EMERGING ADVANCES IN INTEGRATED TECHNOLOGY
e-ISSN: 2773-5540

EmAIT

Vol. 6 No. 2 (2025) 64-80
https://publisher.uthm.edu.my/ojs/index.php/emait

A Mini Review on Edge Computing Devices for IoT Systems

Tan Chee Chen[1]*, Mahdzir Jamiaan[1,2], Farhanahani Mahmud[1], Herdawatie Abdul Kadir[1], Chin Fhong Soon[1,2]*

[1] *Faculty of Electrical and Electronic Engineering,*
Universiti Tun Hussein Onn Malaysia (UTHM) Parit Raja, Batu Pahat, 86400 Johor, MALAYSIA

[2] *Microelectronics and Nanotechnology-Shamsuddin Research Centre, Institute for Integrated Engineering, Universiti Tun Hussein Onn Malaysia (UTHM) Parit Raja, Batu Pahat, 86400 Johor, MALAYSIA*

*Corresponding Author: cvincenttan@gmail.com or soon@uthm.edu.my

DOI: https://doi.org/10.30880/emait.2025.06.02.007

Article Info

Received: 20 May 2025
Accepted: 28 December 2025
Available online: 31 December 2025

Keywords

IoT, edge computing, case study, security

Abstract

Edge computing is a transformational distributed computing paradigm that brings computers and data storage closer to IoT data sources, overcoming critical limitations of cloud-centric approaches such as excessive latency, bandwidth constraints, privacy concerns, and energy inefficiency. This review examines the fundamental requirements of edge computing devices for IoT, exploring essential characteristics including processing capabilities (through microcontrollers, microprocessors, and AI accelerators like GPUs, TPUs, and VPUs), energy efficiency mechanisms, ultra-low latency solutions, connectivity options (LoRa, Wi-Fi, BLE, 5G), and robust security frameworks. Current challenges in designing edge computing for IoT applications include benchmarking difficulties due to a lack of standardized metrics, severe resource constraints, security vulnerabilities in distributed networks, scalability issues in dynamic environments, and limited real-time adaptability with pre-trained models. Emerging technologies such as federated learning, 6G connectivity, TinyML innovations, and sustainable design approaches show significant promise for improving edge computing efficiency. The findings reveal that edge computing has profoundly impacted IoT development by enabling real-time processing and decision-making across industries, enhancing data privacy, improving energy efficiency, and facilitating AI integration into resource-constrained environments, while continuing research focuses on addressing persistent challenges in scalability, security, standardization, and sustainable deployment.

1. Introduction

The Internet of Things (IoT) attracts tremendous interest due to its technical, social, and economic significance. Without human intervention, IoT can transform surrounding objects and infrastructure into intelligent entities that can communicate with each other [1]. This has expanded into various services, including healthcare, transportation, industrialization, and agriculture.

Initially, cloud computing was identified as a viable solution for hosting IoT services, giving rise to cloud-centric IoT. The cloud provides persistent data storage and powerful, unlimited computation resources. However, the proliferation of IoT devices has led to an explosion in data generation. Dispatching this vast amount of data to the cloud can be problematic, bringing challenges and vulnerabilities like data breaches and single points of failure [2]. The exponential increase in connected devices and the transmission of raw data towards centralized cloud data centers increases network congestion and latency, thus reducing the feasibility of cloud-centric IoT analytics [3]. Moreover, traditional machine learning models, which are often needed for processing and decision-making with IoT data, require significant computing resources that make it infeasible to deploy directly at the edge using traditional methods. Outsourcing data to the cloud also increases the risk of privacy concerns [4].

1.1 Motivation

Edge computing is a distributed computing modality that brings computation and data storage closer to the location where they are needed [5]. Edge computing is a paradigm limited to the edge network, consisting of devices like mobile phones and access points at the immediate first hop from IoT devices (Fig. 1). It is a comprehensive platform that integrates network, processing, storage, and application capabilities at the edge of a network, physically proximate to the data source. Edge devices gather and transmit sensor data as quickly and privately as possible, as shown in Fig. 1.

The need for edge computing stems directly from the limitations of purely cloud-based approaches for IoT. Processing data at the edge offers significant advantages, such as minimizing the need for frequent access to the cloud for data uploading, which reduces the required bandwidth for cloud communication and improves response rates. Edge computing leads to enhanced performance, lower network usage, and reduced latency compared to cloud, especially for time-sensitive applications. Edge computing also brings computational capacity, storage space, and quick response times required by demanding IoT applications.

1.2 Scope of Review

As edge computing plays an increasingly critical role in enabling real-time, low-latency, and energy-efficient operations, it is essential to understand the variety of hardware options available to support such functions at the edge of the network, in addition to assessing their technical specifications and how they address the evolving demands of modern IoT applications.

To facilitate this review, edge devices in the IoT can be broadly categorized based on their computational capabilities, role in the network, and hardware properties, starting from basic sensor data collection. These include microcontroller-based units, single-board computers, AI-accelerated platforms, and edge gateways.

Therefore, this review provides an overview of edge devices' design in IoT, their applications in the industry, market trends, and key requirements of edge computing devices in IoT systems across various edge hardware platforms. Additionally, this review surveys current industry implementations and identifies gaps in the research landscape.

1.2.1 Key Design Consideration

Given the critical nature of edge computing in IoT systems, devices must meet stringent standards that ensure security, efficiency, and robustness in performance[6]. One of the key advantages of edge computing is its ability to achieve ultra-low latency by processing data locally at its source, thus eliminating round-trip delays to the cloud [12]. However, several constraints have been identified on edge computing in IoT architecture. Balancing computational power with available resources is a key design challenge, and existing work addresses the use of lightweight models and hardware-software designs for embedded intelligence [2]. In some cases, edge devices must have sufficient computing power to compute locally, perform analytics, and even make ML inferences.

Energy efficiency is essential in edge devices, which are battery-operated or rely on energy harvesting. They typically operate in remote or inaccessible locations, and frequent battery replacement is a hassle. The trade-off between processing demand and available energy is a distinguishing aspect in edge device architectures.

Edge devices should keep local processing latency to a minimum through optimization algorithms and hardware acceleration. The edge device should also support low-latency communication protocols such as BLE, Wi-Fi, 5G, and LoRa for optimal data transfer. The edge device must also have adequate and scalable connectivity to exchange information with other devices, gateways, or cloud services. Based on application requirements, range, and power constraints, the edge device should support multiple communication protocols such as Wi-Fi, BLE, Zigbee, LoRa, NBIoT, and 4G/5G. Edge devices must support adaptive network management to maintain connections in dynamic or heterogeneous environments.

Connectivity also impacts energy consumption and latency. For example, low-power wide-area networks (LPWANs) like LoRa are energy-efficient but offer limited bandwidth, which is suitable for periodic sensor data. In contrast, Wi-Fi or 5G provides higher throughput but at a greater energy cost. Edge devices must manage these trade-offs to ensure power efficiency and connection reliability.

Edge devices often handle sensitive or personal data and operate outside physically secure environments, robust security and privacy measures are non-negotiable [13]. Security and privacy requirements must be balanced with the device’s resource constraints, often necessitating lightweight cryptographic protocols and efficient key management.

