights, predict future states, and automate decision-making processes directly within the IoT network [75].

In a traditional IoT setup (pre-AIoT), devices are typically low-power sensors or actuators that function as simple data collectors, transmitting raw information to a central server or cloud platform. Any 'intelligence' resides exclusively in back-end applications that analyze this data, often in batches and with significant delay [76,77]. The AIoT paradigm fundamentally breaks this centralized mold by embedding intelligence at various points throughout the network. An AIoT device might itself run a machine learning model—for example, a smart camera that performs on-board object recognition to detect intruders and sends only a high-level alert, rather than continuously streaming high-resolution video. Alternatively, intermediate IoT gateways or edge servers might aggregate and locally analyze data from a multitude of simpler devices, such as an AIoT traffic hub that uses data from multiple sensors to optimize traffic signals for an entire city intersection in real time [45]. While AIoT also encompasses powerful cloud AI components for tasks like big data analytics across an entire smart city, the key differentiator is the synergistic and distributed nature of its intelligence. AI algorithms are strategically deployed throughout the IoT continuum—from the microcontroller level up to the cloud—to enable real-time, context-aware, and intelligent behavior [78].

To ensure conceptual clarity, we highlight the following key distinctions:

- **IoT vs. AIoT:** IoT broadly refers to a network of connected physical objects capable of sending and receiving data. IoT alone does not imply advanced data analysis; many legacy IoT systems were designed simply to collect data for offline human interpretation or simple rule-based alarming. AIoT infuses these systems with automated data analysis, learning, and decision-making capabilities via AI. In an AIoT system, IoT devices evolve from being "dumb" data sources into "smart agents" [8].
- **Standalone AI vs. AIoT:** It is possible to have powerful AI systems that are not part of the IoT (e.g., a cloud AI service analyzing financial market data or social media trends). AIoT specifically deals with AI applied within the context of the IoT ecosystem. This means it is fundamentally concerned with processing data from the physical world and operating under the constraints of that world, such as intermittent sensor readings, network bandwidth limitations, and power budgets. Its decisions often directly control physical devices like robots, alarms, or actuators [8].
- **Edge AI vs. AIoT:** Edge AI refers to the execution of AI algorithms on edge devices, which are located closer to the source of data [79]. Edge AI is a critical enabling component of AIoT, but AIoT is a broader concept [80]. AIoT encompasses not only edge AI on devices but also the intelligent coordination and orchestration of tasks between the edge, fog, and cloud layers. AIoT systems often involve a sophisticated, distributed split of AI tasks, rather than having all intelligence reside at the extreme edge.
- **TinyML:** Tiny Machine Learning is a specialized sub-field of AI focused on developing and deploying machine learning models on extremely resource-constrained devices, typically micro-controllers (MCUs). TinyML is a key enabler for the 'on-device' intelligence layer of AIoT. It is one of the core technologies that makes AIoT possible by pushing AI capabilities down to the tiniest, lowest-power IoT nodes [81].

Ultimately, an AIoT system is a self-contained, intelligent ecosystem where data flows from sensors through a network, is processed by AI methods at one or more layers, and culminates in context-specific actions or insights, often without requiring direct human intervention for each individual decision.

 doi:10.20944/preprints202604.0261.v1

5.2. Building on Foundational Architectures: From Fog Computing to an AI-Centric Framework

The concept of a three-tiered architecture spanning the device, the network edge, and the centralized cloud is not new; it is a cornerstone of modern distributed computing. Foundational work by Bonomi et al. on Fog Computing first articulated the vision of a 'continuum of computing" that extends from the cloud to the things, introducing an intermediate fog layer to handle latency-sensitive and geographically distributed applications [82]. This vision was later formalized through standardized definitions, such as the NIST definition of fog computing, which established clear architectural principles for this distributed paradigm [83]. Similarly, the concept of Multi-Access Edge Computing (MEC), championed by Satyanarayanan and others, emphasizes placing computational resources at the edge of the network (e.g., at cellular base stations) to provide ultra-low latency and high-bandwidth services [84,85].

These paradigms provide the essential structural blueprint for distributing computation. They establish the 'what" (the layers: Cloud, Fog/Edge, Device) and the general 'why" (to reduce latency, save bandwidth, and improve reliability). However, these foundational models are application-agnostic and do not offer a prescriptive framework for the specific, nuanced demands of distributing Artificial Intelligence workloads [86]. AI tasks are not monolithic; they encompass a wide spectrum of operations—from lightweight, real-time inference on a microcontroller to massive, parallelized model training in a data center.

This paper builds directly upon this foundational work. We do not propose a new physical architecture; rather, we introduce a logical framework—the EFC Distributed Intelligence Model—that specializes the EFC architecture for the AIoT context [25,76]. The primary contribution of our model is a taxonomy that maps specific classes of AI and ML tasks to the most appropriate architectural tier based on a multi-dimensional analysis of latency, privacy, power, and model complexity. It transitions the discourse from general computation offloading to a structured methodology for intelligent AI workload placement, thereby addressing a critical gap between established distributed computing architectures and the practical needs of modern AIoT deployments.

5.3. The Edge–Fog–Cloud (EFC) Distributed Intelligence Model

Building on these foundational paradigms, our analysis requires a specialized framework to conceptualize precisely how AI tasks should be distributed. To this end, we formalize the Edge-Fog-Cloud (EFC) Distributed Intelligence Model, illustrated in Figure 2. This model provides a high-level, layered blueprint of an AIoT system with multiple computational tiers, each with distinct characteristics and a specific role in processing data and generating intelligence [32,82].

The EFC model conceptualizes the AIoT ecosystem as a three-layer continuum:

- **The Edge Layer (On-Device Intelligence):** This is the outermost layer, consisting of the IoT endpoint devices themselves (sensors, actuators, wearables, smart appliances). In the AIoT paradigm, these devices can perform lightweight inference locally using TinyML and embedded ML techniques [75]. For instance, a smart security camera might run a person-detection neural network on-device to determine if a human is present; only the high-level event (or a cropped image of the person) is sent upstream, rather than the continuous raw video stream. Edge devices typically have very limited computational resources (e.g., a few MHz CPU or a small neural accelerator) and operate under a strict power budget (< 1 W).
- **The Fog Layer (Intermediate Intelligence):** The fog layer consists of one or more intermediate nodes situated between the edge devices and the central cloud. These nodes could be IoT gateways in a smart factory, local servers within a building, or edge computing servers co-located with cellular base stations (often termed Multi-Access Edge Computing or MEC servers). Fog nodes are significantly more powerful than individual IoT devices (often having multi-core processors or GPUs) and can aggregate data from dozens or hundreds of edge devices [82]. In our model, fog nodes are responsible for medium-scale analytics, data pre-processing, or real-time control of a local cluster of devices.

 doi:10.20944/preprints202604.0261.v1

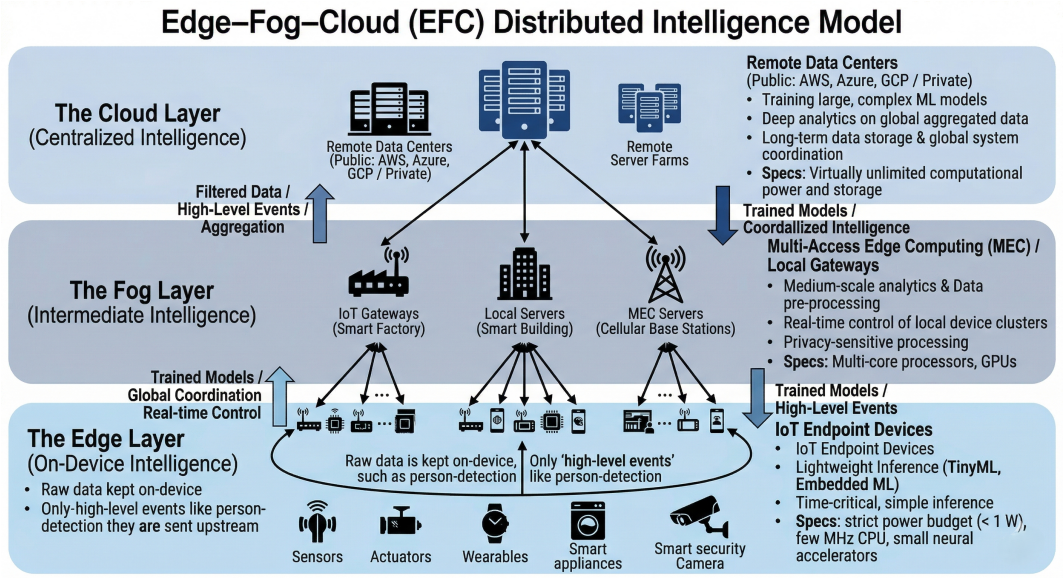

Figure 2. The Edge–Fog–Cloud (EFC) computing continuum for distributed AIoT intelligence. This model illustrates the hierarchical distribution of computational resources and AI tasks from the device level to the central cloud.

- **The Cloud Layer (Centralized Intelligence):** The cloud refers to remote data center infrastructure, whether it be a public cloud (like AWS, Azure, GCP) or a central private server farm. This layer offers virtually unlimited computational power and storage and is responsible for the heavy lifting in an AIoT system. Its primary roles include training large, complex ML models, performing deep analytics on aggregated data from across the entire network, and providing long-term data storage and global system coordination.

A key distinction often made in the literature, which we adopt here, is between "edge" and "fog". While sometimes used interchangeably, "edge computing" typically refers to computation happening at the very periphery of the network, either on the end-device itself or on a gateway immediately adjacent to it. "Fog computing" describes a more distributed and hierarchical intermediate layer of computation that can span from local gateways to regional micro-data centers.

In a well-architected AIoT system using the EFC model, each layer handles the tasks most appropriate for its capabilities, creating a synergistic flow of data and intelligence. Time-critical, simple inference tasks happen at the Edge. Intermediate aggregation, real-time control of a local cluster of devices, or privacy-sensitive processing happens in the Fog. And heavy-duty processing, long-term learning, and global coordination across the entire system happen in the Cloud [25]. This distributed approach effectively mitigates the weaknesses of any single layer operating in isolation.

5.4. Federated Learning (FL): A Key Paradigm for Distributed Training

A critical concept for distributed intelligence is Federated Learning (FL). FL is a decentralized machine learning training approach where the raw training data remains on the end-user devices, thereby preserving privacy [87]. Instead of centralizing data in the cloud, a global model is trained collaboratively. The process typically involves a central server sending a model to a set of devices. Each device then trains the model locally on its own data for a few iterations and sends only the updated model parameters (weights and biases) back to the server. The server aggregates these updates (e.g., by averaging them) to create an improved global model, and the process repeats.

This approach inherently enhances privacy and can significantly reduce bandwidth usage compared to uploading vast amounts of raw data. However, the practical application of FL in the wild, heterogeneous environments of AIoT is fraught with significant challenges. Issues of statistical data imbalance (non-IID data), varying device capabilities, unreliable network connectivity, and security vulnerabilities introduce complexities far beyond the theoretical model [88]. We introduce FL here as a

 doi:10.20944/preprints202604.0261.v1

foundational concept and will provide a detailed critical analysis of its core challenges and potential mitigation strategies in Section 6.

The EFC model and the concept of federated learning provide the foundational mental map for designing modern AIoT systems. They encourage architects to move beyond a cloud-centric default and instead ask the crucial question: "Where should this intelligence live?" for each specific function. The upcoming sections will use this model as a reference to analyze the challenges and compare the architectural trade-offs in detail.

6. Cross-Domain Challenges in Deploying AIoT

Deploying distributed AI in real-world IoT environments is a complex, multi-faceted endeavor. Our systematic review of the literature reveals a set of core challenges that consistently arise across the diverse application domains of IIoT, smart cities, healthcare, and agriculture [76]. This section discusses each of these challenges in depth, organized into key thematic areas: the fundamental resource constraints that necessitate distributed intelligence, the operational complexities of managing distributed systems, and the overarching need for AIoT systems to be secure, private, and trustworthy [78].

6.1. Fundamental Resource Constraints: Power and Computation

One of the most fundamental challenges in AIoT stems from a basic contradiction: we are asking IoT devices, which are often severely resource-constrained, to perform computational tasks (AI inference) that were traditionally reserved for powerful servers. A typical IoT edge device—such as a battery-powered environmental sensor, a medical wearable, or a microcontroller-based industrial controller—possesses a tiny CPU (with clock speeds in the tens or hundreds of MHz), very limited memory (often on the order of tens to hundreds of kilobytes of RAM), and operates under an extremely strict power budget, sometimes needing to run for years on a single coin-cell battery [32,75].

Running sophisticated AI algorithms on such devices pushes the absolute limits of what is currently feasible. To put this in concrete terms, a standard deep learning model for image recognition (like ResNet-50) might be over 50 MB in size and require billions of floating-point operations (GFLOPs) for a single inference—an impossible task for a device with only 128 KB of RAM and no dedicated floating-point unit [89]. Even much smaller models designed for tasks like anomaly detection or predictive analytics might require more memory or CPU cycles than are available on a low-cost microcontroller.

Power constraints are equally, if not more, critical. A sensor node might have a total power budget of only a few milliwatts, whereas an unconstrained CPU or GPU in a server can consume hundreds of watts. AI computations, if not meticulously optimized, can drain batteries with alarming speed. For instance, continuously running even a small neural network on a wearable device could slash its battery life from weeks to mere hours, rendering the device impractical for its intended use [90,91].

These constraints are pervasive across all domains:

- **In IIoT**, many sensors, such as vibration sensors on motors or temperature sensors on equipment, are based on low-power microcontrollers. Deploying a machine learning model for real-time anomaly detection directly on each sensor is challenging when the microcontroller might only have 64 KB of flash memory for code and a few kilobytes of RAM. Yet, this local inference is highly desirable to detect issues immediately at the source [75,77].
- **In Smart Agriculture**, devices are often solar-powered and may operate offline for extended periods. A soil sensor that uses on-board AI to decide when to trigger an irrigation valve must be extraordinarily frugal with its energy consumption to ensure reliable operation [75,92].
- **In Connected Healthcare**, any additional processing on a wearable device translates directly to increased battery usage and heat generation, which is highly undesirable for user comfort and safety. A smartwatch running an AI model to detect atrial fibrillation must do so without significantly reducing the time between charges [75,93].

