UDC:
004.78:004.35
004.85:004.738.5
10.62907/juuntics260101013s

Edge AI and TinyML in IoT Systems: A Review of Applications, Architectures and Limitations

Lazar Stošić[1,2], Željko Stanković[1], Olja Krčadinac[1]*

[1]Faculty of Informatics and Computer science, University Union—Nikola Tesla, Belgrade, Serbia
e-mail: lstosic@unt.edu.rs, makijanac1@gmail.com, okrcadinac@unionnikolatesla.edu.rs
[2]Don state technical university, Rostov on Don, Russia

Abstract: With the rapid development of the Internet, the need for fast and energy-efficient data processing has also increased. In this regard, Edge Artificial Intelligence and Tiny Machine Learning represent significant technological approaches that enable the execution of machine learning models on resource-constrained edge devices, embedded platforms, and microcontrollers. The aim of this review is to analyze the role of Edge AI and TinyML technologies in IoT (Internet of Things) systems, with special reference to their applications, architectural models and key limitations. The paper provides a concise review of scientific and professional literature addressing edge computing, embedded machine learning, intelligent sensors, and IoT architectures. Through a review of numerous literatures, major application areas were identified, including smart homes, smart classrooms, health monitoring, wearables, industrial IoT, predictive maintenance, smart agriculture, and environmental monitoring. Special attention is paid to architectural models, from cloud-centric IoT systems to edge-assisted and fully embedded TinyML architectures. Analysis shows that Edge AI and TinyML can significantly reduce latency, improve privacy, reduce network traffic consumption, and enable real-time decision making. However, their application is limited by small memory, lower processing power, energy consumption, model optimization, security risks, interoperability and maintenance of remote devices.

Keywords: *Edge AI; TinyML; Internet of Things; embedded systems; machine learning; edge computing; intelligent sensors.*

Introduction

Early IoT systems were primarily developed to connect physical objects, sensors, and actuators to the Internet, enabling data collection, transmission, and basic data processing. These systems typically relied on cloud infrastructure, where sensor data were transferred to remote servers for storage, analytics, and decision-making. This method enabled mass connection of devices and centralized processing of large amounts of data. On the other hand, the issue of latency, consumption of network traffic, energy efficiency, privacy and reliability in real time was emerged For this reason, modern IoT systems have increasingly developed towards intelligent distributed systems that can locally analyze data, recognize patterns and make decisions closer to the point of their origin [1, 2].

In the world of modern IoT systems, the concept of Edge AI occupies a special place. It refers to the integration of artificial intelligence and machine learning algorithms on devices or nodes located at the edge of the network. Instead of sending the entirety of the data to the cloud, Edge AI allows part of the processing to be performed locally, on smart sensors, gateway devices, mobile devices, embedded platforms or industrial edge servers. Therefore, IoT systems gained the ability to respond faster, local analytics and adaptive behavior in real time. Numerous recent studies indicate that Edge AI is an important direction of development as it moves AI functionality from centralized data centers to distributed devices, thereby increasing the availability of intelligent services in smart homes, healthcare, industrial IoT, education, autonomous systems and smart cities [3, 4].

Tiny Machine Learning (TinyML) represents a narrower and more specific approach within the broader concept of Edge AI. It refers to the application of machine learning models on devices with very limited hardware resources (microcontrollers, ultra-low-power sensor nodes and small embedded systems). TinyML aims to enable the execution of optimized models on devices with low power consumption, limited memory and minimal computing capacity. Ray [5] points out that TinyML represents the transition of machine learning from high-performance systems to resource-constrained embedded

*Corresponding author: lstosic@unt.edu.rs

devices at the edge of the network, while Elhanashi et al. [6] point out that TinyML enables the integration of machine learning into low-cost, energy-efficient and IoT-oriented devices.

Edge AI and TinyML reduce the need for network traffic and constant cloud communication, which contributes to lower operating costs and greater system resilience in conditions of unstable internet connection. Optimized models on edge and tiny devices can contribute to reducing power consumption, which is especially important for battery-powered IoT devices and long-term sensor networks [7, 6].

In addition to its positive aspects, the application of Edge AI and TinyML in IoT systems also has its limitations. These challenges are mainly associated with the resource limitations of edge and embedded devices, the need to optimize and compress machine learning models for constrained environments, and the requirement to ensure security, interoperability, maintainability, and reliable software updates in distributed IoT systems. The biggest problem is balancing between model accuracy, latency, power consumption and model size. Therefore, it is necessary to systematize existing knowledge about architectures, application areas and limitations of these technologies. The transition from cloud-centric IoT systems to Edge AI and TinyML approaches is not only a technological improvement, but also an architectural change in the way intelligent systems are designed.

The aim of this paper is to provide an overview of contemporary approaches to Edge AI and TinyML technologies in IoT systems, with a particular focus on their architectures, application areas, advantages and key limitations. The paper seeks to explain how IoT systems are transformed from passive networks of sensors into intelligent, distributed and energy-efficient systems capable of local data processing and real-time decision-making.

The contribution of this work is reflected in the systematization of modern knowledge about the application of Edge AI and TinyML technologies in IoT systems, with special emphasis on their role in the transformation of traditional sensor networks into intelligent, distributed and energy-efficient systems. In contrast to approaches that consider IoT mainly as an infrastructure for data collection and transmission, this paper emphasizes the importance of local intelligence, i.e. the ability for devices at the edge of the network to independently process data, execute machine learning models and make decisions in real time.