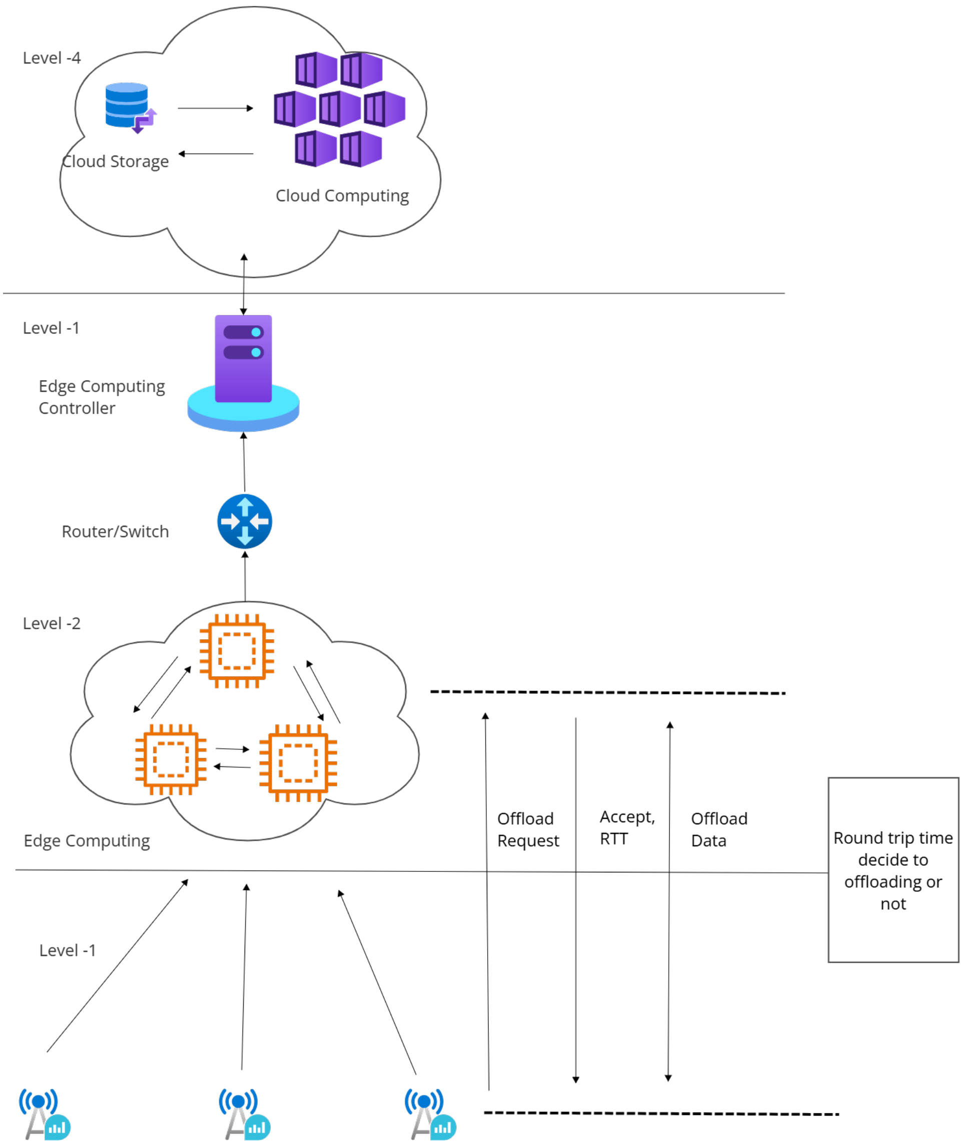

Fig.1 *Overview of the layered architecture of edge-server computing [14]*

1.2.2 Industry Applications

Edge computing is increasingly transforming industry applications by enabling real-time data processing, reducing latency, and supporting intelligent decision-making directly at or near the source of data generation [6]. In manufacturing, edge computing allows for immediate feedback and rapid analysis of large production data volumes, which are essential for intelligent resource scheduling, anomaly detection, quality management, and autonomous systems. This minimizes the time and bandwidth required to transfer data to centralized cloud servers and supports more energy-efficient operations. However, successful deployment involves investment in infrastructure and upskilling employees in modern AI-driven computing methods. [14].

In the automotive sector, edge computing is integral to the Industrial Internet of Things (IIoT), driving advancements in smart manufacturing. By deploying edge devices tailored for automotive production, manufacturers can achieve enhanced connectivity, local data storage, and real-time coordination between cloud and edge resources, which is crucial for the efficient operation of connected and intelligent vehicles [15].

The energy industry also benefits from edge computing, particularly in distribution automation and intelligent acceptance systems. Here, edge technologies provide low-latency, energy-efficient solutions for monitoring and controlling power distribution, improving reliability, and supporting advanced automation. These systems leverage edge computing to address technical challenges such as security, privacy, and the need for robust, scalable architectures [16-17].

Edge computing is set to play a foundational role in Industry 5.0, where collaboration between humans and intelligent technologies (such as AI, robotics, and digital twins) is central. Edge computing supports this paradigm by enabling real-time, context-aware processing and communication, fostering human–robot collaboration, sustainability, and resilient industrial networks.

1.2.3 Research Gap and Objective

Current research on edge computing in IoT reveals critical gaps that hinder widespread adoption. Benchmarking challenges persist due to the lack of standardized metrics for evaluating latency, energy efficiency, and security across heterogeneous edge devices, complicating performance comparisons. Resource constraints, such as limited computational power, memory, energy demand, and optimized ML models, yet balancing accuracy with lightweight designs, remain unresolved. Security and privacy risks are magnified in distributed edge networks, particularly during data transmission, necessitating robust encryption and secure aggregation mechanisms. Scalability issues arise in dynamic environments where edge-node capacity struggles with increasing data volumes. This requires the ability to scale resources dynamically to accommodate fluctuating demands while maintaining performance. Additionally, real-time adaptability is limited, as most systems rely on pre-trained models ill-suited for dynamic IoT data shifts.

The review objectives aim to address these gaps by elaborating the architectural innovations in hardware-software co-design to optimize ML deployment on diverse edge devices. The addressing standardization pathways for benchmarking latency, energy use, and security in hybrid edge-cloud systems [6]. As computation shifts from the cloud to edge devices, particularly within the context of IoT and AI applications, evaluating the performance of these systems becomes crucial [18]. While exploring Edge Computing's role in reducing cloud dependency through on-device inference, enhancing privacy, and enabling low-connectivity operation.

2. Edge Computing in IoT

2.1 Definition and Evolution of Edge Computing

Edge computing is a distributed computing paradigm that brings computation and data storage closer to data generation sources, such as sensors, devices, and local gateways, rather than relying solely on centralized cloud servers. It is recognized as a critical enabler for next-generation IoT systems, providing the infrastructure necessary for scalable, responsive, and secure digital services.

The wide applications of IoT devices, advancements in mobile and wireless networks, and the growing demand for real-time data processing in applications like autonomous vehicles, smart cities, and industrial automation have driven the evolution of edge computing. Early distributed computing models, such as cloudlets and fog computing, paved the way for modern edge computing by demonstrating the benefits of processing data near its source.

Over time, edge computing has become more sophisticated, integrating with artificial intelligence and machine learning to enable intelligent decision-making at the edge, and supporting a wide range of application scenarios that require low latency, enhanced privacy, and efficient resource utilization. Today, edge computing is recognized as a critical enabler for next-generation IoT systems, providing the infrastructure necessary for scalable, responsive, and secure digital services.

2.2 Comparison of edge and cloud computing for IoT

Edge computing and cloud computing present distinct approaches for processing data in IoT systems, each with unique advantages and trade-offs. To gain a more profound comprehension of the trade-offs inherent in these two computing paradigms. Table 1 presents a comparative analysis of edge and cloud computing within the framework of IoT systems.

Table 1 Comparative analysis of edge computing and cloud computing

Aspect	Edge computing	Cloud computing
Processing Location	At or near the IoT device (local)	Centralized data centers
Latency	Very low latency; near real-time response [19]	Higher latency due to network delays
Bandwidth Usage	Reduced bandwidth (local processing reduces transmission) [14]	High bandwidth usage (transmission of raw data to the cloud)
Computational Resources	Limited; relies on lightweight or specialized hardware (e.g., GPUs, TPUs, MCUs)	Virtually unlimited computational power and storage
Privacy & Security	Better privacy; data stays local[13]	Higher privacy risk due to cloud transmission
Energy Efficiency	Lower energy for data transmission, but may require efficient hardware management	Centralized energy consumption is high due to data center operations
Scalability	Challenging at scale (heterogeneous, distributed devices)	Easily scalable with existing cloud infrastructure
Best Use Cases	Real-time analytics, industrial control, and smart devices	Big data analytics, archival storage, and model training

2.3 Market Trends and Adoption in Industries

The adoption of edge computing in IoT is accelerating in all industries, driven by the needs of real-time analytics, energy efficiency, and better privacy. Market research reports project the global edge computing market to be valued at $1.12 trillion by 2023 [4], citing a rapid growth rate in the industry. Technological migration is robust in verticals such as manufacturing, healthcare, smart cities, and transportation, where bandwidth constraints and latency-sensitive workloads are the determining factors. Edge computing is crucial in revolutionizing enterprise application infrastructure by processing data closer to its source. This reduces latency and bandwidth usage, crucial for real-time analytics and IoT applications. In healthcare applications, where decision-making is time-sensitive, increased latency is a significant concern in cloud-based systems.

In logistics, IoT enables companies to track shipments and monitor inventory for improved supply chain visibility and efficiency [21]. Serverless functions process data from IoT devices on delivery vehicles, streamline operations, and process real-time updates with low latency.

Retail businesses benefit from AI-driven insights for customer personalization and innovative inventory management. Edge computing solutions tailored for retail using AWS can streamline tasks like streaming and analyzing camera data, enhancing agility [10].

In industrial automation, edge computing is essential for real-time data processing. Edge computing and AI allow autonomous smart-edge fault diagnostics via edge-cloud collaboration [23]. The convergence of AI and edge computing will result in intelligent systems capable of real-time decision-making without relying on centralized cloud resources, vital for scenarios like predictive maintenance and anomaly detection in industrial IoT.

3. Key Requirements of Edge-Computing Devices

Edge computing devices enable real-time data processing, decision-making, and AI inference directly at or near the data source. To fulfill these roles effectively, edge devices must meet several critical requirements, particularly in processing power and AI acceleration, while operating under tight energy, memory, and cost constraints.