Solutions and Mitigations: Overcoming these resource constraints requires a multi-pronged approach involving innovations in algorithms, hardware, and system design:

 doi:10.20944/preprints202604.0261.v1

- **Model Compression and TinyML Techniques:** A vibrant area of research is focused on compressing and optimizing AI models to fit on microcontrollers. Key techniques include:
  - **Quantization:** Reducing the precision of model weights and activations from 32-bit floating-point numbers to 8-bit or even 4-bit integers. This dramatically reduces memory footprint and can accelerate computation on simple integer-based hardware.
  - **Pruning:** Systematically removing redundant neurons or filters from a trained neural network that have minimal impact on its accuracy, thereby reducing model size and computational complexity.
  - **Knowledge Distillation:** Training a small, compact "student" model to mimic the behavior of a larger, more complex "teacher" model, effectively transferring the knowledge into a more efficient form factor.
  - **Neural Architecture Search (NAS):** Using automated algorithms to discover novel neural network architectures that are optimized for specific hardware constraints, such as latency or memory usage [90,91].
- **Efficient Edge Hardware:** The semiconductor industry is responding with a proliferation of specialized AI accelerators for low-power devices, including novel architectures like analog in-memory computing and event-driven neuromorphic chips that are inspired by the brain's efficiency.
- **Duty Cycling and Event-Driven Compute:** Many IoT devices can remain in a deep-sleep, low-power state for most of their operational life, waking to run an inference only when triggered by a simple threshold or external event. This ensures that computationally expensive tasks are run only intermittently, conserving energy.

6.2. Managing the Data Deluge

The "data deluge" from IoT devices presents another critical challenge: how to efficiently transport, store, and process the sheer volume of data that AIoT systems generate. As noted earlier, global IoT devices could produce on the order of 80–90 zettabytes per year. Pushing all of this raw data to a centralized cloud is often technically infeasible and economically unsustainable, due to both network bandwidth limitations and the high cost of cloud storage and processing [32].

Bandwidth Constraints: Many IoT devices connect over wireless links that have inherently limited bandwidth, such as LoRaWAN, NB-IoT, or cellular connections in remote areas [94]. Even in environments with high-speed wired connections, the aggregate traffic from thousands of devices can strain the network infrastructure. A classic example is a smart city with thousands of CCTV cameras: continuously streaming dozens of high-definition video feeds 24/7 to a central cloud for analysis would tax even a city's fiber optic network and incur massive cloud compute costs for video analytics [45].

Storage and Processing Costs: Even if sufficient bandwidth were available, storing petabytes or exabytes of raw sensor data in a central repository is impractical. The cost of cloud storage at that scale is enormous, and more importantly, a significant portion of the raw data is low-value noise that does not need to be saved. The challenge lies in intelligently identifying what data is valuable and ensuring it can be processed in a timely manner to extract insights [32,82].

Solutions and Mitigations:

- **Edge/Fog Data Filtering and Aggregation:** The primary motivation for moving intelligence to the edge or fog is to perform data reduction near the source. By analyzing data on-site, devices or local gateways can filter out noise, extract relevant features, and transmit only high-level insights or compact summaries upstream.
- **Data-Centric AI Pipelines:** Adopting a data-centric approach, where the quality and management of data are prioritized, is crucial. This involves implementing robust pipelines at the edge for cleaning, filtering, and normalizing sensor data before it is used for inference or transmitted [19].

 doi:10.20944/preprints202604.0261.v1

- **Distributed Storage & Local Analytics:** Not all data needs to be centralized. Fog nodes or localized data stores can retain recent, high-resolution data for immediate local analysis and discard it when it is no longer needed, while only forwarding long-term trends or critical events to the cloud.

6.3. Interoperability and Standardization

The IoT ecosystem is notoriously fragmented. There are thousands of hardware manufacturers, each with their own device interfaces, a myriad of communication protocols (Wi-Fi, Bluetooth, Zigbee, LoRa, 5G, etc.), countless data formats, and numerous proprietary platforms [95,96]. When AI is introduced into this mix, this fragmentation can severely hinder the integration, scalability, and maintenance of AIoT systems.

Interoperability—the ability of diverse systems and devices to work together and exchange information seamlessly—is a critical challenge. The lack of interoperability has been termed the "Balkanization" of IoT, resulting in many small, isolated islands of connected devices rather than a single, cohesive, intelligent network.

The Challenge: For an AIoT system to perform optimally, it is often necessary to aggregate and fuse data from multiple, diverse sources. If each device and platform "speaks" a different language, integration becomes a complex and costly custom engineering project for every new deployment. This not only increases development time and cost but also results in fragile, brittle integrations between subsystems. This lack of interoperability is particularly problematic for AI, as machine learning models thrive on data volume and variety. If data remains siloed in separate, incompatible systems, it is impossible to train models that can glean insights across those silos. It also complicates model deployment: an AI model developed for predictive maintenance should ideally be deployable on hardware from different vendors, which is difficult without standard edge compute interfaces [76].

Solutions and Mitigations: The industry is actively working to address these issues through several initiatives:

- **Standard Protocols and Data Models:** Promoting the use of open, standard protocols for communication (e.g., MQTT, CoAP) and data modeling (e.g., OPC UA in industry).
- **Middleware and Platforms:** Using IoT platforms (e.g., AWS IoT, Azure IoT, EdgeX Foundry) that act as a common integration layer, abstracting away the differences between underlying devices.
- **Standardized Model Formats:** Using interoperable model formats like ONNX (Open Neural Network Exchange) allows models to be trained in one framework (e.g., PyTorch) and deployed on a different runtime or hardware accelerator.

6.4. MLOps for the Edge: Managing the Model Lifecycle

When AI models are deployed in a distributed edge or fog environment, the classical challenges of Machine Learning Operations (MLOps) are compounded by scale, heterogeneity, and intermittent connectivity [25]. MLOps extends the principles of DevOps to the entire AI lifecycle, encompassing data ingestion, model training, deployment, monitoring, and updating. Managing this lifecycle for potentially thousands of heterogeneous, geographically dispersed, and intermittently connected devices requires a new level of operational excellence [39].

Key challenges include:

- **Deployment at Scale:** How can a new model version be efficiently and reliably rolled out to hundreds or thousands of devices? How can a rapid rollback be performed if a bug or performance degradation is detected?
- **Managing Heterogeneity:** Deployed models may need slightly different builds for different hardware targets (e.g., a TensorRT-optimized version for an NVIDIA Jetson, a TFLite Micro version for a Cortex-M microcontroller, an ONNX version for a CPU-only gateway). Managing these multiple, targeted builds and ensuring their consistency is a complex task.

 doi:10.20944/preprints202604.0261.v1

- **Monitoring and Drift Detection:** In a cloud environment, monitoring model performance is straightforward. At the edge, getting feedback is much harder. Monitoring systems must be designed to send back summary statistics or occasional samples without negating the bandwidth and privacy gains of edge computing. This is crucial for detecting concept drift (when the underlying statistical properties of the data change over time) and initiating model retraining.
- **Continuous Improvement:** IoT models often need frequent updates as more data is collected or as the physical environment changes. Automating the entire pipeline—from data collection and labeling to retraining and redeployment—is essential for maintaining model accuracy and relevance.

6.5. Security, Privacy, and Trust

As AI and IoT converge, the security and privacy risks expand dramatically, evolving beyond the traditional challenges of securing connected devices to encompass the integrity of the AI models themselves.

The Expanded and Intelligent Attack Surface: The core challenge is a shift from attacking simple data endpoints to attacking autonomous, intelligent agents. The attack surface expands through both massive scale and new, intelligent vulnerabilities:

- **Model Theft and Integrity Attacks:** Malicious actors may attempt to steal proprietary ML models from edge devices or tamper with them to cause misbehavior.
- **Adversarial Machine Learning:** This involves crafting malicious physical-world inputs (e.g., a specially designed sticker on a stop sign) designed to fool AI models, a critical concern for autonomous systems like self-driving cars or security cameras [97].
- **Privacy Inference Attacks:** Attackers may be able to analyze model outputs or updates (as in federated learning) to infer sensitive information about the private data used to train the model [98].

Privacy in an Inferred World: AI compounds privacy challenges, as the inferences drawn from data (e.g., determining home occupancy patterns from smart meter data) can be more revealing and sensitive than the raw data itself. Regulations like GDPR and HIPAA mandate a "privacy-by-design" approach, requiring data minimization and user consent.

Trust and Explainability (XAI): For systems that make autonomous decisions, particularly in high-stakes environments, trust is both a social and a technical necessity. This involves reliability (robustness against failures), accountability (clear frameworks for determining responsibility when things go wrong), and, critically, explainability. If an autonomous system makes a critical decision (e.g., shutting down a factory line), operators need to understand why. Black-box models are often insufficient in these environments. As highlighted by recent surveys on the topic [27], there is a growing demand for XAI techniques tailored for IoT that can make AIoT systems more transparent, debuggable, and trustworthy to their human collaborators.

Challenges in Distributed Learning and Federation While Federated Learning (FL) and other distributed learning techniques offer a promising paradigm for privacy-preserving AIoT, moving from theory to robust, large-scale deployment reveals several formidable challenges [99]. Simply keeping data local is not enough; the inherent heterogeneity and constraints of AIoT ecosystems introduce significant practical hurdles that can compromise model performance, fairness, and efficiency [100].

- **Statistical Heterogeneity (Non-IID Data):** This is arguably the most significant challenge in AIoT. The data collected on each device is typically not an independent and identically distributed (IID) sample of the overall data distribution. For example, a motion sensor in a busy office will have a vastly different data signature than one in a quiet home. This 'client drift" causes the local model updates to pull the global model in conflicting directions, which can lead to slow convergence, oscillations, or a complete failure to train an effective global model. **Mitigation Strategies:** Research has focused on algorithms that accommodate this heterogeneity. FedProx, for instance, introduces a proximal term to the local objective function, which limits how far local

 doi:10.20944/preprints202604.0261.v1

models can drift from the global model during training. Another emerging area is Personalized FL, which accepts data heterogeneity and aims to train not one single global model, but a base model that can be quickly and effectively personalized for each device's specific data distribution.

- **Systems Heterogeneity:** The AIoT landscape is characterized by a massive diversity of devices. A federated network may include powerful edge gateways, battery-powered microcontrollers, and smartphones, each with vastly different computational power (CPU/RAM), energy budgets, and network connectivity (5G vs. LoRaWAN). This creates two problems:
  - **Stragglers:** Slower devices can significantly delay training rounds if the server uses a synchronous aggregation scheme (waiting for all clients).
  - **Participation Bias:** If only the most powerful devices can consistently participate in training, the final model will be biased towards the data from this privileged subset of devices.

  Mitigation Strategies: Asynchronous FL protocols have been developed where the central server aggregates updates as they arrive, rather than waiting for a complete cohort. Furthermore, client selection algorithms can be used to intelligently select a subset of available and capable clients for each round, although this must be done carefully to avoid introducing bias. Frameworks like Flower and TensorFlow Federated provide mechanisms to manage such asynchronous communication and device sampling.
- **Communication Bottlenecks:** While FL avoids sending raw data, the model updates themselves can still be large (megabytes for deep neural networks). Transmitting these updates over constrained, and often costly, IoT networks (e.g., cellular) from thousands or millions of devices can be a major bottleneck, draining battery life and congesting the network. **Mitigation Strategies:** Significant research is dedicated to model update compression. Techniques include quantization, where updates are converted from 32-bit floating-point numbers to more compact 8-bit integers, and sparsification, where only the most significant weight changes are transmitted, often using methods like structured updates or sketching. These techniques can reduce the communication payload by an order of magnitude or more.
- **Privacy and Security Beyond Data Locality:** The core assumption that 'data never leaves the device" is an insufficient guarantee of privacy. The model updates, though anonymized, are not inert; they are artifacts of the training data [88].
  - **Adversarial Attacks:** Malicious actors with access to model updates can potentially perform inference attacks to deduce sensitive information about a user's private data or even reconstruct training samples.

  Mitigation Strategies: Two key technologies address this. Differential Privacy involves adding carefully calibrated statistical noise to the model updates before they are shared. This provides a mathematical guarantee that the presence or absence of any single user's data has a negligible effect on the final model, thus protecting individual privacy. Secure Aggregation uses cryptographic techniques like secure multi-party computation, allowing the server to compute the sum of all model updates without being able to decrypt any individual update, providing protection even from a malicious or curious server [88].

Addressing these challenges is the central focus of modern FL research and is a prerequisite for its successful application in real-world, large-scale AIoT systems [100].

7. Comparative Analysis of AIoT Architectures

In this section, we provide a detailed analysis and comparison of the main architectural approaches for deploying AI within IoT systems, using the Edge-Fog-Cloud model as our reference framework. The objective is to quantitatively and qualitatively evaluate where intelligence should reside—on the device (TinyML), at an intermediate gateway (Fog/Edge Computing), or in the cloud—and to understand how these architectural choices impact key performance and operational metrics [75]. We

 doi:10.20944/preprints202604.0261.v1

also consider hybrid architectures, including those that leverage federated learning for distributed training [100].

Our analysis follows a structured data-method-insight flow. First, we precisely define the metrics used for our comparison (e.g., inference latency, device power consumption, model footprint). Second, we present a comprehensive summary table (Table 2) that characterizes each architecture with typical, cited performance ranges for these metrics, synthesizing data from numerous benchmark studies [78]. Finally, we provide a quantitative comparison, visualized in Figure 3, using representative data from the literature to highlight the critical trade-offs between latency, energy consumption, and model complexity [101].

7.1. Definition of Comparison Metrics

To ensure a clear and consistent comparison, we define the following key metrics:

- **Inference Latency**: The end-to-end time elapsed from the moment input data is captured by a sensor to the moment the AI model's output (prediction or decision) is available. This critically includes network communication delays if the model is not executed on-device. It is a paramount factor for any real-time application.
- **Device Power Usage**: The energy consumed by the end-device to either perform the AI task locally or offload it to another tier. For battery-powered devices, this is a primary design constraint. We focus on the energy consumed per inference (typically measured in millijoules, mJ), as this metric effectively normalizes for duty-cycled or event-driven operation.
- **Model Size / Complexity**: The memory footprint (e.g., in kilobytes or megabytes) and computational requirements (e.g., in MAC operations) of the AI model. This directly dictates the class of hardware required for deployment and is a major constraint for on-device execution.
- **Data Privacy**: A qualitative assessment of the architecture's ability to protect sensitive user or operational data. Architectures that process data locally without transmitting it to external servers inherently offer stronger privacy guarantees.
- **Scalability**: The ability of the architecture to handle a growing number of devices, an increasing data volume, and greater model complexity. This considers potential bottlenecks in computation, network bandwidth, and operational management.
- **Resilience & Offline Capability**: The ability of the system to continue functioning, at least at a basic level, in the event of a network outage or loss of connectivity to the cloud.