Finally, the paper contributes to the formation of a theoretical and practical framework for future research in the field of intelligent IoT systems. The overview can be used by researchers, engineers, teachers and students as a basis for understanding the current trends in the application of artificial intelligence at the edge of the network, as well as for the development of new solutions that combine low energy consumption, local data processing, security and reliable real-time decision-making.

Internet of Things Systems

Internet of Things Systems represent networks of physical devices, sensors, actuators, software components and communication technologies that enable the collection, exchange and processing of data from the physical environment. Electronics are increasingly integrated into everyday physical objects, resulting in smart devices that have become an unimaginable part of everyday existence [8]. The development of new technology, the high-speed Internet protocol, has also led to an increase in multimedia traffic, which initially changed the meaning of IoT to the so-called Multimedia Internet of Things (mIoT). The growing demands of modern IoT applications, particularly in terms of bandwidth, latency, and real-time data processing, have significantly transformed the architecture of these systems. [9]. In a typical IoT system, sensors collect empirical environmental data such as temperature, humidity, motion, pressure, light, air quality, or other parameters, while actuators perform certain actions, such as turning on devices, regulating lighting, controlling motors, or activating alarms. Contemporary overviews of the IoT field emphasize that IoT systems are based on the integration of heterogeneous devices, communication protocols, software platforms and applications that enable monitoring, control and automated management of physical processes [10, 11].

Traditional IoT systems are usually based on a multi-layered architecture that includes a device layer, a communication layer, a data processing layer and an application layer. The device layer includes sensors, microcontrollers and embedded platforms, such as Arduino, ESP32 or Raspberry Pi.

The communication layer enables data transfer using technologies such as Wi-Fi, Bluetooth, Zigbee, LoRaWAN, MQTT, HTTP and other network technologies. The data processing layer can be implemented in the cloud infrastructure, on edge devices or local gateway nodes, while the application layer enables users to monitor, visualize and manage the system. Modern IoT architectures increasingly combine cloud, edge and fog computing approaches to enable scalability, interoperability, lower latency and more efficient data processing [12, 10].

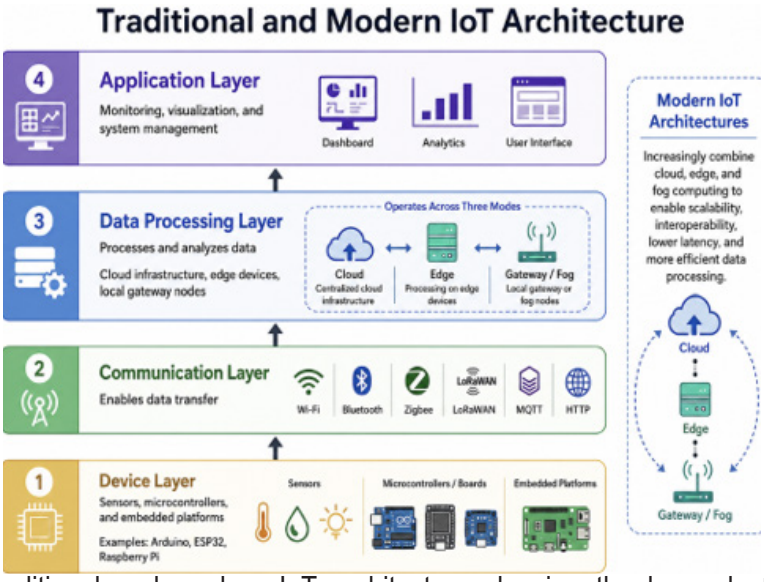

Figure 1. Traditional and modern IoT architecture showing the layered structure of device, communication, data processing, and application components.

The figure 1 presents a layered view of traditional and modern IoT architecture. At the device layer, sensors, microcontrollers, and embedded platforms collect data from the physical environment. The communication layer enables data transmission through technologies such as Wi-Fi, Bluetooth, Zigbee, LoRaWAN, MQTT, and HTTP. The data processing layer may rely on cloud infrastructure, edge devices, or gateway/fog nodes, depending on the required level of latency, scalability, and local processing. Finally, the application layer provides dashboards, analytics, user interfaces, monitoring, visualization, and system management. The figure also highlights the shift from purely cloud-centric IoT systems toward modern architectures that combine cloud, edge, and fog computing to improve responsiveness, interoperability, and processing efficiency.

Contemporary IoT systems have moved beyond passive data collection networks and are increasingly developing into intelligent, distributed systems capable of local analytics, autonomous decision-making, and adaptive behavior. This transformation is particularly significant in the fields of smart homes, smart cities, industrial IoT, healthcare, education, agriculture and environmental monitoring.