3.1 Processing Power

Edge devices, especially those used in IoT applications, typically operate with limited processing power compared to cloud servers [3]. This resource limitation means edge devices must be highly optimized to execute their workloads efficiently. Microcontrollers, microprocessors, and AI accelerators each serve distinct but complementary roles in providing processing power for edge and IoT devices, enabling a range of functionalities from basic control to advanced inference intelligence. The increasing demand for AI capabilities at the network edge has led to the emergence of various specialized hardware. Table 2 summarizes the key characteristics of standard processing units found in edge computing devices, highlighting their roles, strengths, limitations, and application suitability.

Table 2 Key characteristics of standard processing units found in edge computing devices

Type	Devices/Cores	Application	Advantages	Disadvantages	Ref.
MCU	ARM Cortex-M0/M4F/M7, ESP32	Simple sensing, basic edge decision	Low power, low cost, integrated peripherals	Limited processing, memory < 1MB	[7]
MPU	ARM Cortex-A, Raspberry Pi SoCs	Gateways, edge servers	Higher throughput, supports Linux, flexible IO	Higher power, more expensive	[36]
GPU	NVIDIA Jetson Nano	Real-time CV, multitask AI at the edge	Excellent throughput, flexible, real-time inference	High power usage, less energy efficient	[20][24]
TPU	Google Coral Dev Board	On-device ML inference	High performance per watt, efficient for DNNs	Inflexible, not suitable for general-purpose computing	[6][38]
VPU	Intel Movidius NCS2, Myriad X	Image/video processing in embedded IoT	Optimized for vision tasks, energy-efficient	Limited to CV/AI tasks	[8][25]

The choice of processor architecture is a key determinant of an edge device's capabilities. ARM Cortex-M series microcontrollers are widely used due to their balance of performance, power efficiency, and cost. Using 32-bit or 64-bit architecture allows for more efficient data handling and computation, while multi-core designs (e.g., ARM Cortex-M4F, Cortex-M7) further enhance processing throughput for parallel tasks.

The proliferation of AI and machine learning at the rise of Edge-Computing demands hardware that can efficiently execute complex models such as neural networks, even within tight resource budgets. Traditional CPUs, while versatile, are not optimized for the highly parallel operations that occur every day in AI workloads [25]. AI accelerators, GPUs, TPUs, and VPUs, are designed to handle the parallel computations required by deep learning models, drastically reducing inference time and energy consumption while maintaining accuracy [8].

Graphics Processing Units (GPUs) are the most widely used hardware accelerators for deep learning due to their inherent ability to perform highly parallel calculations, which is fundamental to deep learning models [24]. They excel in delivering performance for a wide range of deep learning models and offer high throughput and low latency for real-time processing. Their versatility and ability to handle multiple applications simultaneously make them a popular choice for edge AI, particularly where a balance between performance and flexibility is needed. For instance, a GPU platform showed computation time for time-frequency features to be 50 times faster than on a CPU [20].

While GPUs provide high performance and throughput, they also come with trade-offs, particularly in power consumption. GPUs consume more power than other accelerators like TPUs or FPGAs. This can be a significant drawback in edge environments where energy efficiency is a primary concern, making them unsuitable for power-sensitive applications.

Tensor Processing Units (TPUs) are special accelerators developed by Google specifically for deep learning workloads. They are a hardware accelerator designed to improve performance and efficiency for neural networks. TPUs are optimized for computationally intensive tasks everyday in machine learning models, such as tensor functions like matrix multiplication and convolution. These are core operations in neural networks.

One of the main advantages of TPUs is their energy efficiency, as they are designed to ensure outstanding performance while requiring much less power than Graphics Processing Units (GPUs). TPUs offer remarkable stability and exhibit high effectiveness for operations such as matrix multiplication. Nevertheless, TPUs are more inflexible than GPUs, as they are primarily tailored for deep learning applications and may prove unsuitable for general-purpose computing tasks.

VPUs are specialized hardware accelerators, designed to enable demanding computer vision and AI workloads efficiently [25]. They enable efficient visual data processing (e.g., images, video) in resource-constrained IoT devices while balancing power consumption and performance. VPUs leverage parallel processing and dedicated circuits for convolutional neural networks (CNNs), enabling high-throughput inference for tasks like object detection and image segmentation [8].

Microcontrollers (MCUs) are compact, integrated circuits designed primarily for control-oriented tasks in embedded systems. They typically include a CPU, memory (RAM and flash), and peripherals (such as timers, ADCs, and communication interfaces) all on a single chip [27]. In the IoT context, microcontrollers are widely used for data collection, basic signal processing, and simple decision-making tasks directly at the edge. Their low power consumption and cost make them ideal for battery-operated devices such as sensors and actuators. However, their processing power is limited at clock speeds between 10–1000 MHz and has less than 1 MB of RAM and flash memory. Therefore, they are best suited for lightweight workloads and inference with highly optimized, small-scale machine learning models.

Microprocessors are generally designed for information processing in computer systems. Their design is specifically aimed at serving as the CPU in a microcomputer system. Compared to microcontrollers, the instruction set of a microprocessor is enhanced to provide powerful addressing modes and instructions suitable for operating large-scale data. In edge computing, microprocessors are often found in devices that require higher computational throughput, such as gateways, routers, or edge servers. However, they typically consume more power and are more expensive than microcontrollers, making them less suitable for ultra-low-power or cost-sensitive edge deployments [27].

3.2 Energy Efficiency

The traditional cloud-centric approach to processing data from the increasing number of IoT devices often faces challenges related to energy efficiency. The continuous transmission of voluminous raw data over long distances to centralized cloud servers is described as consuming energy and overloading network bandwidth. This is a significant limitation for scaling IoT solution deployments [28].

Localized processing at the network edge, closer to the data source, enables faster response times and reduces the reliance on cloud-based processing, thus minimizing data travel distance. This reduced reliance on distant servers translates directly to less data transmission to the cloud, which lowers energy consumption.

Another method is CPU-cycle frequency scaling. Since a CPU's power consumption rises exponentially with its frequency, conserving local execution energy can be significantly achieved by reducing the CPU-cycle frequency [29]. Intelligent management of where and how tasks are processed is crucial. Edge computing strives to dynamically distribute computational jobs to edge nodes through energy-aware scheduling algorithms [28]. Algorithms like TOFFEE (Task Offloading and Frequency Scaling for Energy Efficiency) are designed to solve stochastic optimization problems involving task allocation and CPU frequency decisions to minimize energy consumption while keeping task queue length bounded [29]. This involves dynamically allocating the computation workload and scheduling the CPU-cycle frequency to adapt to changing conditions.

Data filtering and aggregation are critical for enhancing power efficiency in IoT systems, particularly edge computing environments. These processes minimize energy consumption by reducing unnecessary data transmission and optimizing local processing [28]. Signal compression is also mentioned as efficient for reducing the transmitted signal length and improving battery life [20].

Deep compression techniques include pruning and quantization, which can significantly reduce neural networks' storage requirements and memory footprint. However, implementing these optimizations often involves a trade-off between model accuracy and the required performance. Reducing precision too much through quantization might result in severe accuracy loss.

On the other hand, machine learning algorithms can provide promising predictive analytics to enhance energy efficiency. By accurately predicting future workload patterns, these algorithms can optimize resource management proactively, dynamically adopting operating parameters of edge nodes for power saving.

3.3 Latency

Edge computing architectures place computation and data processing closer to the data source. This decentralized approach reduces the distance data must travel, significantly reducing latency compared to sending data to distant centralized cloud servers.

The concept of computation offloading involves moving part or all of an application's execution to a remote server. The core idea is to enable devices to offload computation tasks to servers closer to the user, such as those in the mobile edge (ME) or edge computing infrastructure. This proximity drastically reduces the end-to-end access latency to computing resources compared to remote cloud servers.

Specialized hardware accelerators are employed to reduce computational time. These include GPUs, TPUs (Tensor Processing Units), FPGAs (Field-Programmable Gate Arrays), and ASICs (Application-Specific Integrated Circuits). These accelerators are designed to speed up ML/DL inference, often achieving very low inference times.

3.4 Connectivity

Edge computing is a distributed architecture that connects smart embedded devices to analyze and store data at the end of the network to enhance the quality of service. Edge computing in IoT leverages diverse wireless technologies, differing in data rates, communication distances, and working currents. These include technologies suitable for short-range communication like Bluetooth and Wi-Fi, and long-range communication technologies like LoRa [6].

LoRaWAN and other LPWANs (Low-Power Wide-Area Networks) employ low-power radio frequencies and operate at low data speeds9. They are suitable for IoT applications where devices must run on batteries for extended periods and may be scattered across broad regions.

Wi-Fi is listed as a short-range communication technology for IoT. Wi-Fi is widely used in IoT for its high data rates, making it suitable for data-intensive applications such as video streaming, image processing, and real-time analytics [20]. 5G networks are poised to significantly enhance connectivity for edge devices by providing high-speed data transmission and ultra-low latency. This improved performance is crucial for enabling real-time processing at the edge and supporting applications like autonomous vehicles, smart cities, and industrial automation, where minimal delays are critical [30]. 5G also supports a higher density of connected devices and addresses the demand for faster data rates, wider bandwidth, increased capacity, swift transmission, and minimal delays, specifically mentioning ultra-reliable low latency communication [3].