7.2. Architectural Characterization

Table 2 provides a detailed comparison of the quantitative and qualitative characteristics of the primary AIoT architectures. The data presented is a synthesis derived from a wide range of recent benchmark studies and technical literature.

Table 2. Evidence-Based Comparison of AIoT Architectures

Architecture	Inference Latency	Device Power Usage (per inference)	Model Size / Complexity	Data Privacy	Scalability
On-Device (TinyML)	$<$ 30 ms (Real-time, ideal for immediate action)	$\sim$0.1–1 mJ (Ultra-low, enables battery operation for years)	$<$ 500 KB (Highly constrained, simple models)	Excellent (Raw data never leaves the device)	High (Device): Scales with devices; Low (Ops): Model updates are complex
Edge Gateway Computing	30–100 ms (Low latency, includes one network hop)	Device: 1–10 mJ (Transmission cost); Gateway: Wall-powered	1–50 MB (Moderate models, e.g., MobileNet)	Good (Data remains on the local network)	Medium: Scales by adding gateways, which can become local bottlenecks

 doi:10.20944/preprints202604.0261.v1

Table 2. *Cont.*

Architecture	Inference Latency	Device Power Usage (per inference)	Model Size / Complexity	Data Privacy	Scalability
Cloud-Centric	$>$ 150 ms (High latency, dominated by network round-trip)	$>$ 10 mJ (High transmission cost for device)	$>$ 100 MB (Very large, state-of-the-art models)	Poor (Raw data sent to third-party servers)	High (Compute): Scales elastically; Low (Network): Bandwidth bottleneck
Federated Learning (Training)	N/A (Training approach)	High (During local training; device compute is significant)	Partial models (Train large models collaboratively)	Very Good (Only model updates are shared)	Medium: Complex coordination and communication overhead

7.3. Quantitative Trade-Off Analysis

To illustrate the practical implications of these architectural differences, we analyze a representative AI inference task (e.g., image classification on a 224x224 image) across different deployment strategies. Figure 3 visualizes the critical trade-offs between inference latency and device energy consumption based on typical benchmark values reported in the literature for such a task.

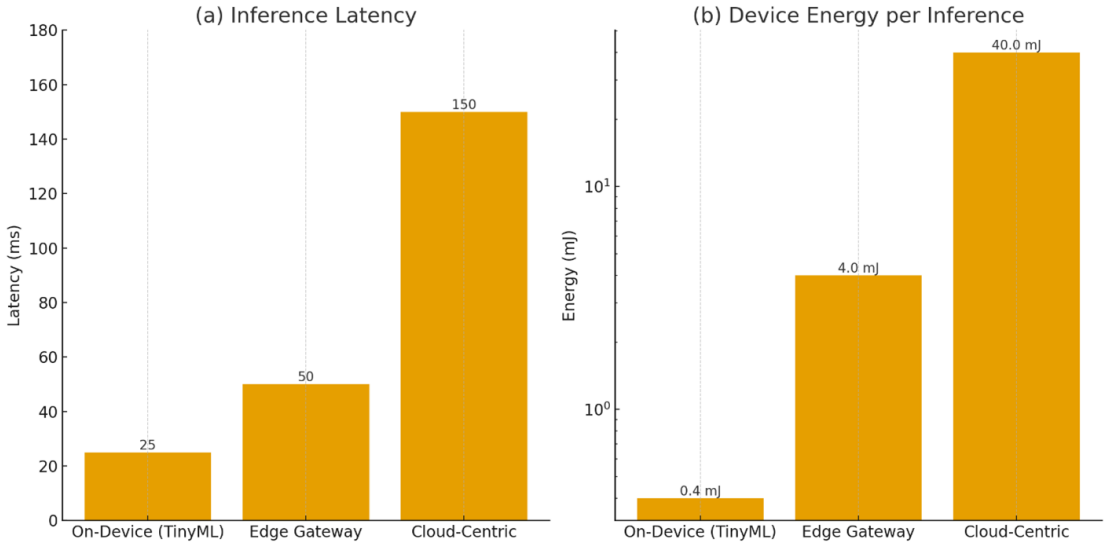

Figure 3. Quantitative Comparison of Latency and Energy per Inference. The charts illustrate the performance of different AIoT architectures for a sample image classification task. (a) On-Device (TinyML) inference latency is lowest, often under 30 ms, while cloud latency is an order of magnitude higher ($>$ 150 ms) due to network delay. (b) Device energy consumption is minimized with On-Device processing. Offloading to the cloud requires significant energy for wireless data transmission, which can be 10–100$\times$ more than local computation for frequent, high-volume data.

The data confirms a clear and fundamental trade-off. On-Device (TinyML) architectures provide unparalleled low latency and exceptional power efficiency, making them the only viable option for many battery-powered, real-time applications such as wearable health monitoring or immediate keyword detection in a smart speaker. However, this extreme efficiency comes at the cost of model complexity; models must typically have a memory footprint of less than 500 KB, which limits their application to simpler tasks like classification or anomaly detection on low-dimensional data [75].

Conversely, Cloud-Centric architectures offer maximum analytical power. They are capable of running enormous, state-of-the-art models that are indispensable for tasks requiring deep historical analysis, the fusion of data from thousands of sources, or the training of models on massive, aggregated

 doi:10.20944/preprints202604.0261.v1

datasets. The unavoidable costs are high latency, typically exceeding 150 ms due to network round-trip times, and a critical reliance on network connectivity. This renders the pure-cloud approach unsuitable for safety-critical systems or real-time control loops. Furthermore, the energy required to transmit raw data—especially high-volume data like images or video—from the device to the cloud is often far greater than the energy required for local processing, making it highly inefficient for battery-powered devices with high-frequency data streams [32,101].

Edge and Fog solutions present a compelling and often optimal middle ground. By processing data on a more powerful local gateway, they significantly reduce latency compared to the cloud (typically in the 30–100 ms range) and keep sensitive data within the confines of the local network, thereby enhancing privacy and security. This architecture is ideal for applications like factory automation, smart building management, or intelligent traffic control at city intersections, where multiple sensors feed into a moderately complex model for localized, real-time decision-making that is beyond the capability of a single microcontroller but does not require the global scope of the cloud [78,102].

This detailed analysis reinforces the central argument of this paper: that hybrid, multi-layered architectures are essential. A pure-cloud or pure-edge approach is rarely optimal. An effective AIoT system often employs a tiered strategy: preliminary data filtering and immediate, low-latency alerts happen on-device (TinyML); more complex decisions and data aggregation for a local area occur at the edge; and long-term analytics, global optimization, and heavy model training are reserved for the cloud. The intelligent interplay between these layers, guided by the quantitative trade-offs identified here, is critical for designing robust, efficient, and truly intelligent systems [76].

8. Software Ecosystem for Distributed AIoT: Runtimes, Platforms, and Orchestration

Beyond architectural patterns, successful AIoT deployment depends on a software stack that spans model execution, hardware-specific optimization, embedded MLOps, fleet management, and edge orchestration. A major editorial weakness in many survey manuscripts is that these layers are treated as if they were interchangeable. They are not. A microcontroller inference runtime, an accelerator SDK, a TinyML development platform, and an edge-orchestration framework solve different problems and should therefore be evaluated against different criteria. To avoid this category slippage, this section organizes the software ecosystem into five layers: inference runtimes, hardware-optimized SDKs, TinyML development platforms, cloud–edge deployment services, and edge middleware/orchestration. This structure aligns the software discussion with the paper's central Edge–Fog–Cloud (EFC) model and makes framework selection more actionable for practitioners [25,78]. Because commercial capabilities evolve quickly, the discussion emphasizes stable architectural roles and documented capabilities rather than transient product-level features [124,125,127,131].

8.1. Inference Runtimes and Model-Execution Layers

TensorFlow Lite and TensorFlow Lite for Microcontrollers

TensorFlow Lite (TFLite) remains one of the most widely used deployment runtimes for resource-constrained and mobile edge inference because it supports model conversion, post-training quantization, and hardware-backed acceleration across a broad device spectrum [104,119]. Within AIoT, its relevance is strongest at the device and near-edge tiers, where developers need a lightweight inference runtime rather than a full training stack. For Linux-class edge devices, TFLite is appropriate for single-board computers and embedded Linux systems that must execute compact vision, audio, or sensor models with modest memory budgets.

TensorFlow Lite for Microcontrollers (TFLite Micro) addresses a more constrained target class: bare-metal or RTOS-based microcontrollers. Official documentation emphasizes three properties that are particularly important in AIoT deployments: static memory planning, the absence of dynamic memory allocation, and the ability to run without operating-system support [119]. These characteristics make TFLite Micro well suited to always-on sensing tasks such as keyword spotting, vibration-based

 doi:10.20944/preprints202604.0261.v1

anomaly detection, and simple event classification on MCU-class devices. Editorially, however, it should not be described as a universal solution. Its strengths are determinism and low footprint, but those come with operator limitations, restricted model complexity, and a development workflow that is most effective when the final model architecture is intentionally designed for constrained inference from the outset [30,104].

PyTorch Edge Tooling and ExecuTorch

PyTorch has historically been stronger in research and training workflows than in deeply embedded inference, but its on-device ecosystem has matured substantially. ExecuTorch is now documented by the PyTorch project as an on-device inference solution spanning mobile phones, wearables, embedded devices, and microcontrollers [120]. For AIoT, this is important because many edge-AI pipelines are prototyped in PyTorch and benefit from a deployment path that does not require a complete retraining or a framework migration.

Even so, the role of ExecuTorch in AIoT should be presented carefully. It is best characterized as a promising and increasingly capable deployment path for PyTorch-native workflows, especially when portability across mobile and embedded edge devices matters. Compared with TFLite Micro, its major advantage is alignment with the broader PyTorch ecosystem; compared with ONNX Runtime, its advantage is tighter integration with PyTorch-authored models. Its limitation is that, in deeply resource-constrained TinyML settings, the surrounding tooling and long-term deployment patterns are not yet as entrenched in the literature as the TFLite Micro workflow [104,120].

ONNX Runtime

ONNX Runtime occupies a different position in the stack. Rather than primarily targeting the smallest MCUs, it functions as a cross-platform model-execution layer for heterogeneous edge and fog systems. Official documentation describes it as a cross-platform machine-learning model accelerator with support for models originating from PyTorch, TensorFlow/Keras, TFLite, scikit-learn, and other frameworks, as well as an extensible execution-provider mechanism for hardware acceleration [121,122]. This makes ONNX Runtime especially valuable in industrial or city-scale AIoT deployments where interoperability and hardware heterogeneity matter more than absolute minimum footprint.

In practice, ONNX Runtime is best suited to gateways, industrial PCs, accelerator-equipped endpoints, and Linux-class edge servers that must support multiple model families and heterogeneous silicon. Its main advantage is portability across frameworks and hardware backends. Its main limitation is that it is not a substitute for ultra-small MCU runtimes; it serves the edge/fog portion of the EFC continuum better than the extreme embedded tier [78,121].

8.2. Hardware-Optimized SDKs and Acceleration Layers

A second category comprises acceleration libraries and hardware-specific deployment SDKs. These tools should not be conflated with full AIoT platforms: they optimize inference on specific processor families or accelerators, but they do not by themselves solve fleet management, model versioning, interoperability, or orchestration.

ARM CMSIS-NN

CMSIS-NN provides optimized neural-network kernels for Arm Cortex-M CPUs and is a foundational building block for MCU-class AI. The original CMSIS-NN paper reported substantial improvements in runtime and energy efficiency relative to less optimized baselines on Cortex-M devices [123]. In AIoT practice, CMSIS-NN is particularly important because runtimes such as TFLite Micro can use it as a backend to accelerate integer inference on Arm microcontrollers. Its role is therefore highly specific but highly valuable: it is an enabling optimization layer for on-device intelligence at the extreme edge.

 doi:10.20944/preprints202604.0261.v1

NVIDIA TensorRT

TensorRT is an SDK for optimizing and accelerating deep-learning inference on NVIDIA GPUs, with support for mixed precision, dynamic shapes, and model-level optimizations that target low latency and high throughput [124]. Within the EFC model, TensorRT is not a device-layer technology; it is primarily a fog/edge-server optimization stack for GPU-equipped platforms such as Jetson-class devices and higher-end edge servers. It is especially relevant for computer-vision-heavy IIoT, smart-city video analytics, and edge robotics workloads. The limitation is portability: the optimization benefits are strong, but they are coupled to the NVIDIA deployment ecosystem.

Intel OpenVINO

OpenVINO serves a similar function for Intel-centric edge deployment. Official documentation positions it as an open-source toolkit for developing and deploying performant AI solutions in the cloud, on-premises, and at the edge, with conversion and optimization support for Intel hardware [125]. In AIoT terms, OpenVINO is best suited to edge gateways, industrial PCs, and vision-oriented deployments in factories, transport systems, or smart facilities that rely on Intel CPUs, integrated GPUs, or VPUs. Its comparative strength is deployment efficiency across Intel hardware; its limitation, again, is that it is best understood as a hardware-optimized execution layer rather than a complete IoT lifecycle platform.

8.3. TinyML Development and Embedded MLOps Platforms

Edge Impulse

Edge Impulse should be treated separately from inference runtimes because it addresses a broader problem: end-to-end TinyML development and embedded MLOps. The MLSys paper on Edge Impulse presents it as a cloud-based MLOps platform for developing embedded and edge ML systems across heterogeneous hardware targets [126]. Its importance in AIoT stems from workflow integration. Data acquisition, labeling, signal-processing blocks, model training, evaluation, and code generation are brought into a single environment, which reduces friction for multidisciplinary teams that include domain engineers rather than only ML specialists.

For academic review purposes, the key analytical point is that Edge Impulse is not primarily a new runtime. Instead, it is a workflow abstraction layer that can target runtimes such as TFLite Micro while standardizing much of the design cycle for embedded ML. It is therefore particularly useful for rapid prototyping, educational settings, and industrial proof-of-concept development. Its limitation is that highly customized production pipelines may still require direct control over lower-level runtimes, toolchains, and deployment infrastructure [104,126].

8.4. Cloud–Edge Deployment Services and Fleet Management

At larger deployment scales, the central problem shifts from single-model execution to lifecycle management across fleets of devices. This includes software packaging, remote deployment, configuration, rollback, security policy enforcement, and monitoring. These capabilities are essential for AIoT, but they are often absent from runtime-centered discussions.