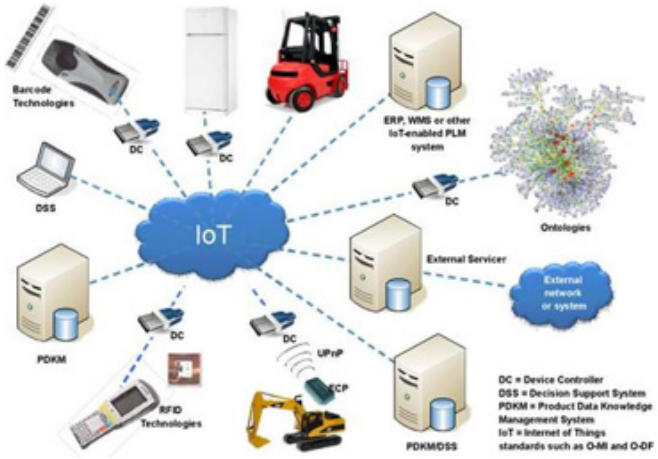

Figure 2. IoT architecture showing the integration of physical devices, sensors, identification technologies, enterprise systems, external services, and networked platforms through a central IoT infrastructure. (Source: https://share.google/h1tKj9QPw08yv1Wnh)

The figure 2 illustrates the role of IoT as a central infrastructure for connecting heterogeneous devices and systems. Physical objects, sensors, RFID and barcode technologies, industrial machines, decision support systems, enterprise platforms, external services, and network systems are linked through IoT-enabled communication mechanisms. Such an architecture allows data to be collected from different sources, transferred through device controllers and network interfaces, and integrated with information

systems for monitoring, automation, analytics, and decision-making.

The integration of Edge AI and TinyML technologies further expands the capabilities of IoT systems, as it enables data processing and decision-making to be performed closer to the data source. In this way, latency, dependence on cloud infrastructure and consumption of network traffic are reduced, while at the same time privacy, autonomy and energy efficiency of the system are increased. This direction of development is confirmed by works on edge-computing IoT architectures and TinyML approaches, which indicate that intelligence is increasingly moving towards devices at the edge of the network and resource-constrained IoT platforms [12, 5, 2].

Edge Artificial Intelligence

Edge Artificial Intelligence (Edge AI) is an approach in which artificial intelligence and machine learning algorithms are executed on devices or nodes located closer to the point of origin of data. Edge AI enables IoT devices to independently analyze data and make real-time decisions. For this reason, we can often meet with the term "edge intelligence", which implies moving AI functionality from centralized data centers to the edge of the network, i.e. closer to users, devices and data sources [4, 13].

Traditional IoT systems are often limited by delayed response, dependence on continuous connectivity, and challenges associated with processing sensitive data in remote infrastructures. Edge AI mitigates some of these limitations by enabling trained models to operate close to the data source. This development has supported the emergence of the Artificial Intelligence of Things (AIoT), where IoT infrastructure and AI-based methods are integrated to provide local analysis, autonomous decision-making, and adaptive system behavior. [2, 13].

Figure 3. Edge AI computing architecture showing the integration of cloud platforms, Edge AI devices, IoT gateways, and application domains for local data processing and low-latency IoT services. (Sources: https://www.ttx.ca/en-CA/blog/blog06-edgeAI-computing-knowabout)

Figure 3. Edge AI computing architecture showing the relationship between cloud platforms, edge AI computing devices, IoT gateways, and application domains such as warehouse and logistics, vehicle diagnostics, oil and gas, Industry 4.0, and smart city systems. The figure illustrates how edge computing devices operate between cloud infrastructure and end-user applications, enabling local data processing, reduced latency, and more efficient deployment of AI-based IoT services.

The main advantages of the Edge AI approach are reflected in reduced latency, lower consumption of network traffic, greater privacy and better system reliability. By processing data locally, such systems significantly reduce reaction time to environmental changes, which is particularly important in industrial automation, health monitoring, autonomous vehicles, security systems, and smart classroom environments. Also, sensitive data can remain on the local device, reducing the risk of unauthorized access during transmission over the network. Edge computing generally brings compute and storage closer to where data is produced and used, which improves the performance of applications that require low latency, mobility, privacy and security [1, 4].

Despite their advantages, IoT and Edge AI also present certain limitations that need to be

considered. Edge devices typically have less computing resources than cloud servers, so AI models must be optimized to work under conditions of limited memory, processing power, and energy. Therefore, techniques such as model compression, quantization, pruning and efficient neural networks are used. In IoT systems, Edge AI represents an important step towards intelligent, autonomous and distributed systems that do not depend solely on centralized cloud infrastructure [2, 13].

Tiny Machine Learning

Tiny Machine Learning (TinyML) is a field of machine learning that deals with the execution of optimized AI models on very small and resource-limited devices (microcontrollers, sensor nodes, embedded platforms and low-power IoT devices). TinyML allows machine learning models to run directly on devices with very limited memory, processing power and power consumption [5, 7]. Ray [5] particularly emphasizes that TinyML is being developed at the intersection of machine learning, edge computing and the Internet of Things, as it enables the application of intelligent algorithms on resource-limited devices at the edge of the network.

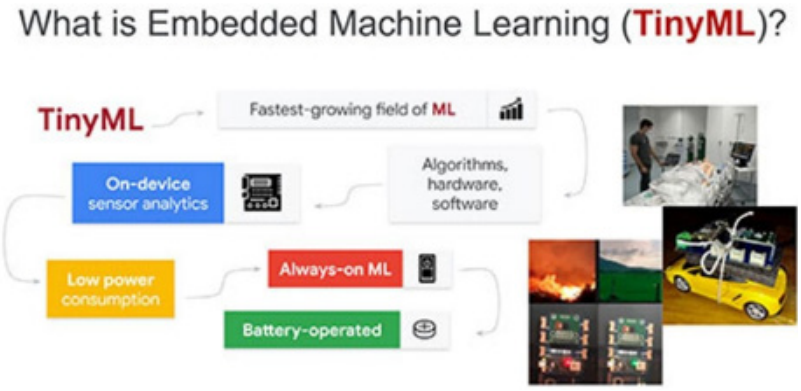

Figure 4. Overview of TinyML as an embedded machine learning approach that enables low-power, always-on, on-device analytics on resource-constrained devices.