Bluetooth/BLE (Bluetooth Low Energy**)** is a short-range wireless technology suitable for IoT communication. BLE is designed for low power consumption, with some chips listing working currents at the mA level or even down to nA/μA in sleep/standby modes.

Selecting the right connectivity technology for IoT edge applications involves balancing these trade-offs. LoRa maximizes range and battery life at the bandwidth cost, while BLE minimizes power consumption with a very short range. Wi-Fi offers high throughput but at higher power and limited range. 5G on the other hand, provides the best latency and bandwidth for demanding applications but is costly and power-hungry.

3.5 Security and Privacy

Security and privacy are crucial requirements for edge computing devices within the Internet of Things (IoT) ecosystem, primarily due to their pervasive deployment, exposure to threats, and the sensitive nature of the data they handle. IoT devices are deployed at the network's edge, often with limited processing capacity and memory footprint. This distribution creates vulnerabilities, as each edge device can be a potential target for cyberattacks, and compromised devices can lead to unauthorized access to sensitive data or manipulation of critical systems [30]. The extensive data generated by edge devices can include sensitive information like user location, health records, activity logs, or proprietary manufacturing data, posing privacy concerns.

Secure Boot is a mechanism that ensures that only authenticated and untampered firmware can be executed on an edge device by verifying digital signatures during the startup process [18]. It establishes a Root of Trust (RoT), which is a fundamental mechanism upon which the entire reliable and safety stack is built, preventing malicious code from executing [31]. The secure boot process typically involves verifying the hash values of loaded software images, while invalid hashes indicate tampering. Therefore, the boot procedure is halted. Fig. 2 demonstrates the root of trust workflow.

Protecting data against unauthorized access and interception is essential, both for data at rest (stored on the device) and in transit (during communication) [33]. Symmetric key cryptography, such as Advanced Encryption Standard (AES), is popular for secure data communication between resource-constrained IoT devices and edge gateways due to its computational requirements being less dependent on key size compared to some public-key algorithms [33]. End-to-end encryption ensures data is encrypted at the edge device and remains encrypted throughout transmission to the cloud or other endpoints, protecting sensitive data even if intercepted.

Trusted Execution Environments (TEEs) provide isolated regions within the device hardware where sensitive operations and data can be processed securely, shielded from the rest of the system and potential attackers [33]. These environments are built on the concept of a secure hardware area where only trusted entities and applications have access. Intel Software Guard Extensions (SGX) is a popular example of a TEE that creates an isolated secure memory container (Enclave) for safe storage and execution of code and data, offering confidentiality, accessibility and integrity.

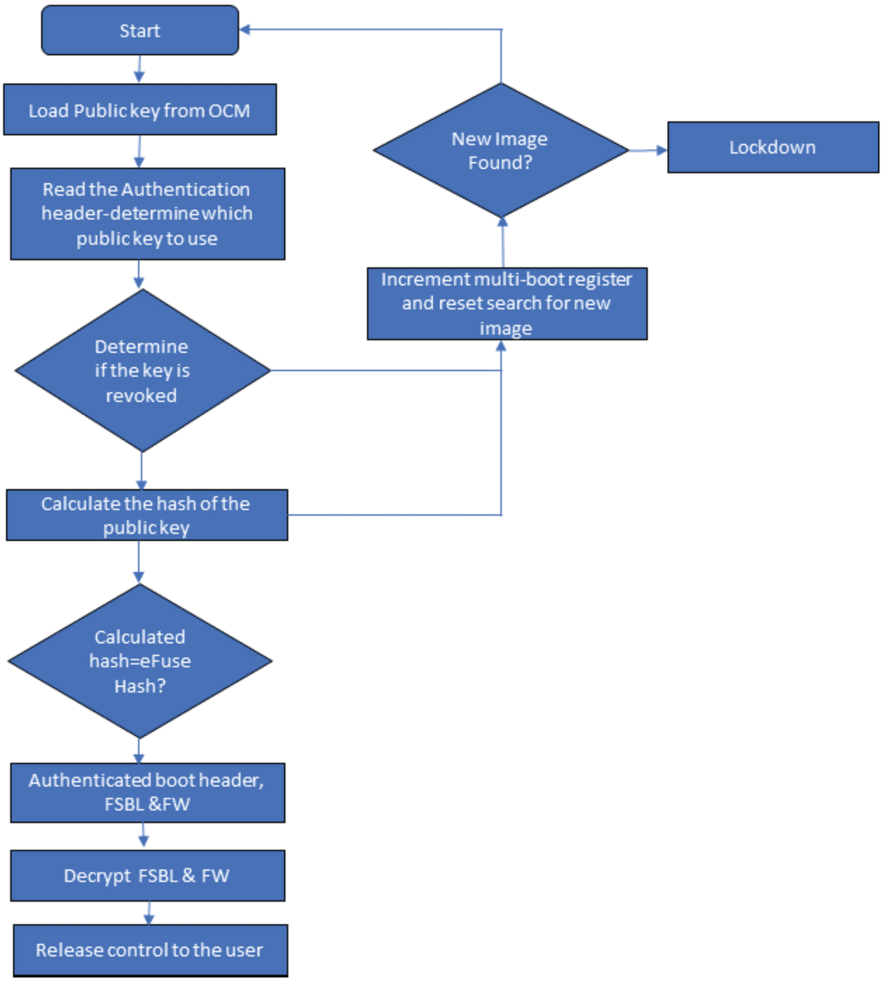
Start
Load Public key from OCM
Read the Authentication header-determine which public key to use
Determine if the key is revoked
Calculate the hash of the public key
Calculated hash=eFuse Hash?
Authenticated boot header, FSBL &FW
Decrypt FSBL & FW
Release control to the user
Increment multi-boot register and reset search for new image
New Image Found?
Lockdown

Fig. 2 *Root of Trust Work Flow [33]*

4. Case Studies: Edge Computing Devices in IoT

4.1 NVIDIA Jetson Nano: AI-Powered Industrial Automation.

Among various edge computing devices, the NVIDIA Jetson Nano is a compelling solution that delivers high-performance artificial intelligence capabilities in a compact, energy-efficient package. The NVIDIA Jetson Nano is a popular system-on-module (SoM) and developer kit that brings high-performance specifications to a small, power-efficient embedded platform. It is considered the entry-level board of the NVIDIA Jetson family [24]. It is a modular computer built around the Tegra X1 SoC3, explicitly designed for AI applications at the edge [9].

The Jetson Nano's hardware architecture includes a quad-core ARM Cortex-A57 CPU, operating at a frequency of 1.43 GHz. It is combined with a 128-core NVIDIA Maxwell GPU, which supports CUDA architecture version 5.3 and operates up to 921MHz. NVIDIA Jetson Nano is designed with a robust hardware architecture that integrates a quad-core ARM Cortex-A57 CPU clocked at 1.43 GHz, along with a 128-core Maxwell GPU operating at up to 921 MHz and supporting the CUDA architecture version 5.3. In its standard configuration, the development board comes with 4GB of LPDDR4 RAM, while the 2GB Jetson Nano variant offers a reduced memory footprint tailored for lighter applications. Fig. 3 shows the implementation of Jetson Nano in the NVR architecture.

Jetson Nano is designed to perform well for its small size while also considering low power consumption, operating in a power envelope of only 5 to 10 watts. This feature makes it ideal for embedded systems and energy-constrained environments, where power efficiency is critical. To accommodate varying computational demands, the platform supports two operating modes: Jetson-L for low-power consumption and Jetson-H for high-performance tasks, allowing developers to optimize the balance between processing power and energy consumption. The integration of Maxwell GPUs with Compute Unified Device Architecture (CUDA) cores enables the platform to efficiently perform neural network inference, supporting artificial intelligence (AI), deep learning, and computer vision applications. Tasks such as object detection and recognition benefit significantly from the GPU's parallel processing capabilities, improving real-time system responsiveness and accuracy [8]. CUDA architecture plays a key role in maximizing the utility of GPUs. In the Jetson Nano ecosystem, CUDA and related APIs are the primary interface for developers accessing CPU and GPU resources for high-performance computing tasks.

Practical applications of Jetson Nano extend beyond theoretical capabilities and into real-world industrial implementations. In the Assembly Time Measurement System, Jetson Nano is an embedded processing unit responsible for executing ML algorithms that automatically monitor, measure, and analyze worker activity [34]. This implementation aligns with the Industrial Internet of Things (IIoT) principles, providing actionable outputs such as task identification, installation duration (in seconds) and performance analytics, thus highlighting its viability as an intelligent edge computing solution in modern industrial environments.