AWS IoT Greengrass

AWS IoT Greengrass is documented by AWS as an open-source IoT edge runtime and cloud service for building, deploying, and managing device software, including local action on device data, machine-learning inference, and data filtering/aggregation [127,128]. In the EFC model, Greengrass is best understood as a fog/edge deployment and management layer. It is particularly well matched to distributed environments where devices must continue acting locally while remaining under centralized lifecycle management from the cloud.

From a review perspective, Greengrass is most valuable when the paper discusses operationalization rather than raw inference efficiency. Its strengths are modular deployment, integration with upstream AWS services, and fleet-level software management. Its limitation is that it is not in-

 doi:10.20944/preprints202604.0261.v1

tended for bare-metal MCU inference; it assumes a more capable runtime host at the gateway or edge-compute layer.

Azure IoT Edge

Azure IoT Edge plays a comparable role in the Microsoft ecosystem. Official documentation describes it as an extension of IoT Hub that allows workloads to run locally, respond quickly, and continue operating offline, with containerized modules as the fundamental deployable unit [131,132]. This makes Azure IoT Edge analytically relevant as a cloud–edge lifecycle platform rather than merely a model-serving tool.

Its major strength is the use of containerized modules, which cleanly separate business logic, AI inference services, protocol translation, and supporting services. That modularity is especially helpful in heterogeneous AIoT deployments where multiple analytics and integration services coexist on the same gateway. As with Greengrass, its limitation is deployment footprint: it is best suited to Linux-class edge nodes and gateways, not MCU-only endpoints.

8.5. Edge Middleware and Orchestration

KubeEdge

KubeEdge extends Kubernetes-style orchestration to edge environments and is documented as an open-source system for extending native containerized application orchestration capabilities to hosts at the edge [134]. The original architecture paper positions it as a Kubernetes-based approach for managing cloud and edge application deployment through a unified API model [133]. In AIoT, KubeEdge is relevant where the operational challenge is not only model execution but multi-service orchestration across geographically distributed edge nodes.

KubeEdge is therefore most appropriate for complex edge estates: smart-city zones, multi-site industrial installations, or distributed surveillance/vision systems that require container orchestration, cloud-edge coordination, and partial offline operation. Its main trade-off is operational complexity. The additional abstraction is valuable at scale, but it can be excessive for small, single-purpose embedded deployments [106,134].

EdgeX Foundry

EdgeX Foundry belongs in this section because its core value is interoperability middleware at the IoT edge. Official project documentation and LF Edge materials describe it as a vendor-neutral, open framework for IoT edge computing that provides modular services for device ingestion, normalization, security, and data exchange between heterogeneous southbound devices and northbound applications [135,136]. This fills an important gap in the original manuscript: interoperability platforms mentioned earlier in the paper should also appear in the software-ecosystem analysis.

EdgeX Foundry is not an AI runtime in the narrow sense. Its importance for AIoT lies in making heterogeneous device data accessible to AI services in a structured and manageable way. In other words, it is often the middleware substrate on top of which edge analytics or AI services are deployed. Its strength is interoperability; its limitation is that it does not replace the need for separate inference runtimes or model-optimization toolchains.

8.6. Comparative Synthesis and Selection Guidance

Table 3 reorganizes the software ecosystem into comparable categories and maps each tool family to the most appropriate EFC tier, architectural role, and deployment scenario.

From this comparison, several selection rules emerge. For *ultra-constrained on-device inference*, TFLite Micro combined with optimized kernels such as CMSIS-NN remains the most appropriate choice. For *PyTorch-centric edge development*, ExecuTorch provides a more direct path to on-device deployment. For *heterogeneous edge gateways and industrial PCs*, ONNX Runtime offers the strongest portability story. For *accelerator-centric edge inference*, TensorRT and OpenVINO are the more relevant choices because they explicitly target NVIDIA and Intel optimization paths, respectively. For *workflow*

 doi:10.20944/preprints202604.0261.v1

simplification in embedded ML, Edge Impulse is strongest as an integrated TinyML development platform. For *fleet-level lifecycle management*, Greengrass and Azure IoT Edge become more important than runtime-level differences. Finally, for *large-scale edge estates and heterogeneous integration*, KubeEdge and EdgeX Foundry address orchestration and interoperability challenges that inference runtimes alone cannot solve.

Table 3. Decision-oriented comparison of AIoT software frameworks, platforms, and middleware.

Tool / Family	Software Layer	Best-fit EFC Tier	Primary Strengths	Main Limitations / Best-fit Scenarios
TFLite / TFLite Micro	Inference runtime	Device / Edge	Compact inference runtime, quantization support, static-memory MCU deployment, strong TinyML ecosystem alignment [104,119]	Best for MCU and low-power embedded inference; limited model complexity and operator coverage relative to larger runtimes. Ideal for keyword spotting, anomaly detection, and simple sensor classification.
ExecuTorch	Inference runtime	Device / Edge	PyTorch-aligned on-device inference across mobile, embedded, and microcontroller targets [120]	Strong portability for PyTorch workflows, but less entrenched in TinyML literature than TFLite Micro. Best for teams standardizing on PyTorch across mobile and embedded edge.
ONNX Runtime	Inference runtime / portability layer	Edge / Fog	Cross-framework interoperability and accelerator integration through execution providers [121,122]	Better suited to Linux-class gateways and accelerator-backed systems than to bare-metal MCUs. Ideal for heterogeneous industrial and edge-server deployments.
CMSIS-NN	Hardware-optimized kernel library	Device	Highly efficient integer kernels for Arm Cortex-M, improving throughput and energy efficiency [123]	Optimization layer rather than full platform. Best when combined with MCU runtimes such as TFLite Micro.
TensorRT	Hardware-optimized SDK	Fog / Edge server	High-performance GPU inference, mixed precision, low latency, strong vision and transformer optimization on NVIDIA hardware [124]	NVIDIA-centric. Best for Jetson-class and GPU-equipped edge servers in vision-heavy AIoT workloads.
OpenVINO	Hardware-optimized SDK	Edge / Fog	Model conversion and deployment optimization across Intel CPUs, GPUs, and VPUs [125]	Intel-centric. Best for industrial PCs, smart cameras, and Intel-based edge gateways.
Edge Impulse	Embedded MLOps platform	Device / Edge	Integrated data acquisition, labeling, signal processing, training, and code generation for TinyML workflows [126]	Excellent for rapid prototyping and embedded ML teams; customized large-scale production pipelines may still require lower-level toolchain control.
AWS IoT Greengrass	Fleet management / edge runtime	Edge / Fog	Modular deployment, local action, cloud integration, data filtering, and ML component management [127,128]	Not for MCU-only endpoints. Best for gateway-centric deployments that require fleet management and cloud coordination.
Azure IoT Edge	Fleet management / edge runtime	Edge / Fog	Containerized modules, offline operation, integration with IoT Hub and Azure ML workflows [131,132]	Best for Linux-class edge nodes and modular gateway software stacks; not intended for deeply constrained MCU inference.
KubeEdge	Orchestration framework	Fog / Edge server	Kubernetes-native cloud–edge orchestration, unified management, distributed application deployment [133,134]	Powerful at scale but operationally heavier than single-purpose deployment stacks. Best for multi-site or multi-service edge estates.
EdgeX Foundry	Interoperability middleware	Edge / Fog	Vendor-neutral device integration, normalization, security, and northbound/southbound service abstraction [135,136]	Middleware rather than inference runtime. Best where heterogeneous devices must feed analytics and AI services through a common integration layer.

The central insight is that no single framework solves the entire AIoT software problem. The software ecosystem, like the hardware architecture, is inherently layered. Effective AIoT systems therefore emerge not from selecting one "best" platform, but from composing complementary layers that match the target EFC tier, operational scale, and interoperability requirements of the deployment [25,76]

9. Practical Deployment: Hardware, Tooling, and Case Studies

Translating the architectural principles and software frameworks discussed in previous sections into robust, real-world deployments requires pragmatic decisions about hardware, tooling, and system

 doi:10.20944/preprints202604.0261.v1

design. This section focuses on providing actionable guidance for practitioners. We first present a structured decision framework for AIoT workload placement—a practical guide to help engineers determine the optimal location for AI tasks (on-device vs. edge vs. cloud) based on specific application requirements and constraints [76]. We then provide a detailed survey of the key hardware platforms that enable AIoT, discussing the roles and trade-offs of microcontroller units (MCUs), single-board computers (SBCs), and specialized AI accelerators. Finally, we illustrate these concepts with concrete case studies from our target domains, demonstrating how the framework and tools are applied in practice.

9.1. Decision Framework for AI Workload Placement

Designing an effective AIoT system begins with the fundamental question: Where should the intelligence reside? The answer is rarely a simple one-size-fits-all solution. The optimal placement of an AI workload is a multi-dimensional optimization problem. We propose the decision framework, visualized as a flowchart in Figure 4, to guide this critical process.

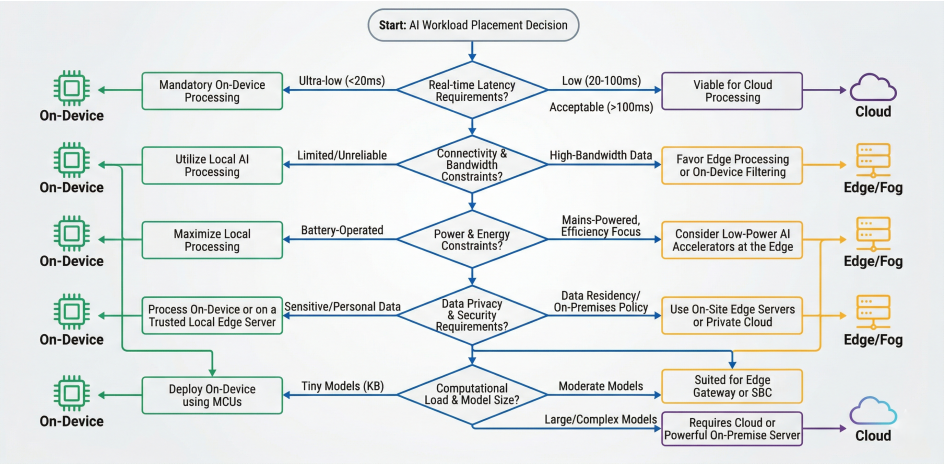

Figure 4. A Decision Flowchart for AI Workload Placement in AIoT Systems. This framework guides developers through a series of critical questions to determine the optimal architectural tier for an AI task.

The framework involves asking a series of questions about the application's requirements:

1. **What are the real-time latency requirements?**
   - If ultra-low latency (< 20 ms) and deterministic responsiveness are required for safety-critical functions or real-time control, **on-device processing is mandatory**. *Example: A collision avoidance sensor in an industrial robot must react in milliseconds to prevent accidents.*
   - If low latency ($20 - 100$ ms) is needed for near-real-time applications, an **edge/fog node is suitable**. *Example: Adjusting traffic lights based on video analysis of traffic flow at an intersection can tolerate a 50 ms delay.*
   - If latencies of hundreds of milliseconds to seconds are acceptable, **cloud processing is viable**. *Example: Generating a daily report on a building's energy consumption.*
2. **What are the connectivity and bandwidth constraints?**
   - If devices are in remote locations with limited, unreliable, or expensive connectivity (e.g., cellular data in a rural farm), minimize data transmission. **Utilize local AI** to process data and send only high-level insights or alerts.
   - If high-bandwidth sensor data (e.g., video, high-frequency audio) is involved and continuous streaming is impractical or cost-prohibitive, **favor edge processing or on-device filtering**. *Example: A security camera performs on-device motion detection and only streams video to the cloud when an alert is triggered.*
3. **What are the power and energy constraints?**

 doi:10.20944/preprints202604.0261.v1

- For battery-operated or energy-harvesting devices, minimize power consumption. Since wireless transmission is often the most power-hungry operation, **perform as much local processing as possible**. *Example: A wearable ECG monitor runs a lightweight anomaly detection model on-device and only transmits data when an anomaly score exceeds a threshold.*
- Mains-powered devices have more flexibility but still benefit from energy-efficient design. Consider if using a low-power AI accelerator can improve the performance-per-watt for your model.

4. **What are the data privacy and security requirements?**
   - If personal, sensitive, or proprietary data is being processed (e.g., health metrics, facial images, audio from private spaces), it should ideally be **processed on-device or on a trusted local edge server** to keep the raw data localized.
   - If corporate policy or data residency laws require data to remain on-premises, **use on-site edge servers or a private cloud** at the fog layer.
5. **What is the computational load and model size?**
   - Tiny models (a few kilobytes in size, requiring minimal operations) can easily be deployed **on-device using MCUs**.
   - Moderately complex models (e.g., MobileNet-SSD for object detection) that require a CPU or a small GPU are well-suited for an **edge gateway or SBC**.
   - Very large and complex models (e.g., high-resolution video analytics, large language models) require the resources of the **cloud or a powerful on-premise server**. In these cases, consider a hierarchical approach: a smaller "filter" model at the edge can decide when to send data to the larger cloud model for more detailed analysis.

By systematically working through this framework, designers can allocate each AI workload to the most appropriate tier, often resulting in a hybrid, multi-tiered solution.

9.2. Hardware Platforms: Devices, Gateways, and Accelerators

The choice of hardware is intrinsically linked to the workload placement decision. The AIoT hardware ecosystem is diverse and rapidly evolving.

Microcontroller Units (MCUs) for On-Device AI

MCUs are the workhorses of the IoT, found in billions of embedded devices. They are designed for low cost and ultra-low power consumption.

- **Examples:** The ARM Cortex-M series (M4, M7, M33) and RISC-V equivalents are common in IoT sensors and wearables. The ESP32 from Espressif is another popular choice, offering integrated Wi-Fi and Bluetooth.
- **Capabilities:** Traditionally, MCUs had no specific hardware for ML. However, they can run small neural networks efficiently using optimized libraries like ARM's CMSIS-NN. A new generation of MCUs is emerging with built-in micro-NPUs (Neural Processing Units) or DSP extensions specifically designed to accelerate ML operations, providing a significant performance boost for always-on vision or audio tasks with minimal power draw.
- **Trade-offs:** MCUs are extremely power-efficient and low-cost but are severely limited in memory (typically 256KB–2MB of flash and 64KB–512KB of RAM) and computational power. They are ideal for simple classification, keyword spotting, or anomaly detection tasks.