The figure 4 explains the basic concept of embedded machine learning, commonly referred to as TinyML. It shows that TinyML enables machine learning models to be deployed directly on small, resource-constrained embedded devices rather than relying on remote cloud infrastructure. Such systems are designed for on-device sensor analytics, low power consumption, battery-powered operation, and continuous or always-on processing. By combining algorithms, hardware, and software, TinyML allows local data analysis and real-time decision-making in applications such as healthcare monitoring, environmental sensing, smart mobility, and industrial systems.

Basically, TinyML approach is based on the fact that simpler and optimized models can be used for local data processing and decision-making without the constant need to send information to the cloud platform. Schizas et al. [7] point out that TinyML enables ultra-low power consumption and local analytics in large-scale IoT implementations, while Elhanashi et al. [6] emphasize its role in the transformation of embedded and IoT systems through local data processing and reduced dependence on cloud infrastructure. In practice, TinyML is often implemented using platforms and tools such as TensorFlow Lite for Microcontrollers, Edge Impulse, Arduino AI libraries, and various development frameworks for ESP32, ARM Cortex-M, and other microcontroller platforms. TensorFlow Lite for Microcontrollers is designed to run machine learning models on microcontrollers, DSPs, and other memory-constrained devices, making it one of the most widely used frameworks for TinyML applications [14, 7].

The importance of TinyML technology in IoT systems is reflected in the ability to move intelligence directly to the end device. This reduces latency, saves energy, reduces network traffic and increases user privacy, as data can be processed locally. In terms of disadvantages, TinyML has limitations in terms of memory, processing power, model complexity, prediction accuracy, model maintenance and updating on a large number of distributed devices. Despite its limitations, TinyML represents one of the key directions in the development of future intelligent, autonomous and energy-efficient IoT systems.

Relationship Between Cloud AI, Edge AI and TinyML

Cloud AI, Edge AI, and TinyML represent different levels of data processing and AI model execution. In practice, they are often used together: cloud infrastructure provides high computing power and scalability, edge devices enable processing closer to the data source, while TinyML enables the execution of optimized models directly on microcontrollers and other devices with very limited resources [2, 5, 13].

Cloud AI represents a centralized approach in which data from IoT devices is sent to remote servers or data centers, where storage, model training, advanced analytics and decision-making are performed. Cloud-based processing is well suited for large datasets and complex deep learning models due to the substantial computing, memory, and storage capacity provided by cloud environments. However, its applicability is limited in scenarios requiring immediate response, as data transmission to remote servers and the subsequent return of results may increase latency. In addition, constantly sending data to the cloud increases the consumption of network traffic and can raise privacy and security issues, especially when processing sensitive data from health, education or surveillance systems [1, 2].

Edge AI is emerging as an intermediate level between the centralized cloud and the end IoT devices. It involves the execution of AI algorithms and machine learning models on devices located closer to the point of origin of data, such as gateway devices, local servers, smartphones, industrial controllers, embedded computers or network edge nodes. In the literature, Edge AI is explained as an approach that moves AI calculations from centralized cloud data centers to the network periphery, i.e. closer to users, devices and data sources [4, 13]. Therefore, model optimization, compression, quantization, pruning, and distributed processing approaches are often used in Edge AI systems [4, 2, 13].

TinyML represents the most local form of intelligence in this continuum. While Edge AI can involve relatively more powerful devices, such as gateway computers or local edge servers, TinyML refers to executing machine learning on very small and power-constrained devices, most often microcontrollers and embedded platforms. TinyML moves machine learning from traditional high-performance systems to resource-constrained embedded devices at the edge of the network, thereby reducing dependence on cloud infrastructure and enabling local data processing on IoT devices [5, 6, 7].

In a hierarchical sense, Cloud AI can be seen as a tier for training large models, long-term storage and global analytics; Edge AI as a layer for local inference, aggregation and rapid data processing; and TinyML as a level for direct execution of simpler models on the sensor or microcontroller itself.

Therefore, it is most accurate to consider Cloud AI, Edge AI and TinyML as complementary layers of intelligence in modern IoT systems. Cloud provides power and scalability, Edge AI provides local autonomy and fast response, while TinyML enables ultra-low power consumption and intelligence directly on the end device. The key challenge is to determine the optimal location of data processing depending on the application requirements, available resources, security requirements, privacy needs, power consumption and expected response time [1, 7, 6].

Applications of Edge AI and TinyML

Unlike traditional cloud-centric systems, in which data is sent to remote servers, Edge AI and TinyML enable part of the intelligence to be moved closer to the data source, i.e. to gateway devices, embedded platforms, microcontrollers or sensor nodes themselves. This approach is particularly important for applications that analyze sound, movement, vibrations, low-resolution images and various sensor data, because classification and detection can be performed locally, without constant dependence on cloud infrastructure [5, 7, 13].

Smart Homes and Smart Buildings

In smart homes and smart buildings, Edge AI and TinyML enable local automation, energy-efficient management and personalized services. IoT devices can collect data on temperature, humidity, lighting, user presence, air quality and energy consumption, while local AI models can make decisions about regulating heating, ventilation, lighting or security alarms. TinyML is particularly suitable for voice control, motion detection, recognition of simple acoustic signals, and local classification of sensor data on low-power devices.