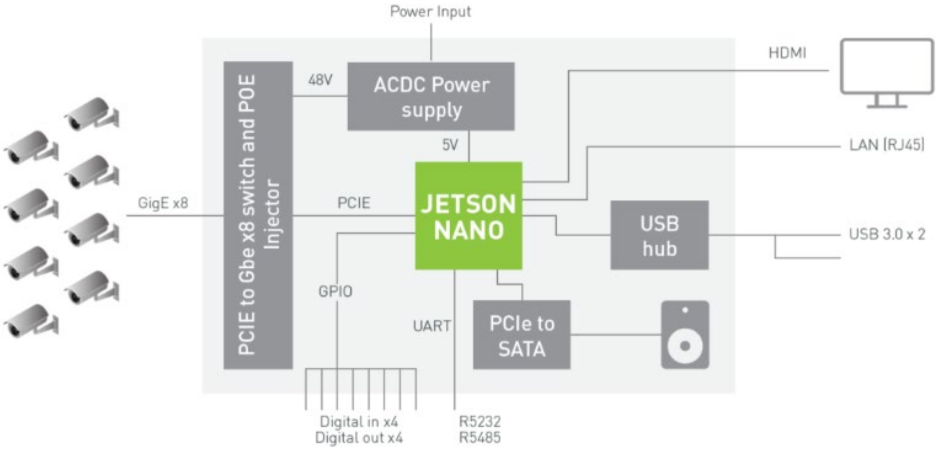

Fig. 3 *Reference NVR system architecture with Jetson Nano and 8x HD camera inputs [49]*

4.2 Raspberry Pi 4: Smart home and environmental monitoring

Raspberry Pi 4 features a 64-bit ARM Cortex-A72 quad-core CPU running at 1.5 GHz, a 400 MHz VideoCore IV GPU, and 8 GB SDRAM. The Raspberry Pi Compute Module 4 (RPi CM4) also offers a 4 GB LPDDR4 option with ECC RAM. The Raspberry Pi 4 can function as an edge node, an edge server, or a gateway (RPi CM4). It can also be part of distributed computing clusters, such as quad-node clusters, for parallel processing tasks. It is used for data collection, real-time data analysis and processing, feature extraction, model deployment, and prediction/classification tasks [20]. It can run machine learning inference without the need for additional hardware. With a lightweight design, it is suitable for implementing multiple types of Deep Neural Network (DNN) models.

The real-world IoT application demonstrated using Raspberry Pi4 is an IoT-based Indoor Air Quality Monitoring system [35]. This system is designed for continuous remote monitoring of the air quality within an indoor environment. The system utilizes a Raspberry Pi4 kit. Along with various sensors. These sensors include the Grove - Air Quality Sensor v1.3, CCS811 CO2 Air Quality Sensor, and DHT 11 Temperature and Humidity Sensor. Communication between sensors and the Raspberry Pi4 can occur via serial port protocols. Raspberry Pi 4 acts as a gateway, receiving and processing the information before transmitting it to the cloud.

4.3 Google Coral Dev Board: Edge AI applications

The Google Coral Dev Board is a compact single-board computer (SBC) which is usually applied as an IoT gateway, transmitting data from sensor nodes to CPU and storage systems. The board is suitable for fast ML inference in a very compact package [36] with very low power consumption of only about two watts. The board is based on the Edge Tensor Processing Unit (TPU); supports Python and C++ as programming languages and serves as a dedicated processor or co-processor for accelerating neural network computations [37] and performing highly efficient matrix computations in ML algorithms [6]. The Edge TPU System-on-Module (SoM) contains Quad-core Arm Cortex-A53 and Cortex-M4F, and the Google Edge TPU ML accelerator co-processor, as well as other components such as 8 GB eMMC and up to 4 GB LPDDR4 memory.

The Coral Dev Board is optimized for high-speed on-device ML computing due to its integrated Edge TPU. This enables real-time inference for various applications. It is capable of tasks like the calculation of feature vectors. A significant advantage of the Coral Dev Board is its minimal power consumption. The Edge TPU USB accelerator consumes around two watts.

The Coral Dev Board supports the use of TensorFlow Lite for model inference. Model optimization techniques like quantization are necessary to make models Coral-compatible. Knowledge Distillation (KD) can also be used to train efficient student models, such as MobileNetV2 distilled from a larger ViT model, which can then be deployed on the Coral Dev Board after quantization.

The Edge TPU's restriction is that it can only accelerate 8-bit integer models. This necessitates the quantization of models for acceleration [38]. The Coral Dev Board has limited RAM and CPU resources. This limitation means it is not possible to use it for large-scale data involving thousands or millions of images.

Google Coral Dev Board can be applied to a real-world scenario like Face Mask Detection as a low-cost Edge AI implementation [39]. A custom-trained Face Mask Detection model was developed using the TensorFlow Lite Model Maker library. The model architecture used was based on EfficientDet-lite0, and it was trained using a specific Face Mask Dataset.

This performance of approximately 6 FPS is sufficient for a photo search by keyword. The key advantages of using the Dev Board in this case are its low-cost implementation and small form factor device, which is engineered to be lightweight, portable, and easily integrated into constrained environments such as embedded systems.

4.4 Arduino Portenta H7: Industrial IoT and predictive maintenance

Arduino Portenta H7 is designed as a hardware platform for high-performance edge computing applications, particularly those involving real-time processing and ML. Its architecture is characterized by a dual-core processor setup [40].

Arduino Portenta H7 features a dual-core STM32H747 architecture with a 480 MHz Cortex M7 and a 240 MHz Cortex M4 that can run in parallel [35]. Cortex M7 processor is used for executing sophisticated activities such as ML programs, computer vision, and other compute-intensive operations [27]. The Cortex M4 processor is used for real-time execution, like reading sensor data and low-latency task execution [27]. The hardware comes equipped with 16 MB Flash and 8 MB SDRAM memory, which would be more than sufficient to support light models and in-board data processing. Additionally, the board supports versatile connectivity options like Wi-Fi and Bluetooth Low Energy (BLE), which makes it even more useful for various IoT applications and advantages.

One of the strongest aspects of Portenta H7 is the fact that it can deploy and execute ML models on the device itself. With processing taking place locally, rather than in cloud infrastructure, the system has the benefit of lower latency, improved data privacy, and better security. Such a model of on-device intelligence is gaining ground in the scenario of edge computing, where real-time decision-making assumes precedence.

The actual world application of Portenta H7 in ML tasks is presented through its implementation in the Keyword Spotting (KWS) task [40]. In a particular implementation, the model is not only trained but also deployed and executed on the device itself, offering possible new avenues for trying out more complex models for more complex problems with a higher memory budget and to fine-tune a model to new data in lower latency [41].

4.5 AWS Greengrass: Cloud-edge hybrid computing

AWS IoT Greengrass is an open-source edge runtime and cloud service that extends AWS capabilities to on-premises devices, enabling them to respond intelligently, process data on-premises, and securely link to the AWS Cloud [10]. The platform offers the required tools and frameworks to develop secure, scalable, and responsive IoT applications by bringing cloud processing capabilities closer to the device, solving connectivity problems, enabling remote management, and promoting localized data processing and synchronization. The software also handles authentication, authorization, and secure message routing between devices, Lambda functions, and the cloud using the MQTT protocol. Messages are sent to the AWS IoT Hub in the cloud [10].

The concept of integrating AI and ML capabilities into edge computing enables intelligent data processing directly at the source. Edge intelligence enables local data analysis, decision-making, and autonomous behavior. Local execution of serverless functions ensures rapid responses in IoT applications.

AWS Greengrass architecture is designed to extend AWS cloud capabilities to the edge, enabling IoT devices to process data locally, run AWS Lambda functions, execute machine learning inference, and securely communicate with the cloud and each other (Fig. 4).

Edge computing offers potential improvements in privacy and security by shifting security schemes from the cloud to IoT-edge devices. Processing sensitive data closer to its source minimizes exposure during transit. Greengrass core handles secure message routing. The need for robust security measures, such as encryption, is highlighted for edge-cloud interactions, especially concerning sensitive data.

In real real-world application [42], AWS Greengrass is used to build and deploy Internet of Things (IoT) applications that require edge computing capabilities. The application is developed by connecting an ESP32 microcontroller for sensor data, using a Raspberry Pi as the core device for local processing and display, and transferring data to the AWS cloud for analytics.

Greengrass provides features for managing devices in scalable groups. This includes capabilities for remote monitoring and over-the-air updates targeted at specific groups of devices, allowing many devices to be updated efficiently. Greengrass acts as a local cloud service, allowing other local devices to securely connect and share messages or state. It supports typical AWS Cloud functionalities at the edge, such as executing serverless Lambda functions to respond to events, managing and running Docker containers, and accessing device resources like GPIO ports.

AWS Greengrass provides the necessary tools and framework to build robust, scalable, and responsive IoT applications by bringing cloud processing capabilities to the edge, addressing connectivity issues, enabling remote management, and facilitating local data processing and synchronization.