Single-Board Computers (SBCs) and Edge Gateways

SBCs are essentially small, self-contained computers on a single circuit board, often running a full Linux operating system. They serve as powerful edge gateways or controllers.

- **Examples:**

 doi:10.20944/preprints202604.0261.v1

  - **Raspberry Pi:** A popular, low-cost SBC used widely in prototyping and some production deployments.
  - **NVIDIA Jetson Series (Nano, Orin):** These SBCs include a powerful integrated GPU with CUDA cores, making them ideal for real-time computer vision and other parallel-processing AI tasks.
  - **Google Coral Dev Board:** This board features Google's Edge TPU, a dedicated ASIC designed to accelerate TensorFlow Lite models with very high performance-per-watt.
- **Role:** They often act as aggregators, collecting data from multiple downstream MCU-based sensors, or as standalone processing units for a single high-bandwidth sensor like a camera. Their ability to run a full OS makes development significantly easier than on MCUs.

AI Accelerators and Co-Processors

These are specialized chips or modules designed for one purpose: to execute AI inference tasks with maximum speed and energy efficiency.

- **Examples:** Google's Edge TPU, Intel's Movidius VPU, and chips from companies like Hailo and Kneron. These accelerators can be integrated into custom hardware designs or accessed via plug-in modules (e.g., the Google Coral USB Accelerator).
- **Advantage:** The key advantage is a massive improvement in inference speed per watt, which can be an order of magnitude or more better than a general-purpose CPU. The main challenge is that they often require a specific toolchain and model format (e.g., the Edge TPU requires 8-bit quantized TFLite models).

9.3. Case Studies in Practice

Case Study 1: Industrial Predictive Maintenance (IIoT) A manufacturing plant wants to monitor the health of hundreds of critical motors on its production lines.

- **Workload Placement Decision:**
  - **Latency:** An alert about an impending failure must be near-instantaneous (<100 ms) to allow for a controlled shutdown.
  - **Connectivity:** Wi-Fi coverage on the factory floor is spotty.
  - **Data:** Each motor is equipped with a high-frequency vibration sensor generating a constant stream of data. Streaming this all to the cloud is infeasible.
- **Implementation:**
  - **On-Device (MCU):** A low-power MCU with an accelerometer is attached to each motor. It runs a tiny, quantized anomaly detection model (e.g., an autoencoder trained with TFLite Micro). The MCU continuously analyzes the vibration signature and sends an alert only when an anomaly is detected.
  - **Edge Gateway (SBC):** A Raspberry Pi or an industrial PC is installed for each group of 10-15 motors. It receives alerts from the MCUs and can run a more complex model (e.g., an LSTM) to classify the type of fault and predict the remaining useful life.
  - **Cloud:** The cloud dashboard receives only the final alerts and predictions from the edge gateways. It is used for long-term fleet-wide analysis, scheduling maintenance, and periodically retraining the models.

Case Study 2: Remote Patient Monitoring (Connected Healthcare) A healthcare provider wants to continuously monitor elderly patients living at home for falls.

- **Workload Placement Decision:**
  - **Latency:** A fall detection alert must be immediate to ensure a fast response.
  - **Privacy:** The system may use a camera, so sending continuous video to the cloud is a major privacy concern.
  - **Power:** The device should be unobtrusive and not require constant charging.

 doi:10.20944/preprints202604.0261.v1

- **Implementation:**
  - **Edge Device (with Accelerator):** A smart camera with an integrated AI accelerator (like an Intel Movidius VPU or a Google Edge TPU) is placed in the patient's home. The entire fall detection model (e.g., a pose estimation model) runs directly on the device. The video stream never leaves the device.
  - **On-Device Logic:** If a fall is detected, the device sends a secure, encrypted alert (not video) over the internet to an emergency response service and designated caregivers.
  - **Cloud:** The cloud's role is limited to managing the device subscription, receiving the alerts, and routing them to the correct contacts. No sensitive patient video is ever stored or processed in the cloud.

These case studies illustrate that effective AIoT deployment requires a thoughtful combination of hardware at each tier, coupled with a structured decision process to place AI workloads where they can deliver the most value while respecting system constraints.

10. Open Challenges and Future Directions

Despite the significant progress in AIoT, the field is still in its ascendancy, with numerous research and engineering challenges remaining on the horizon [76]. This section explores forward-looking topics that are expected to define the next generation of AIoT systems. Each subsection outlines a key open challenge, explains its importance, and poses specific research questions that will drive future work [77]. The themes include fostering deeper human-AI collaboration, advancing sensor fusion, developing self-sustaining AIoT devices, enabling truly decentralized learning at scale, and realizing the ambitious vision of the Tactile Internet.

10.1. Human-in-the-Loop and Collaborative AIoT

As AI systems become more autonomous and are deployed in complex, real-world environments, a critical challenge is how to keep humans meaningfully in the loop. The goal is to evolve beyond simple automated alerts towards a paradigm of true collaboration between humans and AI agents in shared physical spaces [107]. In a smart factory, for instance, rather than an AI system unilaterally shutting down a machine upon detecting an anomaly (which could disrupt production), a collaborative system might present its diagnosis and confidence score to a human operator, suggesting several courses of action and incorporating the operator's expert feedback.

A primary challenge is designing intuitive and effective interfaces and interaction models for AIoT. How should a distributed AIoT system communicate its state, reasoning, and intentions to a human supervisor? Simply presenting raw sensor data or black-box predictions is insufficient. This necessitates significant research into Explainable AI (XAI) for IoT. For example, an AIoT predictive maintenance system should be able to highlight which specific sensor readings or temporal patterns led it to predict a machine failure, allowing an engineer to verify the reasoning and build trust in the system [108].

Another key research area is adaptive autonomy, where the level of AI autonomy dynamically varies depending on the context. In routine, well-understood situations, the AI can act automatically. In novel, uncertain, or high-risk scenarios, it should gracefully defer to human judgment. Creating algorithms that can accurately self-assess their own uncertainty and know when to seek human input is a major open problem.

Ultimately, the future lies in "symbiotic AIoT"—systems where humans and intelligent devices leverage their complementary strengths. Humans provide contextual understanding, common sense, creativity, and ethical judgment; AI provides speed, precision, and the ability to process vast amounts of sensor data. Bridging this gap will require advances in user experience design, interactive machine learning, and a deeper understanding of the social and trust dynamics between humans and autonomous systems.

 doi:10.20944/preprints202604.0261.v1

10.2. Next-Generation Multi-Modal Sensor Fusion

AIoT systems are increasingly being deployed with heterogeneous sensor arrays, combining visual cameras with LiDAR, radar, acoustic sensors, vibration sensors, temperature arrays, and even biochemical sensors. The grand challenge is to develop AI models that can intelligently and efficiently fuse data from these multiple modalities in real time, especially at the resource-constrained edge [109].

Multi-modal sensor fusion can dramatically improve the robustness and contextual awareness of an AIoT system, but it is notoriously difficult. Different sensors have different data rates, formats, noise characteristics, and can sometimes provide conflicting information [110]. Current practice often relies on relatively simple "late fusion" techniques, where each sensor stream is processed independently and the results are combined at a high level (e.g., through an ensemble vote).

Future research is aimed at deep fusion, where raw or low-level data streams are combined early in the processing pipeline, allowing the AI model to learn complex, cross-modal representations. A key research direction is the use of attention mechanisms for sensor fusion. These are models that can learn to dynamically weigh which sensor modality is more informative or reliable at any given moment. For instance, in an autonomous vehicle, an AI model could learn to pay more attention to LiDAR data in foggy conditions when visual camera data is degraded. Another frontier is developing models that are robust to sensor failure, where the system can continue to operate effectively even if one or more sensor streams become unavailable.

10.3. Energy-Harvesting and Self-Sustaining AIoT

A truly transformative vision for AIoT is the deployment of "deploy-and-forget" intelligent devices—sensors with on-board AI that can operate perpetually without battery replacements by harvesting energy from their environment (e.g., from solar, thermal gradients, RF signals, or vibrations). This would enable the instrumentation of the physical world at a massive scale—in agricultural fields, within civil infrastructure, or even inside the human body—without the logistical nightmare of maintaining and replacing billions of batteries.

While energy-harvesting sensors exist today, running AI on them presents a profound challenge because the available power is extremely limited (microwatts to milliwatts) and often intermittent [111]. This requires a fundamental rethinking of both hardware and software. Research is focusing on ultra-low-power computation and energy-aware hardware–software co-design, including:

- **Analog and Mixed-Signal Computing**: Performing neural network operations in the analog domain to avoid power-hungry digital circuits [112].
- **Event-Driven and Neuromorphic Computing**: Designing sensors and processors that, like the brain, only consume power when there is new information or an "event" to process. Spiking Neural Networks (SNNs) are a key area of research in this domain.
- **Intermittent Computing**: Designing algorithms and hardware that can save their state during a power outage and gracefully resume computation when energy becomes available again.

The ultimate goal is to create a new class of AIoT devices that can learn and adapt over their entire lifespan, powered solely by the ambient energy in their environment.

10.4. The Next Frontier in Distributed and Federated Training

While our focus has primarily been on inference, the ultimate goal of a truly intelligent edge is to move the training process itself to the network periphery. As detailed in Section 5.6, significant research is maturing to address the foundational challenges of statistical heterogeneity, systems constraints, and communication bottlenecks in Federated Learning. The next frontier of research builds upon these mitigations to envision truly decentralized and robust learning ecosystems.

One promising direction is fully decentralized or serverless FL, which removes the reliance on a central coordinating server—a potential single point of failure and control. In these architectures, devices share model updates directly with their peers using gossip-based learning protocols. This

 doi:10.20944/preprints202604.0261.v1

is highly relevant for applications like vehicle-to-vehicle networks, where cars could collaboratively learn a model of local road conditions without a central cloud service.

Another major research thrust is enhancing the robustness and trustworthiness of the learning process. This includes using blockchain and other distributed ledger technologies to create a transparent and auditable record of model contributions, preventing malicious actors from poisoning the global model. Furthermore, as the privacy guarantees of basic FL are now better understood, future work will focus on making advanced techniques like differential privacy and secure aggregation more computationally efficient and scalable for resource-constrained IoT devices.

10.5. The Tactile Internet and Ultra-Low-Latency AIoT

Looking further ahead, the concept of the Tactile Internet has emerged as a key vision for 5G and beyond networks. It aims to enable real-time, interactive systems where the end-to-end latency is so low (on the order of 1 ms) that haptic feedback and remote control feel instantaneous and immersive [70]. For AIoT, this means integrating intelligent control loops with extremely stringent latency and reliability requirements.

A classic example is remote surgery, where a surgeon in one location controls a robotic instrument in another. For the surgeon to feel as if they are "present," the control commands and sensory feedback (including haptics) must have a round-trip time of less than a few milliseconds. Achieving this is not just a networking challenge; any AI-driven processing in the loop must also be executed in microseconds. This will likely mandate specialized hardware, such as FPGA or ASIC implementations of AI algorithms, and tight co-design between the network and the edge computing resources.

Another use case is augmented reality with haptics, where a firefighter wearing an AR helmet and haptic gloves could receive real-time AI analysis of a building's structural integrity, with physical cues (vibrations, resistance) guiding them away from danger. This requires a seamless, sub-millisecond pipeline from sensors to AI analysis to actuation.

Realizing the Tactile Internet will require research into:

- Sub-millisecond AI decision-making: This may require moving beyond traditional software and implementing critical AI components directly in hardware.
- Predictive AI to mitigate latency: AI models that can accurately predict the next few milliseconds of an environment's state could be used to mask any residual network jitter.
- Network-AI co-design: The network itself could use AI to predict congestion and proactively reroute critical tactile data streams to guarantee ultra-low latency.

The Tactile Internet is an ambitious vision, but it represents the ultimate convergence of AI, IoT, and high-performance networking, pushing AIoT to its absolute limits of performance and reliability.

11. Scope, Limitations, and Future Inquiry

As with any comprehensive survey of a broad and rapidly evolving field, it is important to explicitly acknowledge the scope and limitations of this work, and to transparently identify areas that were beyond our coverage or that warrant further dedicated investigation [75,114].

This review focused on the paradigm of distributed intelligence within four specific domains: IIoT, Smart Cities, Connected Healthcare, and Smart Agriculture. While we selected these domains as being highly representative of critical application areas, other important sectors such as consumer IoT (smart homes), autonomous vehicles, and military/defense IoT were not explicitly covered in the same depth [45]. The challenges and solutions in AIoT may manifest differently in those contexts. For example, defense applications may prioritize security and resilience above all else, accepting costs that would be prohibitive in civilian domains.

Furthermore, given the exceptionally rapid pace of innovation in this field, specific technologies, hardware platforms, and software versions mentioned in this review may become obsolete in a relatively short time frame [115]. We have endeavored to focus on the underlying conceptual frameworks

 doi:10.20944/preprints202604.0261.v1

and comparative insights that will remain relevant despite specific product changes. Nevertheless, this review should be seen as a snapshot of a fast-moving landscape as of early 2026.

Another limitation is our primary focus on the technical challenges of AIoT. Broader socio-economic factors, such as user acceptance, the need for workforce retraining, evolving regulatory environments, and sustainable business models, are crucial for the real-world adoption of these technologies but were largely outside the scope of this computer-science-centric review [116,117]. While we touched upon ethical and trust issues, a full treatment of the legal and regulatory aspects (e.g., the impact of GDPR on AIoT data strategies or questions of liability in autonomous systems) was not possible here [118].

Building on these limitations, we suggest several directions for future inquiry:

- **Standardized Benchmarking for AIoT:** There is a pressing need for standardized benchmarks, similar to MLPerf, but specifically designed to evaluate AIoT systems under realistic device, network, and power constraints. This would allow for more rigorous and fair comparisons of different hardware and software solutions.
- **In-depth, Real-world Deployment Studies:** While we have covered frameworks and case anecdotes, there is a scarcity of detailed, longitudinal case studies of large-scale AIoT deployments in the academic literature. Such studies, documenting not only successes but also failures and unexpected challenges, would be invaluable for the community.
- **Socio-Economic and Ethical Analyses:** Future interdisciplinary research is needed to explore the broader societal impacts of AIoT, including its effects on labor markets, privacy norms, and social equity.