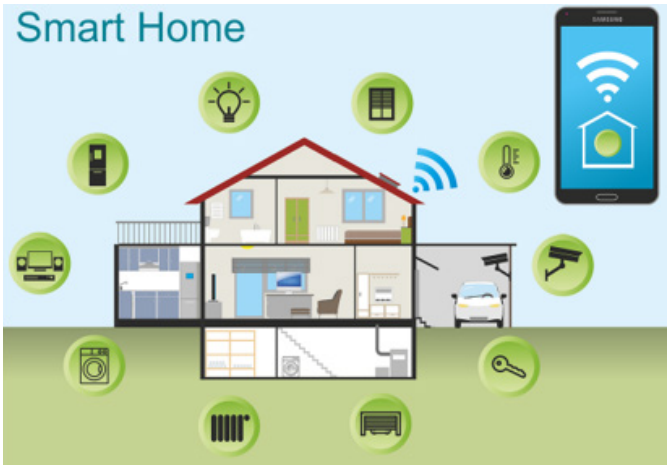

Figure 5. Smart home IoT architecture illustrating the integration of household devices, sensors, and appliances through wireless connectivity and mobile-based control. (Source: https://share.google/tFirL5Nfl84jPJEdS)

The figure 5 shows how smart lighting, heating, security cameras, access control, household appliances, and environmental sensors can be connected within a residential environment to enable remote monitoring, automation, and centralized management through a smartphone application.

Smart home systems offer solutions for affordable and efficient home automation. With these improvements, the system has the potential to bring smart home technology into everyday use for a wide range of households, promoting integrated, more energy-conscious living environments [15].

Smart Classrooms and Educational Environments

In smart classrooms and educational environments, Edge AI and TinyML can contribute to monitoring learning conditions, improving comfort and supporting adaptive education systems. IoT sensors can monitor temperature, humidity, noise level, lighting, air quality and space occupancy, while local models can classify classroom conditions as optimal, warning or critical. The advantage of local processing is especially evident in situations where a quick response is required, for example in case of poor air quality, excessive noise or increased concentration of students in the space.

Figure 6. Smart classroom environment supported by IoT-based interactive learning technologies Source: https://share.google/WLrg2GiQgHD78H0JO

The figure 6 presents an example of a smart classroom in which IoT-enabled devices, interactive displays, and cloud-based services are integrated into the teaching and learning process. Students use connected devices to access and share learning materials, while the teacher controls and presents content through an interactive display. Features such as wireless device casting, multi-touch control, real-time collaboration, cloud storage, and lesson recording support more flexible, interactive, and data-driven learning environments.

Edge AI allows data from multiple classrooms to be processed on a local gateway device, while TinyML allows simpler classification directly on a microcontroller or sensor node. This approach reduces network traffic and increases privacy, because raw data from the educational space does not have to be constantly sent to the cloud [2, 13].

Healthcare and Wearable Devices

Healthcare and wearable devices represent one of the most important areas of application of Edge AI and TinyML technologies. Smart watches, wristbands, heart rate sensors, accelerometers, temperature sensors and other wearable devices can locally analyze the user's physiological and behavioral data. TinyML makes it possible to perform movement classification, fall detection, physical activity analysis, basic heart rhythm assessment or anomaly detection on the device, without the need to send all sensitive health data to the cloud. This is especially important for privacy, latency reduction and energy savings. One study [15] highlighted that the encapsulated information collected by UNTLab in a hazardous (contaminated) environment can be of value in critical situations, without exposing human resources to danger. This is another proof of the advantages of using these technologies in protecting and exposing human lives.

Figure 7. Wearable IoT-based healthcare system showing the connection between patient-generated data, wearable and medical monitoring devices, cloud infrastructure, and electronic patient health records.
Source: https://techno-soft.com/wearable-technology-in-healthcare.html

The figure 7 presents a healthcare IoT scenario in which wearable devices and medical sensors collect patient-related data, including glucose level, physical activity, calories, weight, medication, and nutrition information. These data can be transferred through wireless communication technologies such as Bluetooth, Wi-Fi, and USB-based connections to cloud infrastructure, where they may be stored, processed, and integrated with patient health records. Such systems support continuous health monitoring, remote patient management, and more timely access to health-related information by healthcare professionals.

Recent reviews indicate that TinyML has significant potential for continuous health monitoring and wearable applications, as it enables local inference on devices with strict memory, power, and response time constraints [16, 17].

Industrial IoT and Predictive Maintenance

In industrial IoT, Edge AI and TinyML are most often applied for predictive maintenance, machine monitoring, anomaly detection, quality control and optimization of production processes. Industrial sensors collect data on vibration, temperature, pressure, sound, energy consumption and equipment condition. Edge AI allows this data to be analyzed locally, close to the production line, reducing reaction time and enabling faster fault detection. TinyML can be useful for simpler local detection of vibration patterns, acoustic anomalies or changes in machine operation on low-power sensor nodes.