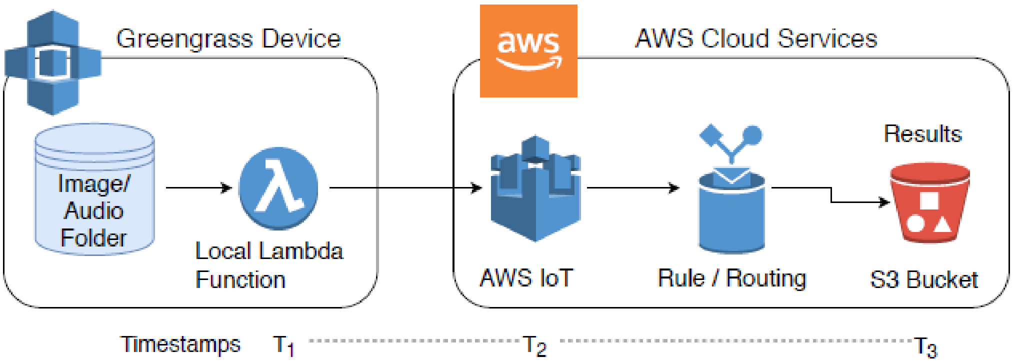

Fig. 4 *Amazon Greengrass Architecture [10]*

4.6 Intel Movidius NCS2: AI-accelerated vision processing

The Intel Neural Compute Stick 2 (NCS2) 4GB RAM 700MHz is powered by the Intel Movidius-X Vision Processing Unit (VPU) featuring 16 programmable cores and a dedicated neural computation engine used with the computer to accelerate DNN inference [9].

The Intel Neural Compute Stick 2 is used with a computer to accelerate Deep Neural Network (DNN) inference. It is utilized for ML tasks like image classification, semantic segmentation, and natural language processing. Intel provides the OpenVINO library to streamline the deployment of ML models on the NCS2. OpenVINO optimized DNN model with Inference Engine (e.g., VPUs) to create an optimized model, which is called Intermediate Representation (IR). The IR model contains only the network architecture, weights, and biases, which are the part being operated during inference time. Therefore, the OpenVINO inference engine API can be used to evaluate latency and power efficiency with the acceleration of the Inference Engine (e.g., VPUs). Fig. 5 shows the model optimizer mechanism of the OpenVINO DL Inference Engine.

Hardware-aware Neural Architecture Search (NAS) technologies [25] are proposed to automate model design to meet quality and inference efficiency requirements on the Intel Movidius VPU. Movidius VPUs aim for a balance of power efficiency and computing performance. The sources mention that TinyML solutions running neural networks on microcontrollers typically consume only a few milliwatts of power [32].

In a real-world application, Movidius NCS was implemented in a system for monitoring and capturing frames, and detecting and recognizing objects [42]. This system applies a deep learning technique using the Single Shot Detector (SSD) algorithm with MobileNet architecture. MobileNet is specifically designed for resource-constrained gadgets like smartphones and embedded vision applications.

The workflow involves passing an input image through the MobileNet architecture, followed by extra convolutional layers used by the SSD method to make predictions for objects of different sizes. The system then performs detection and recognition on the real-time video feed. Utilizing the Movidius NCS increased its performance to 3.5 FPS, making real-time object detection and recognition feasible. The stick helps build an efficient system for embedded vision applications.

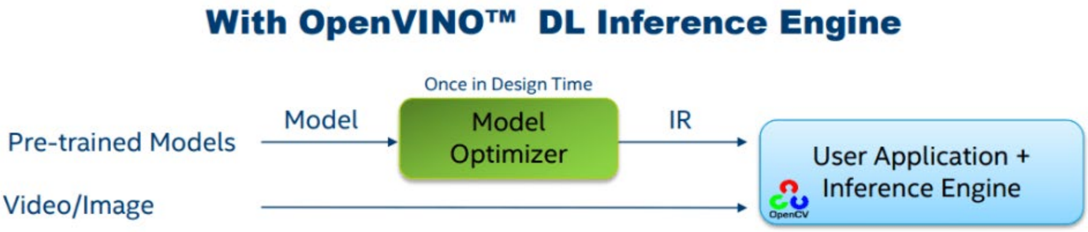

Fig. 5 *Optimizer mechanism of OpenVINO DL Inference Engine [50]*

5. Challenges and Limitations in Edge Computing for IoT

Edge computing for IoT faces several critical challenges and limitations that impact its scalability, efficiency, security, and interoperability. Scalability issues arise as IoT deployments expand, making it increasingly complex to manage resources across a decentralized network of heterogeneous edge devices. Efficient allocation of processing, storage, and network resources is essential, yet orchestrating workloads and synchronizing data between the edge and cloud remains challenging [30]. The diversity of device capabilities and protocols further complicates service discovery and seamless scaling, as each node may have different processing power, memory, and communication standards [3].

Another challenge is the power consumption versus performance trade-off. Edge devices, particularly those based on microcontrollers, are inherently resource-constrained, which restricts the complexity of machine learning models they can execute [2]. While offloading computation-intensive tasks to edge servers or the cloud (Mobile Edge Computing, MEC) can optimize both latency and energy consumption, this strategy is highly dependent on network reliability and low latency [8]. Unstable or high-latency networks can negate the benefits of offloading, forcing more processing to occur locally and straining device resources.

Security threats are also critical in edge IoT networks. IoT devices often communicate over various wireless and wired networks. Many IoT systems lack built-in secure communication protocols, including proper encryption and authentication, making them susceptible to data interception or manipulation during transmission [43]. Many devices lack robust built-in encryption and authentication, making them vulnerable to data interception, manipulation, and attacks such as Denial-of-Service (DoS) or Man-in-the-Middle (MitM) [3]. The diversity of network connections increases the attack surface, and while processing data closer to its source can reduce exposure, it does not eliminate security risks.

Hardware-level protection is required to offer robust security against cyberattacks. Trusted computing (TC) techniques typically rely on tamper-resistant hardware modules, such as TPMs, but vulnerabilities may be introduced if the module interface is not suitably protected [13]. Alternatively, software-based security technologies are more adaptable but vulnerable if supporting hardware or platform security has been breached, leaving sensitive information such as cryptographic keys exposed.

Particularly, RISC-V-based IoT devices have special security issues since they are open-source and have inherent resource limitations. This calls for the implementation of lightweight yet effective security mechanisms that can achieve a fine balance between robust protection, power usage, and computational power [18]. A trustworthy Root of Trust (RoT) module becomes crucial at this point, especially to boot securely and ensure only legitimate firmware is executed on startup.

Apart from that, standardization and interoperability remain significant challenges, given the multi-vendor and heterogeneous edge device environment. Inconsistencies between hardware platforms, operating systems, and communication protocols pose difficulties to the implementation of one security solution. These inconsistencies make it highly challenging to integrate systems, manage them, scale them, and develop reusable applications [8]. Overcoming these challenges requires advancements in effective resource orchestration, secure hardware-software co-design methods, and the development of open and standardized frameworks specifically for edge-based IoT deployments [4]

6. Future Directions and Research Opportunities in Edge Computing for IoT

6.1 Federated Learning on Edge Devices

Federated learning (FL) enables collaborative model training across distributed edge devices while preserving data privacy, addressing critical IoT challenges such as bandwidth constraints and security risks [7]. It enables collaborative training of shared predictive models among distributed edge devices [11]. Localizing raw data and sharing only model updates will help FL reduce exposure to attacks during transmission [43]. Sensitive raw data is not sent to a central server or cloud for training. Instead, edge devices train the model using their local data and then send only the model updates or parameters. However, deploying ML models, including those used in FL, on resource-constrained edge devices presents significant challenges. Therefore, the resource limitations necessitate lightweight approaches for on-device training and Federated Learning on these constrained devices [7].

6.2 6G and Advanced IoT Connectivity

6G aims to transcend 5G's capabilities by delivering sub-millisecond latency and Tera bps data rates, critical for real-time applications like autonomous vehicles and smart manufacturing [44]. Its ultra-reliable low-latency communication (URLLC) will enable seamless integration of distributed edge AI systems, such as NVIDIA Jetson Nano clusters for traffic analytics [45]. Network slicing and AI-driven resource allocation will optimize bandwidth for heterogeneous IoT workloads, while reconfigurable intelligent surfaces (RIS) enhance signal reliability in non-line-of-sight scenarios [46].

6.3 Edge AI and TinyML Innovations

TinyML democratizes AI for ultra-low-power devices via hardware-software co-design [4], compressing models through quantization and pruning while retaining accuracy [6]. Innovations such as neural architecture search (NAS) [25] automating and speeding up model design to meet quality and inference efficiency requirements on Intel Movidius NCS2. Challenges include balancing model complexity with memory constraints and standardizing benchmarks for cross-platform evaluation [7].