12. Conclusions

The convergence of Artificial Intelligence and the Internet of Things—the Artificial Intelligence of Things (AIoT)—represents a fundamental and transformative shift in how we design, deploy, and interact with computing systems embedded in the physical world. Throughout this comprehensive review, we have argued that neither legacy cloud-centric AI nor isolated, non-intelligent IoT devices alone are sufficient to meet the demands of modern, real-time, and scalable applications. Instead, a paradigm of distributed intelligence, strategically partitioned across the edge, fog, and cloud continuum, is required to achieve the low latency, high resilience, robust privacy, and autonomous operation that emerging use cases in industry, cities, healthcare, and agriculture demand.

We began by quantifying the powerful drivers of this convergence: the exponential growth of connected devices generating unprecedented volumes of data and the clear economic imperative for greater responsiveness and automation at scale. We demonstrated how traditional, siloed architectures falter under this load, with purely cloud-based systems introducing untenable latencies and network bottlenecks, while purely local systems lack the capacity for global insight and advanced, large-scale analytics.

This review introduced a systematic framework for distributing intelligence in AIoT, built upon the established Edge–Fog–Cloud (EFC) paradigm. The core of this contribution is a detailed taxonomy that articulates how AI tasks can be strategically partitioned: immediate perception and action on resource-constrained devices (TinyML) to ensure speed and privacy; intermediate data aggregation and inference at fog and edge nodes to balance computational load and achieve local contextual awareness; and large-scale, complex learning and global coordination in the cloud. We also presented a comprehensive, cross-domain taxonomy of the key challenges—from energy efficiency and data management to interoperability and MLOps—and surveyed current solutions and promising research directions for addressing each.

A central theme of our analysis is that security, privacy, and trust are not afterthoughts but are foundational design objectives. Failures in these areas can undermine even the most technologically sophisticated AIoT systems. Our practitioner-oriented analysis of hardware platforms and software tools revealed a rapidly maturing ecosystem that is increasingly capable of supporting the EFC

 doi:10.20944/preprints202604.0261.v1

model. From microcontrollers running neural networks on microwatts of power to edge gateways equipped with powerful AI accelerators, the hardware is evolving to support distributed intelligence. Concurrently, software frameworks for embedded inference, federated learning, and cloud-edge orchestration are maturing to manage the inherent complexity of these distributed deployments.

The future of AIoT is both immensely promising and profoundly challenging. We envision a future of ambient intelligence, where the distinction between the physical and digital worlds blurs, and our environments are populated by intelligent systems that learn, adapt, and collaborate. Realizing this vision will require concerted effort to address the significant open challenges we have outlined, including enhancing human-AI collaboration, advancing multi-modal sensor fusion, achieving sustainable energy autonomy for devices, and enabling collective learning without centralized control.

The transformation toward distributed intelligence will not be immediate. It will demand sustained research innovation, rigorous engineering practices, and cross-sector collaboration involving technologists, policymakers, ethicists, and end-users. Nevertheless, the trajectory is clear: intelligence is progressively and inexorably migrating from centralized clouds toward pervasive, embedded systems.

Distributed intelligence in AIoT constitutes a cornerstone of the next generation of cyber-physical systems. This survey has mapped the current state of the art, identified persistent challenges, and outlined potential pathways toward practical, robust solutions. When AI is deployed as a network of collaborative, layered components rather than as a monolithic, centralized entity, IoT systems become more responsive, resilient, secure, and context-aware. The coming years will be pivotal as these concepts are translated from research into large-scale deployments that will reshape our industries and our world.

Author Contributions: Conceptualization, L.A.P.O. and A.D.C.S.; methodology, L.A.P.O.; software, L.A.P.O.; validation, L.A.P.O., A.D.C.S., and B.G.L.C.; formal analysis, L.A.P.O.; investigation, L.A.P.O.; resources, A.D.C.S.; data curation, L.A.P.O.; writing—original draft preparation, L.A.P.O.; writing—review and editing, L.A.P.O., A.D.C.S., and B.G.L.C.; visualization, L.A.P.O.; supervision, A.D.C.S; project administration, L.A.P.O. All authors have read and agreed to the published version of the manuscript.

Funding: The APC was funded by Escuela Politécnica Nacional.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article. All sources analyzed are publicly available and are cited within the text and reference list.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AGV	Autonomous Guided Vehicle
AI	Artificial Intelligence
AIoT	Artificial Intelligence of Things
DSP	Digital Signal Processor
EFC	Edge–Fog–Cloud
FL	Federated Learning
IID	Independent and Identically Distributed
IIoT	Industrial IoT
IoMT	Internet of Medical Things
IoT	Internet of Things
MCU	Microcontroller Unit
MEC	Multi-Access Edge Computing
mJ	Millijoules
ML	Machine Learning
MLOps	Machine Learning Operations
NAS	Neural Architecture Search

 doi:10.20944/preprints202604.0261.v1

NPU	Neural Processing Unit
ONNX	Open Neural Network Exchange
OTA	Over-the-Air
RTOS	Real-Time Operating System
SBC	Single-Board Computer
SDK	Software Development Kit
TFLite	TensorFlow Lite
TinyML	Tiny Machine Learning
VPU	Vision Processing Unit
XAI	Explainable AI
ZB	Zettabytes

References

1. Cannizzaro, P. IoT Technologies Are Expected to Soon Generate 80 Zettabytes of Data. *EE Times* **2020**, May 8. https://www.eetimes.com/iot-technologies-are-expected-to-soon-generate-80-zettabytes-of-data/.
2. Shafique, K.; Khawaja, B.; Sabir, F.; Qazi, S.; Mustaqim, M. Internet of Things (IoT) for Next-Generation Smart Systems: A Review of Current Challenges, Future Trends and Prospects for Emerging 5G-IoT Scenarios. *IEEE Access* **2020**, *8*, 23022–23040. https://doi.org/10.1109/ACCESS.2020.2970118.
3. Zikria, Y.; Ali, R.; Afzal, M.; Kim, S. Next-Generation Internet of Things (IoT): Opportunities, Challenges, and Solutions. *Sensors* **2021**, *21*, 1174. https://doi.org/10.3390/s21041174.
4. Ifesinachi, A.; Sodiya, E.; Umoga, U.; Obaigbena, A.; Jacks, B.; Ugwuanyi, E.; Daraojimba, A.; Lottu, O. Current State and Prospects of Edge Computing within the Internet of Things (IoT) Ecosystem. *International Journal of Scientific Research Archive* **2024**. https://doi.org/10.30574/ijsra.2024.11.1.0287.
5. Premsankar, G.; Di Francesco, M.; Taleb, T. Edge Computing for the Internet of Things: A Case Study. *IEEE Internet of Things Journal* **2018**, *5*, 1275–1284. https://doi.org/10.1109/JIOT.2018.2805263.
6. Dallaf, A. Edge Computing in IoT Networks: Enhancing Efficiency, Reducing Latency, and Improving Scalability. *International Journal of Advanced Networking, Monitoring and Controls* **2025**, *10*, 103–115. https://doi.org/10.2478/ijanmc-2025-0009.
7. Carvalho, G.; Cabral, B.; Pereira, V.; Bernardino, J. Edge Computing: Current Trends, Research Challenges and Future Directions. *Computing* **2021**, *103*, 993–1023. https://doi.org/10.1007/s00607-020-00896-5.
8. Zhang, J.; Tao, D. Empowering Things with Intelligence: A Survey of the Progress, Challenges, and Opportunities in Artificial Intelligence of Things. *IEEE Internet of Things Journal* **2020**, *8*, 7789–7817. https://doi.org/10.1109/JIOT.2020.3039359.
9. Chang, Z.; Liu, S.; Xiong, X.; Cai, Z.; Tu, G. A Survey of Recent Advances in Edge-Computing-Powered Artificial Intelligence of Things. *IEEE Internet of Things Journal* **2021**, *8*, 13849–13875. https://doi.org/10.1109/JIOT.2021.3088875.
10. U, I.; Kumaravelu, V.; C, V.; A, R.; Chinnadurai, S.; Venkatesan, R.; Hai, H.; Selvaprabhu, P. AI-Powered IoT: A Survey on Integrating Artificial Intelligence with IoT for Enhanced Security, Efficiency, and Smart Applications. *IEEE Access* **2025**, *13*, 50296–50339. https://doi.org/10.1109/ACCESS.2025.3551750.
11. Adli, H.; Remli, M.; Wong, K.; Ismail, N.; González-Briones, A.; Corchado, J.; Mohamad, M. Recent Advancements and Challenges of AIoT Application in Smart Agriculture: A Review. *Sensors* **2023**, *23*, 3752. https://doi.org/10.3390/s23073752.
12. Andriulo, F.; Fiore, M.; Mongiello, M.; Traversa, E.; Zizzo, V. Edge Computing and Cloud Computing for Internet of Things: A Review. *Informatics* **2024**, *11*, 71. https://doi.org/10.3390/informatics11040071.
13. Yu, W.; Liang, F.; He, X.; Hatcher, W.; Lu, C.; Lin, J.; Yang, X. A Survey on Edge Computing for the Internet of Things. *IEEE Access* **2018**, *6*, 6900–6919. https://doi.org/10.1109/ACCESS.2017.2778504.
14. He, Q.; Xi, Z.; Feng, Z.; Teng, Y.; L.; Cai, Y.; Yu, K. Telemedicine Monitoring System Based on Fog/Edge Computing: A Survey. *IEEE Transactions on Services Computing* **2025**, *18*, 479–498. https://doi.org/10.1109/TSC.2024.3506473.
15. Banoth, S.; M, V.; Punna, H.; P, M.; Prakash, V.; M, J. Edge Computing Architectures for Low-Latency Data Processing in Internet of Things Applications. *ITM Web of Conferences* **2025**, *76*, 03003. https://doi.org/10.1051/itmconf/20257603003.
16. Abouaomar, A.; Cherkaoui, S.; Mlika, Z.; Kobbane, A. Resource Provisioning in Edge Computing for Latency-Sensitive Applications. *IEEE Internet of Things Journal* **2021**, *8*, 11088–11099. https://doi.org/10.1109/JIOT.2021.3052082.

 doi:10.20944/preprints202604.0261.v1

17. Pathak, M.; Mishra, K.; Singh, S. Securing Data and Preserving Privacy in Cloud IoT-Based Technologies: An Analysis of Threats and Effective Safeguards. *Artificial Intelligence Review* **2024**, *57*, 269. https://doi.org/10.1007/s10462-024-10908-x.
18. Pinto, G.; Donta, P.; Dustdar, S.; Prazeres, C. A Systematic Review on Privacy-Aware IoT Personal Data Stores. *Sensors* **2024**, *24*, 2197. https://doi.org/10.3390/s24072197.
19. De Haro-Olmo, F.; Valencia-Parra, Á.; Varela-Vaca, Á.; Álvarez-Bermejo, J.; Gómez-López, M. ELI: An IoT-Aware Big Data Pipeline with Data Curation and Data Quality. *PeerJ Computer Science* **2023**, *9*, e1605. https://doi.org/10.7717/peerj-cs.1605.
20. Singh, S.; Kumar, N.; Kumar, G.; Balusamy, B.; Bashir, A.; Dabel, M. Enhancing Quality of Service in IoT-WSN through Edge-Enabled Multi-Objective Optimization. *IEEE Transactions on Consumer Electronics* **2025**. https://doi.org/10.1109/TCE.2025.3526992.
21. Al-Fuqaha, A.; Guibene, W.; Mohammadi, M.; Aledhari, M.; Ayyash, M. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Communications Surveys & Tutorials* **2015**, *17*, 2347–2376. https://doi.org/10.1109/COMST.2015.2444095.
22. Liu, H.; Galindo, M.; Xie, H.; Wong, L.; Shuai, H.; Li, Y.; Cheng, W. Lightweight Deep Learning for Resource-Constrained Environments: A Survey. *ACM Comput. Surv.* **2024**, *56*, 1–42. https://doi.org/10.1145/3657282.
23. Keivanimehr, A.; Akbari, M. TinyML and Edge Intelligence Applications in Cardiovascular Disease: A Survey. *Computers in Biology and Medicine* **2025**, *186*, 109653. https://doi.org/10.1016/j.compbiomed.2025.109653.
24. Oliveira, F.; Costa, D.; Assis, F.; Silva, I. Internet of Intelligent Things: A Convergence of Embedded Systems, Edge Computing and Machine Learning. *Internet of Things* **2024**, *26*, 101153. https://doi.org/10.1016/j.iot.2024.101153.
25. Liu, S.; Guo, B.; Fang, C.; Wang, Z.; Luo, S.; Zhou, Z.; Yu, Z. Enabling Resource-Efficient AIoT System with Cross-Level Optimization: A Survey. *IEEE Communications Surveys & Tutorials* **2023**, *26*, 389–427. https://doi.org/10.1109/COMST.2023.3319952.
26. Aouedi, O.; Vu, T.; Sacco, A.; Nguyen, D.; Piamrat, K.; Marchetto, G.; Pham, V. A Survey on Intelligent Internet of Things: Applications, Security, Privacy, and Future Directions. *IEEE Communications Surveys & Tutorials* **2024**, *27*, 1238–1292. https://doi.org/10.1109/COMST.2024.3491572.
27. Kök, İ.; Okay, F. Y.; Muyanlı, Ö.; Özdemir, S. Explainable Artificial Intelligence (XAI) for Internet of Things: A Survey. *IEEE Internet of Things Journal* **2023**, *10*, 14764–14779. https://doi.org/10.1109/JIOT.2023.3287678.
28. Zhang, P.; White, J.; Schmidt, D.; Lenz, G.; Rosenbloom, S. FHIRChain: Applying Blockchain to Securely and Scalably Share Clinical Data. *Computational and Structural Biotechnology Journal* **2018**, *16*, 267–278. https://doi.org/10.1016/j.csbj.2018.07.004.
29. Khan, S.; Naseer, M.; Hayat, M.; Zamir, S.; Khan, F.; Shah, M. Transformers in Vision: A Survey. *ACM Computing Surveys* **2021**, *54*, 1–41. https://doi.org/10.1145/3505244.
30. Capogrosso, L.; Cunico, F.; Cheng, D.; Fummi, F.; Cristani, M. A Machine Learning-Oriented Survey on Tiny Machine Learning. *IEEE Access* **2024**, *12*, 23406–23426. https://doi.org/10.1109/ACCESS.2024.3365349.
31. Yang, Y.; Wu, L.; Yin, G.; Li, L.; Zhao, H. A Survey on Security and Privacy Issues in Internet-of-Things. *IEEE Internet of Things Journal* **2017**, *4*, 1250–1258. https://doi.org/10.1109/JIOT.2017.2694844.
32. Shi, W.; Cao, J.; Zhang, Q.; Li, Y.; Xu, L. Edge Computing: Vision and Challenges. *IEEE Internet of Things Journal* **2016**, *3*, 637–646. https://doi.org/10.1109/JIOT.2016.2579198.
33. Sicari, S.; Rizzardi, A.; Grieco, L. A.; Coen-Porisini, A. Security, Privacy and Trust in Internet of Things: The Road Ahead. *Computer Networks* **2015**, *76*, 146–164. https://doi.org/10.1016/j.comnet.2014.11.008.
34. Wang, J.; Ma, Y.; Zhang, L.; Gao, R. X.; Wu, D. Deep Learning for Smart Manufacturing: Methods and Applications. *Journal of Manufacturing Systems* **2021**, *48*, 144–156. https://doi.org/10.1016/j.jmsy.2018.01.003.
35. Zanella, A.; Bui, N.; Castellani, A.; Vangelista, L.; Zorzi, M. Internet of Things for Smart Cities. *IEEE Internet of Things Journal* **2014**, *1*, 22–32. https://doi.org/10.1109/JIOT.2014.2306328.
36. Tariq, M. I.; Memon, N. A.; Ahmed, S.; Tayyaba, S.; Mushtaq, M. T.; Mian, N. A.; Imran, M.; Ashraf, M. W. A Review of Deep Learning Security and Privacy Defensive Techniques. *Mobile Information Systems* **2020**, *2020*, 6535834. https://doi.org/10.1155/2020/6535834.
37. Wolfert, S.; Ge, L.; Verdouw, C.; Bogaardt, M.-J. Big Data in Smart Farming—A Review. *Agricultural Systems* **2017**, *153*, 69–80. https://doi.org/10.1016/j.agsy.2017.01.023.
38. OpenFog Consortium Architecture Working Group. OpenFog Reference Architecture for Fog Computing. Technical report, OpenFog Consortium, 2017. https://www.iiconsortium.org/pdf/OpenFog_Reference_Architecture_2_09_17.pdf.