Figure 8. Predictive maintenance in an industrial IoT environment, showing the use of connected sensors, robotic systems, and digital monitoring dashboards for real-time equipment supervision. Source: https://neurosys.com/blog/predictive-maintenance-with-iot-technology

The figure 8 presents an example of predictive maintenance supported by IoT technology in an industrial environment. Sensors and connected machines continuously collect operational data, such as temperature, vibration, performance indicators, and equipment status. These data can be analyzed through digital dashboards or cloud/edge platforms to identify abnormal patterns, predict possible failures, and schedule maintenance activities before serious faults occur. Such an approach improves reliability, reduces downtime, and supports more efficient production management.

The application of the Edge AI approach in predictive maintenance is particularly significant because it reduces cloud dependency, increases system reliability and enables local decision-making in real time [18, 12, 13].

Smart Agriculture and Environmental Monitoring

Smart agriculture and environmental monitoring are also important application domains for Edge AI and TinyML technologies. IoT sensors can be used to monitor soil moisture, temperature, air humidity, light intensity, air quality, gas concentrations, water levels, and other environmental parameters. Edge AI supports local data analysis on farms, in greenhouses, and at remote monitoring stations, whereas TinyML enables small sensor devices to classify plant conditions locally, detect anomalies in microclimatic conditions, and trigger irrigation systems when necessary.

Figure 9. IoT-based smart agriculture solution illustrating the use of connected sensors, mobile applications, and digital monitoring tools for crop and environmental management. Source: https://cepdnaclk.github.io/e20-3yp-Smart-Environmental-Monitoring-System-for-Palm-Oil-Plantation/

The figure 9 illustrates a smart agriculture scenario in which IoT technologies are used to monitor crop and environmental conditions through connected sensors and mobile applications. Parameters such as crop yield, crop damage, farm mapping, soil erosion, and water stress can be collected and visualized in real time, allowing farmers to make more informed decisions. When combined with Edge AI or TinyML, such systems can also support local analysis, anomaly detection, irrigation control, and early identification of plant or environmental stress.

The literature provides examples of TinyML systems for autonomous plant watering, as well as the application of models for plant disease detection and agricultural data analysis on resource-constrained devices [19, 5].

In addition to the mentioned areas, Edge AI and TinyML are increasingly used in security systems and smart cities, where fast local detection of events, such as movement, sound, traffic anomalies, changes in air quality or unusual behavior patterns, is required. In all these applications, the core value of the Edge AI and TinyML approach is reflected in the fact that they enable the transition from passive data collection to intelligent, autonomous and contextual decision-making at the level of the IoT system itself [2, 7, 13].

Architectures of Edge AI and TinyML in IoT Systems

A typical data flow in a modern IoT system can be shown as: sensors and actuators → microcontrollers/embedded devices → edge gateway → cloud platform. Depending on the needs of the application, the AI model can be executed at different levels: in the cloud, on the edge gateway device, on the embedded computer or directly on the microcontroller/sensor. IoT architecture is increasingly developing in the direction of combining cloud, edge and embedded layers in order to achieve better scalability, lower latency, lower consumption of network traffic and greater system autonomy [12, 10, 2].

Cloud-Centric IoT Architecture

Cloud-centric IoT architecture represents a traditional model in which sensors and IoT devices collect data and send it to a remote cloud platform, where storage, processing, model training and decision-making are performed. In this model, the AI model is most often executed in a cloud environment, while local devices mainly have the role of data collection and transmission. The advantage of this architecture is high computing power, the ability to process large data sets and centralized system management. However, the main disadvantages are higher latency, dependence on a stable Internet connection, increased consumption of network traffic and potential privacy risks, as raw data is often transmitted outside the local environment [10, 1].

Edge-Assisted IoT Architecture

Edge-assisted IoT architecture introduces an intermediate layer between the end IoT devices and the cloud platform. In this model, data is first processed on edge gateway devices, local servers, industrial controllers or embedded computers, while only selected, aggregated or long-term significant data is sent to the cloud. This approach reduces latency and network load, while retaining the benefits of the cloud for long-term storage, advanced analytics, and model updates. Edge computing architectures are especially important for IoT applications that require fast response, such as industrial IoT, smart cities, health monitoring and security systems [12, 13].

TinyML-Based Embedded Architecture

TinyML-based embedded architecture represents the most local form of intelligence in IoT systems. In this approach, the AI model is executed directly on a microcontroller, sensor node, or other device with very limited resources. This means that the device does not have to constantly send raw data either to the gateway or to the cloud, but can locally classify the signal, detect an anomaly, recognize a pattern or activate a certain reaction. The TinyML architecture requires a high degree of model optimization, including quantization, compression, pruning, and parameter reduction, because microcontrollers have limited memory, processing power, and energy resources [5, 7, 6].

Comparatively, a cloud-centric architecture is suitable when large computing power and centralized analysis are needed, but it is less suitable for applications that require minimal latency and local privacy.

Figure 10. Comparative overview of cloud-centric IoT, edge-assisted IoT, fully Edge AI, and TinyML architectures, highlighting their execution environments, main advantages, and key limitations.

The figure 10 compares four architectural approaches used in IoT and intelligent edge systems. Cloud-centric IoT relies on remote cloud platforms and offers strong computing and storage capacity, but may introduce higher latency, internet dependency, and privacy concerns. Edge-assisted IoT moves part of the processing to gateways, local servers, or embedded computers, reducing latency and limiting the amount of data sent to the cloud. Fully Edge AI performs inference directly on edge devices and local nodes, enabling faster decisions and greater autonomy, although it requires model optimization and may face scalability and maintenance challenges. TinyML represents the most resource-constrained approach, where lightweight models run on microcontrollers, sensors, or low-power embedded devices, supporting offline operation and strong privacy but with limited memory, computing power, and model complexity.