6.4 Sustainable and Green Edge Computing

The integration of renewable energy (RE) sources, such as solar and wind energy, to power edge computing facilities and IoT devices. This approach is seen as a promising way to address challenges like high energy consumption and carbon emissions in computing [47]. The concept is using predictions of renewable energy generation to optimize computing tasks, particularly offloading and scheduling. Predictive analytics, particularly using machine learning algorithms, can enhance energy efficiency in edge computing environments by predicting future workload patterns to optimize resource management [28].

In sustainable edge computing, devices are powered by renewable energy, but the power density of wind and solar energy is temporally dynamic and spatially heterogeneous. The energy harvested may not always allow devices to run at full speed. Managing computing power to adapt to the fluctuations in energy strength is a unique problem in sustainable edge computing. Therefore, optimizing CPU frequency scaling [48] is highlighted as crucial for energy management in sustainable edge computing. Reducing the CPU-cycle frequency can significantly conserve local execution energy.

In summary, advancements in FL, 6G, TinyML, and sustainable design will drive next-gen IoT systems, prioritizing interoperability, adaptability, and efficiency. Interdisciplinary collaboration across ML, networking, and embedded systems is critical to overcoming scalability, security, and energy challenges.

7. Conclusion

Edge computing has emerged as a transformative paradigm in the IoT ecosystem, fundamentally reshaping how data is processed, analyzed, and utilized across various industries. This review has revealed several key findings regarding the evolution and implementation of edge computing in IoT environments. Edge computing brings computation and data storage closer to the source of data generation, overcoming critical limitations of cloud-centric approaches, including high latency, bandwidth constraints, privacy concerns, and energy inefficiency. The paradigm has evolved from the early distributed computing model to systems that incorporate artificial intelligence to enable intelligent decision-making at the edge itself. Edge computing makes the integration of artificial intelligence in resource-constrained IoT environments possible, enabling intelligent analytics and automation at the edge of the network.

Sufficient processing power through a variety of hardware options (microcontrollers, microprocessors and AI accelerators such as GPUs, TPUs and VPUs), energy efficiency through techniques such as data filtering and intelligent task scheduling, ultra-low latency achieved through hardware acceleration and computational offloading, robust connectivity through (various communication technologies, Wi-Fi) have been identified as key requirements for edge computing hardware analysis in IoT settings.

The impact of edge computing on IoT architecture is both exciting and profound. Through real-time processing and decision-making, edge computing has unlocked new applications for manufacturing, healthcare, transportation, and cities that were unimaginable when cloud-centric architectures were the foundation. The decentralized framework has significantly improved data privacy and security by minimizing the transmission of sensitive information. Energy efficiency has been significantly improved by minimizing the amount of data transmission and optimizing local processing, increasing the battery life of IoT devices.

Despite its exciting potential, IoT edge computing has a list of challenges that need to be further investigated. These include scalability limitations in distributed resource management, challenging power vs. performance trade-offs, persistent security threats in the presence of heterogeneous edge networks, and standardization challenges across different hardware platforms. Future research should focus on advancing federated learning techniques to enable collaborative model training while preserving data privacy, exploring the potential of 6G connectivity to further reduce latency to sub-millisecond levels, developing more efficient TinyML models for resource-constrained devices, and designing sustainable edge computing infrastructures powered by renewable energy sources. As edge computing continues to evolve, addressing these challenges will be crucial for realizing its full potential in revolutionizing IoT ecosystems across industries.

Acknowledgement

We acknowledge the research funding from the Malaysia Ministry of Higher Education, Industry Matching Program Grant (IMaP) A183 or Ref. No. JPT.S (BPKI) 2000/018/07/01 Jld5(4).

Conflict of Interest

Authors declare that there is no conflict of interest regarding the publication of the paper.

Author Contribution

The authors confirm contribution to the paper as follows: ***study conception and design:*** *Tan Chee Chen;* ***data curation:*** *Chin Fhong Soon, Mahdzir Jamiaan;* ***draft manuscript preparation:*** *Tan Chee Chen, Mahdzir Jamiaan and Chin Fhong Soon. All authors reviewed the results and approved the final version of the manuscript*

References

[1] H. Kuchuk and E. Malokhvii (2024) Integration of IoT with Cloud, Fog, and Edge Computing: A Review, Advanced Information Systems, vol. 8, no. 2, pp. 65–78, https://10.20998/2522-9052.2024.2.08.

[2] V. M. R. Tummala, S. Korada, S. P. Lingamallu, B. S. Hari, and A. Hazra (2025) TinyML for Edge Networks: Challenges and Future Directions, ICCECE International Conference on Computer, Electrical and Communication Engineering, pp. 1–5, 2025, https://10.1109/ICCECE61355.2025.10941453.

[3] J. Zhang and B. Fan, (2024) Edge Computing in Information Technology: Enhancing Real- Time Data Processing for IoT Applications, vol. 1, no. 1, pp. 1–19.

[4] D. L. Dutta and S. Bharali, (2021) TinyML Meets IoT: A Comprehensive Survey, Internet of Things (Netherlands), vol. 16, no. July, p. 100461, https://10.1016/j.iot.2021.100461.

[5] R. H. Kumar and B. Rajaram, Design and Simulation of an Edge Compute Architecture for IoT-Based Clinical Decision Support System (2024) IEEE Access, vol. 12, no. April, pp. 45456–45474, https:// 10.1109/ACCESS.2024.3380906.

[6] O. Jouini, (2024) A Survey of Machine Learning in Edge Computing: Techniques, Technologies (Basel), vol. 12, no. 6, p. 81, https://doi.org/10.3390/technologies12060081.

[7] P. P. Ray, (2022) A review on TinyML: State-of-the-art and prospects, Journal of King Saud University - Computer and Information Sciences, vol. 34, no. 4, pp. 1595–1623, https:// 10.1016/j.jksuci.2021.11.019.

[8] M. A. Burhanuddin, (2023) Efficient Hardware Acceleration Techniques for Deep Learning on Edge Devices: A Comprehensive Performance Analysis, Khwarizmia, vol. 2023, pp. 103–112, https://10.70470/KHWARIZMIA/2023/010.

[9] M. Mohammadi, A. Abdullah, A. Juneja, I. Rekleitis, M. J. Islam, and R. Zand, (2024) Edge-Centric Real-Time Segmentation for Autonomous Underwater Cave Exploration, Proceedings - 2024 International Conference on Machine Learning and Applications, pp. 1404–1411, 2024, https:// 10.1109/ICMLA61862.2024.00218.

[10] A. Das, S. Patterson, and M. Wittie, (2018) EdgeBench: Benchmarking edge computing platforms, Proceedings - 11th IEEE/ACM International Conference on Utility and Cloud Computing Companion, UCC Companion 2018, pp. 175–180. https://10.48550/arXiv.1811.05948

[11] Y. Abadade, A. Temouden, H. Bamoumen, N. Benamar, Y. Chtouki, and A. S. Hafid, (2023) A Comprehensive Survey on TinyML, IEEE Access, vol. 11, no. 7, pp. 96892–96922, doi: 10.1109/ACCESS.2023.3294111.

[12] F. Messaoudi, A. Ksentini, and P. Bertin, (2017) On Using Edge Computing for Computation Offloading in Mobile Network, in GLOBECOM 2017 - 2017 IEEE Global Communications Conference, pp. 1–7. https:// 10.1109/GLOCOM.2017.8254635.

[13] Y. C. Mohd Khan, Mohsen Hatami, Wenfeng Zhao, (2024) A novel trusted hardware-based scalable security framework for IoT.pdf, Discov. Internet Things, vol. 4,4, 2024, https://10.1007/s43926-024-00056-7. [14]

[14] K. Kubiak, G. Dec, and D. Stadnicka, (2022) Possible Applications of Edge Computing in the Manufacturing Industry—Systematic Literature Review, Sensors, vol. 22, no. 7, https:// 10.3390/s22072445.

[15] O. Pilipczuk, (2024) Overview of edge computing applications in energy sector, Przeglad Elektrotechniczny, vol. 5, no. 5, pp. 240–243, https:// 10.15199/48.2024.05.45.

[16] M. Zhu, M. Liang, H. Li, Y. Lu, and M. Pang, (2023) Intelligent acceptance systems for distribution automation terminals: an overview of edge computing technologies and applications, Journal of Cloud Computing, vol. 12, no. 1, https:// 10.1186/s13677-023-00529-0.

[17] P. R. Nikiema et al., (2023) Towards Dependable RISC-V Cores for Edge Computing Devices, Proceedings - 2023 IEEE 29th International Symposium on On-Line Testing and Robust System Design, IOLTS 2023, pp. 1–7, https:// 10.1109/IOLTS59296.2023.10224862.

[18] N. Waranugraha and M. Suryanegara (2020) The Development of IoT-Smart Basket: Performance Comparison between Edge Computing and Cloud Computing System, 3rd Int. Conf. Comput. Informatics Eng. IC2IE 2020, pp. 410–414, 2020, doi: 10.1109/IC2IE50715.2020.9274596.