 doi:10.20944/preprints202604.0261.v1

39. Verma, D.; Santhanam, P. MLOps at the Edge in DDIL Environments. In *Proceedings of SPIE* **2024**, *13051*, 130510V. https://doi.org/10.1117/12.3013300.
40. Cinar, Z.; Nuhu, A.; Zeeshan, Q.; Korhan, O.; Asmael, M.; Safaei, B. Machine Learning in Predictive Maintenance towards Sustainable Smart Manufacturing in Industry 4.0. *Sustainability* **2020**, *12*, 8211. https://doi.org/10.3390/su12198211.
41. Zhang, W.; Yang, D.; Xu, Y.; Huang, X.; Zhang, J.; Gidlund, M. DeepHealth: A Self-Attention-Based Method for Instant Intelligent Predictive Maintenance in Industrial Internet of Things. *IEEE Transactions on Industrial Informatics* **2021**, *17*, 5461–5473. https://doi.org/10.1109/TII.2020.3029551.
42. Resende, C.; Folgado, D.; Oliveira, J.; Franco, B.; Moreira, W.; Oliveira, A.; Cavaleiro, A.; Carvalho, R. TIP4.0: Industrial Internet of Things Platform for Predictive Maintenance. *Sensors* **2021**, *21*, 4676. https://doi.org/10.3390/s21144676.
43. Ayvaz, S.; Alpay, K. Predictive Maintenance System for Production Lines in Manufacturing: A Machine Learning Approach Using IoT Data in Real-Time. *Expert Systems with Applications* **2021**, *173*, 114598. https://doi.org/10.1016/j.eswa.2021.114598.
44. Putteti, S.; Santhi, G.; Mittoor, G.; Nagamani, C.; Udayaraju, P. Intelligent Industrial IoT: A Data-Driven Approach for Smart Manufacturing and Predictive Maintenance. In *Proceedings of the 2025 Third International Conference on Augmented Intelligence and Sustainable Systems (ICAISS)*, 2025; pp. 1032–1040. https://doi.org/10.1109/ICAISS61471.2025.11041978.
45. Alahi, M.; Sukkuea, A.; Tina, F.; Nag, A.; Kurdthongmee, W.; Suwannarat, K.; Mukhopadhyay, S. Integration of IoT-Enabled Technologies and Artificial Intelligence (AI) for Smart City Scenario: Recent Advancements and Future Trends. *Sensors* **2023**, *23*, 5206. https://doi.org/10.3390/s23115206.
46. Lilhore, U.; Imoize, A.; Li, C.; Simaiya, S.; Pani, S.; Goyal, N.; Rana, A.; Lee, C. Design and Implementation of an ML and IoT-Based Adaptive Traffic-Management System for Smart Cities. *Sensors* **2022**, *22*, 2908. https://doi.org/10.3390/s22082908.
47. Puzio, E.; Drożdż, W.; Kolon, M. The Role of Intelligent Transport Systems and Smart Technologies in Urban Traffic Management in Polish Smart Cities. *Energies* **2025**, *18*, 2580. https://doi.org/10.3390/en18102580.
48. Neelakantachari, A.; S, N.; Pujar, R. AI-Enabled IoT for Smart Cities and Infrastructure. *World Journal of Advanced Research and Reviews* **2021**, *9*, 65–78. https://doi.org/10.30574/wjarr.2021.9.2.0065.
49. Jagatheesaperumal, S.; Bibri, S.; Huang, J.; Rajapandian, J.; Parthiban, B. Artificial Intelligence of Things for Smart Cities: Advanced Solutions for Enhancing Transportation Safety. *Computational Urban Science* **2024**, *4*, 15. https://doi.org/10.1007/s43762-024-00120-6.
50. Almaazmi, K.; Almheiri, S.; Khan, M.; Shah, A.; Abbas, S.; Ahmad, M. Enhancing Smart City Sustainability with Explainable Federated Learning for Vehicular Energy Control. *Scientific Reports* **2025**, *15*, 7844. https://doi.org/10.1038/s41598-025-07844-3.
51. Van Hoang, T. Impact of Integrated Artificial Intelligence and Internet of Things Technologies on Smart City Transformation. *J. Tech. Educ. Sci.* **2024**, *19*, 123–135. https://doi.org/10.54644/jte.2024.1532.
52. Silvestri, S.; Tricomi, G.; Bassolillo, S.; De Benedictis, R.; Ciampi, M. An Urban Intelligence Architecture for Heterogeneous Data and Application Integration, Deployment and Orchestration. *Sensors* **2024**, *24*, 2376. https://doi.org/10.3390/s24072376.
53. Gondhalekar, G.; Tewari, D.; P, B.; Bhardwaj, I.; Ponnusamy, S.; S, A. Internet of Things Integration in Smart Cities: Enhancing Urban Living through Connected Technologies. *ITM Web of Conferences* **2025**, *76*, 03001. https://doi.org/10.1051/itmconf/20257603001.
54. Khan, L.; Yaqoob, I.; Tran, N.; Kazmi, S.; Tri, N.; Hong, C. Edge-Computing-Enabled Smart Cities: A Comprehensive Survey. *IEEE Internet of Things Journal* **2020**, *7*, 10200–10232. https://doi.org/10.1109/JIOT.2020.2987070.
55. Bibri, E.; Krogstie, J.; Kaboli, A.; Alahi, A. Smarter Eco-Cities and Their Leading-Edge Artificial Intelligence of Things Solutions for Environmental Sustainability: A Comprehensive Systematic Review. *Environmental Science and Ecotechnology* **2023**, *19*, 100330. https://doi.org/10.1016/j.ese.2023.100330.
56. Anthony, B. Enabling Seamless Interoperability of Digital Systems in Smart Cities Using API: A Systematic Literature Review. *J. Urban Technol.* **2024**, *31*, 123–156. https://doi.org/10.1080/10630732.2024.2427543.
57. Bibri, S.; Huang, J. Artificial Intelligence of Things for Sustainable Smart City Brain and Digital Twin Systems: Pioneering Environmental Synergies between Real-Time Management and Predictive Planning. *Environmental Science and Ecotechnology* **2025**, *26*, 100591. https://doi.org/10.1016/j.ese.2025.100591.
58. Huang, C.; Wang, J.; Wang, S.; Zhang, Y. Internet of Medical Things: A Systematic Review. *Neurocomputing* **2023**, *557*, 126719. https://doi.org/10.1016/j.neucom.2023.126719.

 doi:10.20944/preprints202604.0261.v1

59. Ozcelik, M.; Kok, I.; Ozdemir, S. A Survey on Internet of Medical Things (IoMT): Enabling Technologies, Security and Explainability Issues, Challenges, and Future Directions. *Expert Systems* **2025**, *42*, e70010. https://doi.org/10.1111/exsy.70010.
60. Damera, V.; Cheripelli, R.; Putta, N.; Sirisha, G.; Kalavala, D. Enhancing Remote Patient Monitoring with AI-Driven IoMT and Cloud Computing Technologies. *Scientific Reports* **2025**, *15*, 9727. https://doi.org/10.1038/s41598-025-09727-z.
61. Bhattacharya, P.; Mukherjee, A.; Bhushan, B.; Gupta, S.; Gadekallu, T.; Zhu, Z. A Secured Remote Patient Monitoring Framework for IoMT Ecosystems. *Scientific Reports* **2025**, *15*, 4774. https://doi.org/10.1038/s41598-025-04774-y.
62. Alshehri, D.; Noman, N.; Chiong, R.; Miah, S.; Sverdlov, A.; Ngo, D. Factors Influencing the Adoption of Internet of Medical Things for Remote Patient Monitoring: A Systematic Literature Review. *Computers in Biology and Medicine* **2025**, *192*, 110142. https://doi.org/10.1016/j.compbiomed.2025.110142.
63. Fang, W.; Lo, Y. Artificial Intelligence-Enhanced Multi-Lead ECG Monitoring for Early Cardiovascular Disease Detection and Cuffless Blood Pressure Estimation in Smart Healthcare. In *Proceedings of the 2025 IEEE International Conference on Consumer Electronics (ICCE)*, 2025; pp. 1–5. https://doi.org/10.1109/ICCE63647.2025.10929814.
64. Rahman, M.; Morshed, B. Resource-Constrained On-Chip AI Classifier for Beat-by-Beat Real-Time Arrhythmia Detection with an ECG Wearable System. *Electronics* **2025**, *14*, 2654. https://doi.org/10.3390/electronics14132654.
65. Kim, D.; Seo, J.; Kwon, S.; Park, C.; Han, C.; Kim, Y.; Kim, J.; Kim, C.; Park, S.; Yoon, D.; Kim, K. Predicting In-Hospital Fall Risk Using Machine Learning with Real-Time Location System and Electronic Medical Records. *Journal of Cachexia, Sarcopenia and Muscle* **2025**, *16*, 13713. https://doi.org/10.1002/jcsm.13713.
66. Rosa, S.; Evizal, M.; Assidiqi, F. Patient Monitoring System Using Smart Sensor Technology, Internet of Things and AI. *International Journal of Scientific Research* **2025**, *6*, 123–135. https://doi.org/10.25299/ijsr.2024.21941.
67. Muhammed, D.; Ahvar, E.; Ahvar, S.; Trocan, M.; Montpetit, M.; Ehsani, R. Artificial Intelligence of Things (AIoT) for Smart Agriculture: A Review of Architectures, Technologies and Solutions. *Journal of Network and Computer Applications* **2024**, *228*, 103905. https://doi.org/10.1016/j.jnca.2024.103905.
68. Luo, X.; Xiong, S.; Jia, X.; Zeng, Y.; Chen, X. AIoT-Enabled Data Management for Smart Agriculture: A Comprehensive Review on Emerging Technologies. *IEEE Access* **2025**, *13*, 102964–102993. https://doi.org/10.1109/ACCESS.2025.3578751.
69. Miller, T.; Mikiciuk, G.; Durlik, I.; Mikiciuk, M.; Łobodzińska, A.; Śnieg, M. The IoT and AI in Agriculture: The Time Is Now—A Systematic Review of Smart Sensing Technologies. *Sensors* **2025**, *25*, 3583. https://doi.org/10.3390/s25123583.
70. Sharma, K.; Shivandu, S. Integrating Artificial Intelligence and Internet of Things (IoT) for Enhanced Crop Monitoring and Management in Precision Agriculture. *Sensors International* **2024**, *5*, 100292. https://doi.org/10.1016/j.sintl.2024.100292.
71. Nyakuri, J.; Nkundineza, C.; Gatera, O.; Nkurikiyeyezu, K.; Mwitende, G. AI and IoT-Powered Edge Device Optimized for Crop Pest and Disease Detection. *Scientific Reports* **2025**, *15*, 6452. https://doi.org/10.1038/s41598-025-06452-5.
72. Pintus, M.; Colucci, F.; Maggio, F. Emerging Developments in Real-Time Edge AIoT for Agricultural Image Classification. *IoT* **2025**, *6*, 13. https://doi.org/10.3390/iot6010013.
73. Kum, S.; Oh, S.; Moon, J. Edge AI Framework for Large-Scale Smart Agriculture. In *Proceedings of the 2024 27th Conference on Innovation in Clouds, Internet and Networks (ICIN)*, 2024; pp. 143–147. https://doi.org/10.1109/ICIN60470.2024.10494451.
74. Michaels, H.; Rinderle, M.; Benesperi, I.; Freitag, R.; Gagliardi, A.; Freitag, M. Emerging Indoor Photovoltaics for Self-Powered and Self-Aware IoT towards Sustainable Energy Management. *Chemical Science* **2023**, *14*, 5350–5360. https://doi.org/10.1039/D3SC00659J.
75. Merenda, M.; Porcaro, C.; Iero, D. Edge Machine Learning for AI-Enabled IoT Devices: A Review. *Sensors* **2020**, *20*, 2533. https://doi.org/10.3390/s20092533.
76. Siam, S.; Ahn, H.; Liu, L.; Alam, S.; Shen, H.; Cao, Z.; Shroff, N.; Krishnamachari, B.; Srivastava, M.; Zhang, M. Artificial Intelligence of Things: A Survey. *ACM Transactions on Sensor Networks* **2024**, *21*, 1–75. https://doi.org/10.1145/3690639.
77. Fu, H.; Rao, J.; Deng, F.; Wang, Y.; Zhao, B.; Liu, Z.; Guan, H.; Malinowski, P.; Xu, L. AIoT: Artificial Intelligence and the Internet of Things for Monitoring and Prognosis of Systems and Structures. *IEEE Transactions on Instrumentation and Measurement* **2025**, *74*, 1–32. https://doi.org/10.1109/TIM.2025.3557124.