Based on this comparison, it can be concluded that there is no universally best architecture for all IoT systems. The choice of architectural model depends on the nature of the application, required response speed, available hardware resources, security requirements, data volume, power consumption and privacy needs. In modern solutions, hybrid models are most often applied, in which the cloud, edge and TinyML layers function as complementary levels of intelligence.

Limitations and Challenges

Although Edge AI and TinyML significantly improve IoT systems through local data processing, lower latency and greater autonomy, their implementation is accompanied by numerous technical, security and organizational limitations. The most important challenges are related to limited hardware resources, energy consumption, model optimization, data protection, interoperability, standardization and maintenance of distributed devices. It is especially important to emphasize that designing Edge AI and TinyML solutions always involves a compromise between model accuracy, latency, energy consumption and model size. A more accurate model often requires more memory and processing power, while smaller and more energy efficient models may have lower accuracy or limited generalizability [5, 7, 6].

Hardware and Memory Constraints

The first limitation refers to the hardware and memory capacities of the device. Edge devices have significantly fewer resources than cloud servers, while TinyML devices are even more limited because models are executed on microcontrollers, sensor nodes, and embedded platforms. Such devices often have limited working memory, less capacity to store models and weaker processing power. Therefore, it is not possible to directly apply the large deep learning models used in the cloud environment. Instead, it is necessary to use light models, reduced architectures and algorithms adapted to limited resources. These limitations affect not only the size of the model, but also the complexity of the input data, the speed of inference, and the ability to process multiple sensor streams simultaneously [5, 13].

Energy Consumption and Battery Life

Another significant challenge is energy consumption. Many IoT devices run on batteries or under power-limited conditions, making energy efficiency one of the key design criteria. Local data processing can reduce the need to wirelessly send large amounts of information, which has a positive effect on energy consumption. However, running the AI model on the device also requires some processing activity and can drain the battery faster. Therefore, it is necessary to balance carefully between the frequency of sensor readings, the complexity of the model, the number of inferences and the way of communication with the gateway or cloud layer. TinyML is particularly notable because it seeks to enable ultra-low power consumption, but still requires optimization of the entire system, not just the model itself [7, 5].

Model Compression, Quantization and Optimization

The third group of challenges relates to model optimization. In order to run machine learning models on edge and TinyML devices, techniques such as model compression, quantization, pruning, parameter reduction, and selection of efficient neural architectures are often used. Quantization reduces the memory footprint of the model and speeds up inference by representing numerical values with less precision. Pruning removes less significant connections or neurons from the model, while compression reduces the overall size of the model. However, these techniques may lead to a drop in accuracy, model instability, or poorer generalization in real-world conditions. Therefore, optimization must not be seen only as a technical reduction of the model, but as a process of finding an acceptable compromise between performance, size, speed and energy consumption [2, 5, 20].

Security, Privacy and Trust

Security, privacy and trust present additional challenges in Edge AI and TinyML systems. Local data processing can improve privacy because sensitive data does not have to be constantly sent to the cloud. However, the distributed nature of edge and IoT environments increases the number of potential attack points. Edge gateway devices, sensor nodes and microcontrollers can be physically accessible to attackers, exposed to unauthorized access, data manipulation, malicious updates or attacks on the AI model itself. An additional problem is the trust in the decisions of local models, especially when they are used in healthcare, industrial automation, security systems or smart cities. Therefore, it is necessary to develop secure authentication, encryption, model protection, secure updating and explainability mechanisms that increase user confidence in locally made decisions [13, 3].

Deployment, Maintenance and Scalability

The fifth challenge is related to the implementation, maintenance and scalability of the system. Edge AI and TinyML solutions often consist of a large number of distributed devices located in different physical environments. This makes installation, monitoring, diagnostics, model updates and troubleshooting difficult. A particular problem is the remote updating of models on devices with limited memory and weak network connectivity. In addition, the lack of standardization and interoperability between different hardware platforms, communication protocols and software frameworks hinders the wider application of these technologies. Scalability is not only a matter of the number of devices, but also of their heterogeneity, secure management, model compatibility and long-term system maintenance. Therefore, the future development of Edge AI and TinyML systems requires standardized architectures, efficient MLOps approaches for edge environments, secure update mechanisms, and tools to monitor model performance over time [6, 22].

Conclusion

Edge AI and TinyML represent important technological trends in the development of modern Internet of Things systems. Their key contribution is reflected in the transformation of IoT systems from a predominantly passive infrastructure for data collection and transmission into intelligent, distributed and context-aware systems capable of local data processing and real-time decision-making. Instead of relying exclusively on centralized cloud platforms, modern IoT architectures increasingly distribute intelligence between sensors, microcontrollers, embedded devices, edge gateway nodes and cloud services.

The review shows that Edge AI enables faster and more autonomous processing of data closer to its source, while TinyML extends this capability to devices with extremely limited resources, such as microcontrollers and low-power sensor nodes. The most important advantages of these approaches include reduced latency, lower consumption of network traffic, greater privacy, higher degree of system autonomy and better energy efficiency. Therefore, Edge AI and TinyML are particularly relevant for smart homes, smart classrooms, health monitoring, wearables, industrial IoT, predictive maintenance, smart agriculture, environmental monitoring, security systems and smart cities.