[19] S. Lu, J. Lu, K. An, X. Wang, and Q. He, (2023) Edge Computing on IoT for Machine Signal Processing and Fault Diagnosis: A Review, IEEE Internet Things J, vol. 10, no. 13, pp. 11093–11116, https:// 10.1109/JIOT.2023.3239944.

[20] A. Zulkarnain and R. Z. Ikhsan, (2025) Advancing Management Strategies with AI and IoT for Operational Excellence and Competitive Edge, vol. 9, no. 1, pp. 50–60, https://doi.org/10.33050/atm.v9i1.2396

[21] Y. Y. Chen, S. Y. Jhong, S. K. Tu, Y. H. Lin, and Y. C. Wu, (2024) Autonomous Smart-Edge Fault Diagnostics via Edge-Cloud-Orchestrated Collaborative Computing for Infrared Electrical Equipment Images, IEEE Sens J, vol. 24, no. 15, pp. 24630–24648, https:// 10.1109/JSEN.2024.3415639.

[22] P. Schulz and G. Sleahtitchi (2023) FPGA-based Accelerator for FFT-Processing in Edge Computing, Proc. IEEE Int. Conf. Intell. Data Acquis. Adv. Comput. Syst. Technol. Appl. IDAACS, vol. 1, pp. 590–595, doi: 10.1109/IDAACS58523.2023.10348654.

[23] I. K. Kasmeridis and V. V. Dimakopoulos, (2022) OpenMP Offloading in the Jetson Nano Platform, ACM International Conference Proceeding Series, https:// 10.1145/3547276.3548517.

[24] Q. Xu, V. Li, and C. D. S, (2023) Neural Architecture Search for Intel Movidius VPU, pp. 1–10, https://doi.org/10.48550/arXiv.2305.03739

[25] M. Rafiquzzaman (2018) Microcontroller Theory and Applicationswith The PIC18F. John Wiley & Sons, Inc. (page. 1-5)

[26] A. Fanariotis, T. Orphanoudakis, K. Kotrotsios, V. Fotopoulos, G. Keramidas, and P. Karkazis, (2023) Power Efficient Machine Learning Models Deployment on Edge IoT Devices, Sensors, vol. 23, no. 3, https:// 10.3390/s23031595.

[27] R. R. Asaad, A. A. Hani, A. B. Sallow, S. M. Abdulrahman, H. B. Ahmad, and R. M. Subhi, (2024) A Development of Edge Computing Method in Integration with IOT System for Optimizing and to Produce Energy Efficiency System, 2024 4th International Conference on Advance Computing and Innovative Technologies in Engineering, ICACITE 2024, pp. 835–840, , https:// 10.1109/ICACITE60783.2024.10617436.

[28] Y. Chen, N. Zhang, Y. Zhang, X. Chen, W. Wu, and X. S. Shen, TOFFEE: Task Offloading and Frequency Scaling for Energy Efficiency of Mobile Devices in Mobile Edge Computing, (2021) IEEE Transactions on Cloud Computing, vol. 9, no. 4, pp. 1634–1644, https://10.1109/TCC.2019.2923692.

[29] K. Krishna and P. Brahmaji, (2024) Edge Computing and Analytics for IoT Devices: Enhancing Real-Time Decision Making in Smart Environments, vol. 6, no. 5, pp. 1–9. https://10.2139/ssrn.5012466.

[30] M. Singh and S. Sankaran, Lightweight Security Architecture for IoT Edge Devices, (2022) Proceedings - 2022 IEEE International Symposium on Smart Electronic Systems, pp. 455–458, https:// 10.1109/iSES54909.2022.00099.

[31] M. S. Islam, H. Verma, L. Khan, and M. Kantarcioglu, (2019) Secure real-time heterogeneous IoT data management system, Proceedings - 1st IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications, TPS-ISA 2019, pp. 228–235, 2019, https:// 10.1109/TPS-ISA48467.2019.00037.

[32] K. M. Sadique, R. Rahmani, and P. Johannesson, (2020) Identity management and secure communication for edge IoT devices, Sensors (Switzerland), vol. 20, no. 22, pp. 1–38, 2020, https:// 10.3390/s20226546.

[33] M. O. Silva et al. (2022) Action and Assembly Time Measurement System of Industry Workers using Jetson Nano, Proc. - 2022 IEEE Int. Conf. Consum. Electron. - Taiwan, ICCE-Taiwan, pp. 319–320, 2022, doi: 10.1109/ICCE-Taiwan55306.2022.9869028.

[34] S. Faiazuddin, M. V. Lakshmaiah, K. T. Alam, and M. Ravikiran, (2020) IoT-based Indoor Air Quality Monitoring system using Raspberry Pi4, Proceedings of the 4th International Conference on Electronics, Communication and Aerospace Technology, pp. 714–719, 2020, https://10.1109/ICECA49313.2020.9297442.

[35] V. Melikyan, T. Khachatryan, D. Galstyan, and E. Harutyunyan, (2024) Efficient Vision Transformer Deployment on Google Coral Through Knowledge Distillation, IEEE East-West Des. Test Symp. EWDTS 2024, pp. 1–3, 2024, doi: 10.1109/EWDTS63723.2024.10873747.

[36] N. Gabdullin and A. Raskovalov, (2023) Google Coral-based edge computing person reidentification using human parsing combined with analytical method, Internet of Things (Netherlands), vol. 22, no. Dl, pp. 1–11, doi: 10.1016/j.iot.2023.100701

[37] T. B. Khachatryan and D.F. Davtyan, (2022) Depth Estimation Ai Inferencing Comparison of Jetson Xavier Nx and Coral Dev Board, Proc West Mark Ed Assoc Conf, pp. 72–80, https:// 10.53297/0002306x-2022.v75.1-72.

[38] J. Winzig, J. C. A. Almanza, M. G. Mendoza, and T. Schumann, (2022) Edge AI - Use Case on Google Coral Dev Board Mini, Proceedings - 2022 IET International Conference on Engineering Technologies and Applications, pp. 12–13, https:// 10.1109/IET-ICETA56553.2022.9971614.

[39] N. L. Gimenez, F. Freitag, J. K. Lee, and H. Vandierendonck, (2022) Comparison of Two Microcontroller Boards for On-Device Model Training in a Keyword Spotting Task, 2022 11th Mediterranean Conference on Embedded Computing, MECO 2022, pp. 7–10, https:// 10.1109/MECO55406.2022.9797171.

[40] T.Lehikoinen (2023) AWS Greengrass in Building IoT Applications, SAVONIA UNIVERSITY OF APPLIED SCIENCES

[41] N. A. Othman and I. Aydin, (2018) A New Deep Learning Application Based on Movidius NCS for Embedded Object Detection and Recognition, 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies, Proceedings, pp. 1–5, https:// 10.1109/ISMSIT.2018.8567306.

[42] J. L. H. Ramos and A. Skarmeta, (2020) Security and Privacy on the Internet of Things: Challenges and Solutions, IOP Press, p. 204.

[43] A. Ticku et al., (2024) Next-Gen IoT: 5G Realities and 6G Possibilities, 15th International Conference on Computing Communication and Networking Technologies, pp. 1–8, 2024, https:// 10.1109/ICCCNT61001.2024.10724540.

[44] A. Kumar, M. Masud, M. H. Alsharif, N. Gaur, and A. Nanthaamornphong, (2025) Integrating 6G technology in smart hospitals: challenges and opportunities for enhanced healthcare services, Front Med (Lausanne), vol. 12, no. 1, https:// 10.3389/fmed.2025.1534551.

[45] J. Du, M. Xu, S. S. Gill, and H. Wu, (2024) Computation Energy Efficiency Maximization for Intelligent Reflective Surface-Aided Wireless Powered Mobile Edge Computing, IEEE Transactions on Sustainable Computing, vol. 9, no. 3, pp. 371–385, https:// 10.1109/TSUSC.2023.3298822.

[46] M. Alhartomi, A. Salh, L. Audah, S. Alzahrani, and A. Alzahmi, (2024) Enhancing Sustainable Edge Computing Offloading via Renewable Prediction for Energy Harvesting, IEEE Access, vol. 12, pp. 74011–74023, https:// 10.1109/ACCESS.2024.3404222.

[47] Y. Luo, L. Pu, and C. H. Liu, (2023) CPU Frequency Scaling Optimization in Sustainable Edge Computing, IEEE Transactions on Sustainable Computing, vol. 8, no. 2, pp. 194–207, https:// 10.1109/TSUSC.2022.3217970.

[48] D. Franklin (2019, March 18) Jetson Nano Brings AI Computing to Everyone. Nvidia Developer. https://developer.nvidia.com/blog/jetson-nano-ai-computing/

[49] V. Shrimali (2019, January 6) Using OpenVINO with OpenCV. Learn OpenCV. https://learnopencv.com/using-openvino-with-opencv/