 doi:10.20944/preprints202604.0261.v1

78. Jouini, O.; Sethom, K.; Namoun, A.; Aljohani, N.; Alanazi, M.; Alanazi, M. A Survey of Machine Learning in Edge Computing: Techniques, Frameworks, Applications, Issues, and Research Directions. *Technologies* **2024**, *12*, 81. https://doi.org/10.3390/technologies12060081.
79. Singh, R.; Gill, S. Edge AI: A Survey. *Internet of Things and Cyber-Physical Systems* **2023**, *3*, 71–92. https://doi.org/10.1016/j.iotcps.2023.02.004.
80. Bourechak, A.; Zedadra, O.; Kouahla, M.; Guerrieri, A.; Seridi, H.; Fortino, G. At the Confluence of Artificial Intelligence and Edge Computing in IoT-Based Applications: A Review and New Perspectives. *Sensors* **2023**, *23*, 1639. https://doi.org/10.3390/s23031639.
81. Lin, J.; Zhu, L.; Chen, W.; Wang, W.; Han, S. Tiny Machine Learning: Progress and Futures. *IEEE Circuits and Systems Magazine* **2024**, *23*, 8–34. https://doi.org/10.1109/MCAS.2023.3302182.
82. Bonomi, F.; Milito, R.; Natarajan, P.; Zhu, J. Fog Computing and Its Role in the Internet of Things. In *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing (MCC '12)*, Helsinki, Finland, 17 August 2012; ACM: New York, NY, USA, 2012; pp. 13–16. https://doi.org/10.1145/2342509.2342513.
83. Iorga, M.; Feldman, L.; Barton, R.; Martin, M. J.; Goren, N. S.; Mahmoudi, C. The NIST Definition of Fog Computing. NIST Special Publication 500-325; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2018. https://doi.org/10.6028/NIST.SP.500-325.
84. Mach, P.; Becvar, Z. Mobile Edge Computing: A Survey on Architecture and Computation Offloading. *IEEE Communications Surveys & Tutorials* **2017**, *19*, 1628–1656. https://doi.org/10.1109/COMST.2017.2682318.
85. Mao, Y.; You, C.; Zhang, J.; Huang, K.; Letaief, K. B. A Survey on Mobile Edge Computing: The Communication Perspective. *IEEE Communications Surveys & Tutorials* **2017**, *19*, 2322–2358. https://doi.org/10.1109/COMST.2017.2745201.
86. Zhou, Z.; Chen, X.; Li, E.; Zeng, L.; Luo, K.; Zhang, J. Edge Intelligence: Paving the Last Mile of Artificial Intelligence with Edge Computing. *Proceedings of the IEEE* **2019**, *107*, 1738–1762. https://doi.org/10.1109/JPROC.2019.2918951.
87. Kairouz, P.; McMahan, H. B.; Avent, B.; et al. Advances and Open Problems in Federated Learning. *Foundations and Trends in Machine Learning* **2021**, *14*, 1–210. https://doi.org/10.1561/2200000083.
88. Bonawitz, K.; Ivanov, V.; Kreuter, B.; et al. Practical Secure Aggregation for Privacy-Preserving Machine Learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS '17)*, 2017; pp. 1175–1191. https://doi.org/10.1145/3133956.3133982.
89. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016; pp. 770–778. https://doi.org/10.1109/CVPR.2016.90.
90. Cheng, Y.; Wang, D.; Zhou, P.; Zhang, T. A Survey of Model Compression and Acceleration for Deep Neural Networks. *Proceedings of the IEEE* **2018**, *106*, 2295–2329. https://doi.org/10.1109/JPROC.2017.2765659.
91. Sze, V.; Chen, Y.-H.; Yang, T.-J.; Emer, J. S. Efficient Processing of Deep Neural Networks: A Tutorial and Survey. *Proceedings of the IEEE* **2017**, *105*, 2295–2329. https://doi.org/10.1109/JPROC.2017.2761740.
92. Liakos, K. G.; Busato, P.; Moshou, D.; Pearson, S.; Bochtis, D. Machine Learning in Agriculture: A Review. *Sensors* **2018**, *18*, 2674. https://doi.org/10.3390/s18082674.
93. Ghamari, M.; Janko, B.; Sherratt, R. S.; Harwin, W.; Piechockic, R.; Soltanpur, C. A Survey on Wireless Body Area Networks for eHealthcare Systems in Residential Environments. *Sensors* **2016**, *16*, 831. https://doi.org/10.3390/s16060831.
94. Centenaro, M.; Vangelista, L.; Zanella, A.; Zorzi, M. Long-Range Communications in Unlicensed Bands: The Rising Stars in the IoT and Smart City Scenarios. *IEEE Wireless Communications* **2016**, *23*, 60–67. https://doi.org/10.1109/MWC.2016.7721743.
95. Atzori, L.; Iera, A.; Morabito, G. The Internet of Things: A Survey. *Computer Networks* **2010**, *54*, 2787–2805. https://doi.org/10.1016/j.comnet.2010.05.010.
96. Gubbi, J.; Buyya, R.; Marusic, S.; Palaniswami, M. Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions. *Future Generation Computer Systems* **2013**, *29*, 1645–1660. https://doi.org/10.1016/j.future.2013.01.010.
97. Akhtar, N.; Mian, A. Threat of Adversarial Attacks on Deep Learning in Computer Vision: A Survey. *IEEE Access* **2018**, *6*, 14410–14430. https://doi.org/10.1109/ACCESS.2018.2807385.
98. Shokri, R.; Stronati, M.; Song, C.; Shmatikov, V. Membership Inference Attacks Against Machine Learning Models. In *Proceedings of the IEEE Symposium on Security and Privacy (SP)*, 2017; pp. 3–18. https://doi.org/10.1109/SP.2017.41.

 doi:10.20944/preprints202604.0261.v1

99. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; Arcas, B. A. y. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*; PMLR: 2017; pp. 1273–1282. https://arxiv.org/abs/1602.05629.
100. Li, T.; Sahu, A. K.; Talwalkar, A.; Smith, V. Federated Learning: Challenges, Methods, and Future Directions. *IEEE Signal Processing Magazine* **2020**, *37*, 50–60. https://doi.org/10.1109/MSP.2020.2975749.
101. Wang, S.; Zhang, X.; Zhang, Y.; Wang, L.; Yang, J.; Wang, W. A Survey on Mobile Edge Networks: Convergence of Computing, Caching and Communications. *IEEE Transactions on Wireless Communications* **2017**, *16*, 4924–4938. https://doi.org/10.1109/TWC.2017.2703901.
102. Satyanarayanan, M. The Emergence of Edge Computing. *Computer* **2017**, *50*, 30–39. https://doi.org/10.1109/MC.2017.9.
103. Vecchio, M.; Azzoni, P.; Menychtas, A.; Maglogiannis, I.; Felfernig, A. A Fully Open-Source Approach to Intelligent Edge Computing: AGILE's Lesson. *Sensors* **2021**, *21*, 1309. https://doi.org/10.3390/s21041309.
104. Dutta, J.; Bharali, A. TinyML Meets IoT: A Comprehensive Survey. *Internet of Things* **2021**, *16*, 100461. https://doi.org/10.1016/j.iot.2021.100461.
105. Trigkas, D.; et al. Edge Intelligence in Urban Landscapes: Reviewing TinyML Applications for Connected and Sustainable Smart Cities. *Electronics* **2025**, *14*, 2890. https://doi.org/10.3390/electronics14142890.
106. Kim, S.-H.; Kim, T. Local Scheduling in KubeEdge-Based Edge Computing Environment. *Sensors* **2023**, *23*, 1522. https://doi.org/10.3390/s23031522.
107. Abubakri, A.; Nyarko, E. K.; Ouyang, L.; Carmichael, S.; Okojie, E.; Tetteh, R. From Bytes to Insights: A Human-in-Loop Explainable AI Framework for Optimal AI Systems. *Electronics* **2025**, *14*, 3384. https://doi.org/10.3390/electronics14173384.
108. Panagoulias, G.; Papanikolaou, K.; Karakolis, C.; Skianis, C. Human-in-the-Loop: A Review of Advancements, Challenges, and Opportunities in Smart Manufacturing. *Systems* **2023**, *11*, 35. https://doi.org/10.3390/systems11010035.
109. Younis, H.; Pompili, F.; Taamallah, A.; Htike, K. K.; Treekittiphun, C.; Tong, K.; Pal, R. Recent Advances in Multimodal Data Fusion and AI for Human Mobility Monitoring: A Survey. *Sensors* **2024**, *24*, 566. https://doi.org/10.3390/s24020566.
110. Hossny, M.; Anwar, A.; Georgy, J.; Samir, S. Robot Sensor Fusion: A Survey of the Current Research and Future Trends. *Sensors* **2022**, *22*, 305. https://doi.org/10.3390/s22010305.
111. Mirza, H.; Islam, R.; Jarrahi, F. Multi-Exit Neural Network Inference with Energy Harvesting and Intermittent Computing. *J. Low Power Electron. Appl.* **2025**, *15*, 19. https://doi.org/10.3390/jlpea15020019.
112. Hady, M.; Bader, A.; Schellenberg, J.; Rübsamen, M. Downlink Performance Modeling of an Energy Harvesting LoRaWAN Class A Device. *Electronics* **2021**, *9*, 904. https://doi.org/10.3390/electronics9060904.
113. Makhdoom, I.; Abolhasan, M.; Ni, W.; Lipman, J.; Jamalipour, A. Blockchain-Based Federated Learning in Internet of Things: A Comprehensive Survey. *Future Internet* **2021**, *13*, 276. https://doi.org/10.3390/fi13090276.
114. Saadi, M.; Loukil, A.; Bouchoucha, M.; Boujemaa, H. A Survey on IoT Application Architectures and Their Deployment Frameworks. *Sensors* **2024**, *24*, 5320. https://doi.org/10.3390/s24165320.
115. Kister, D.; Garcia, N.; Cardona, V. Selecting an Edge AI Hardware Platform: A Practical Framework for Embedded Deep Learning. *Applied Sciences* **2025**, *15*, 7870. https://doi.org/10.3390/app15137870.
116. Nabil, N.; Gendreau, M.; Le Digabel, S.; et al. Artificial Intelligence of Things (AIoT) in Smart Buildings and Smart Cities: A Comprehensive Review. *Sustainability* **2025**, *17*, 10313. https://doi.org/10.3390/su172210313.
117. Bhuyan, M. N. R.; Islam, M. R.; Rahman, M. M.; et al. A Systematic Review of Industry 4.0 Technology on Workforce Employability and Skills: Driving Success Factors and Challenges in South Asia. *Economies* **2024**, *12*, 35. https://doi.org/10.3390/economies12020035.
118. Pan, S.; Zhang, N.; Chen, Y.; Lin, X.; Zhang, Q. Personal Information Lifecycle Framework in the Internet of Things: A Practical Approach to Support GDPR Compliance. *Sensors* **2021**, *21*, 7592. https://doi.org/10.3390/s21227592.
119. TensorFlow. LiteRT for Microcontrollers. Official documentation, accessed March 2026. https://ai.google.dev/edge/litert/microcontrollers/overview.
120. PyTorch. ExecuTorch Documentation. Official documentation, accessed March 2026. https://docs.pytorch.org/executorch/index.html.
121. ONNX Runtime. ONNX Runtime Documentation. Official documentation, accessed March 2026. https://onnxruntime.ai/docs/.
122. ONNX Runtime. Execution Providers. Official documentation, accessed March 2026. https://onnxruntime.ai/docs/execution-providers/.

 doi:10.20944/preprints202604.0261.v1

123. Lai, L.; Suda, N.; Chandra, V. CMSIS-NN: Efficient Neural Network Kernels for Arm Cortex-M CPUs. *arXiv* **2018**, arXiv:1801.06601. https://arxiv.org/abs/1801.06601.
124. NVIDIA. NVIDIA TensorRT Documentation. Official documentation, accessed March 2026. https://docs.nvidia.com/deeplearning/tensorrt/latest/index.html.
125. Intel. OpenVINO Documentation. Official documentation, accessed March 2026. https://docs.openvino.ai/.
126. Hymel, S.; Banbury, C.; Situnayake, D.; Elium, A.; Ward, C.; Kelcey, M.; Baaijens, M.; Majchrzycki, M.; Plunkett, J.; Tischler, D.; et al. Edge Impulse: An MLOps Platform for Tiny Machine Learning. *Proceedings of Machine Learning and Systems* **2023**, *5*, 14–30. https://proceedings.mlsys.org/paper_files/paper/2023/file/49fe55f5e9574714dda575bfb2177662-Paper-mlsys2023.pdf.
127. Amazon Web Services. What Is AWS IoT Greengrass? Official documentation, accessed March 2026. https://docs.aws.amazon.com/greengrass/v2/developerguide/what-is-iot-greengrass.html.
128. Amazon Web Services. Machine Learning Components for AWS IoT Greengrass. Official documentation, accessed March 2026. https://docs.aws.amazon.com/greengrass/v2/developerguide/machine-learning-components.html.
129. Amazon Web Services. Model Performance Optimization with SageMaker Neo. Official documentation, accessed March 2026. https://docs.aws.amazon.com/sagemaker/latest/dg/neo.html.
130. Amazon Web Services. Set Up Neo on Edge Devices. Official documentation, accessed March 2026. https://docs.aws.amazon.com/sagemaker/latest/dg/neo-getting-started-edge.html.
131. Microsoft. Azure IoT Edge Documentation. Microsoft Learn, accessed March 2026. https://learn.microsoft.com/en-us/azure/iot-edge/.
132. Microsoft. How Azure IoT Edge Modules Run Logic on Devices. Microsoft Learn, accessed March 2026. https://learn.microsoft.com/en-us/azure/iot-edge/iot-edge-modules.
133. Xiong, Y.; Sun, Y.; Xing, L.; Huang, Y. Extend Cloud to Edge with KubeEdge. In *Proceedings of the 2018 Third ACM/IEEE Symposium on Edge Computing (SEC)*, 2018; pp. 373–377. https://doi.org/10.1109/SEC.2018.00048.
134. KubeEdge. Why KubeEdge. Official documentation, accessed March 2026. https://kubeedge.io/docs/.
135. EdgeX Foundry. Overview. Official documentation, accessed March 2026. https://docs.edgexfoundry.org/4.0/.
136. LF Edge. EdgeX Foundry Project Overview. Official project page, accessed March 2026. https://lfedge.org/projects/edgex-foundry/.