However, the wider application of Edge AI and TinyML technologies is still limited by a number

of challenges. The most significant limitations are related to small memory, weaker processing power, limited battery capacity, the need to optimize the model, security risks, privacy issues, interoperability and difficulties in maintaining and updating the model on a large number of distributed devices. A particular challenge is the trade-off between model accuracy, latency, power consumption and model size. Edge AI and TinyML should not be seen as a replacement for cloud computing, but as complementary layers of intelligence in modern IoT ecosystems.

References

[1] Carvalho, G., Cabral, B., Pereira, V., & Bernardino, J. (2021). Edge computing: current trends, research challenges and future directions. Computing, 103(5), 993-1023. https://doi.org/10.1007/s00607-020-00896-5
[2] Chang, Z., Liu, S., Xiong, X., Cai, Z., & Tu, G. (2021). A survey of recent advances in edge-computing-powered artificial intelligence of things. IEEE Internet of Things Journal, 8(18), 13849-13875. https://doi.org/10.1109/JIOT.2021.3088875
[3] Gill, S. S., Golec, M., Hu, J., Xu, M., Du, J., Wu, H., ... & Uhlig, S. (2025). Edge AI: A taxonomy, systematic review and future directions. Cluster Computing, 28(1), 18. https://doi.org/10.1007/s10586-024-04686-y
[4] Zhou, Z., Chen, X., Li, E., Zeng, L., Luo, K., & Zhang, J. (2019). Edge intelligence: Paving the last mile of artificial intelligence with edge computing. Proceedings of the IEEE, 107(8), 1738-1762. https://doi.org/10.1109/JPROC.2019.2918951
[5] Ray, P. P. (2022). A review on TinyML: State-of-the-art and prospects. Journal of King Saud University-Computer and Information Sciences, 34(4), 1595-1623. https://doi.org/10.1016/j.jksuci.2021.11.019
[6] Elhanashi, A., Dini, P., Saponara, S., & Zheng, Q. (2024). Advancements in TinyML: Applications, limitations, and impact on IoT devices. Electronics, 13(17), 3562. https://doi.org/10.3390/electronics13173562
[7] Schizas, N., Karras, A., Karras, C., & Sioutas, S. (2022). TinyML for ultra-low power AI and large scale IoT deployments: A systematic review. Future Internet, 14(12), 363. https://doi.org/10.3390/fi14120363
[8] Stošić, L., Dimitrovska, M., & Pushova Stamenkova, L. (2023). Exploring the expanding world of IoT: A comprehensive overview and considerations. KNOWLEDGE - International Journal , 60(3), 359–363. Retrieved from https://ojs.ikm.mk/index.php/kij/article/view/6273
[9] Stošić, L., Dimitrovska, M., Pushova Stamenkova, L., & Smelcerović, M. (2023). From concept to reality: Understanding the internet of things. SCIENCE International Journal, 2(4), 181–184. https://doi.org/10.35120/sciencej0204181s
[10] Domínguez-Bolaño, T., Campos, O., Barral, V., Escudero, C. J., & García-Naya, J. A. (2022). An overview of IoT architectures, technologies, and existing open-source projects. Internet of Things, 20, 100626. https://doi.org/10.1016/j.iot.2022.100626
[11] Choudhary, A. (2024). Internet of Things: a comprehensive overview, architectures, applications, simulation tools, challenges and future directions. Discover Internet of Things, 4(1), 31. https://doi.org/10.1007/s43926-024-00084-3
[12] Hamdan, S., Ayyash, M., & Almajali, S. (2020). Edge-computing architectures for internet of things applications: A survey. Sensors, 20(22), 6441. https://doi.org/10.3390/s20226441
[13] Singh, R., & Gill, S. S. (2023). Edge AI: a survey. Internet of Things and Cyber-Physical Systems, 3, 71-92. https://doi.org/10.1016/j.iotcps.2023.02.004
[14] TensorFlow. (n.d.). TensorFlow Lite for Microcontrollers. GitHub. https://github.com/tensorflow/tflite-micro
[15] Olja Krčadinac, Željko Stanković, Dragana Dudić, Lazar Stošić (2024). Development of an Open-Source Voice-Controlled Smart Home System, JITA – Journal of Information Technology and Applications, 14(2), 111-116, https://doi.org/10.7251/JIT2402111K
[16] Heydari, S., & Mahmoud, Q. H. (2025). Tiny machine learning and on-device inference: A survey of applications, challenges, and future directions. Sensors, 25(10), 3191. https://doi.org/10.3390/s25103191
[17] Nguyen, D. C., & Welch, C. (2026). Generative artificial intelligence in qualitative data analysis: Analyzing—Or just chatting?. Organizational Research Methods, 29(1), 3-39. https://doi.org/10.1177/10944281251377154
[18] Artiushenko, V., Lang, S., Lerez, C., Reggelin, T., & Hackert-Oschätzchen, M. (2024). Resource-efficient Edge AI solution for predictive maintenance. Procedia Computer Science, 232, 348-357. https://doi.org/10.1016/j.procs.2024.01.034
[19] Gookyi, D. A. N., Wulnye, F. A., Wilson, M., Danquah, P., Danso, S. A., & Gariba, A. A. (2024). Enabling intelligence on the edge: leveraging edge impulse to deploy multiple deep learning models on edge devices for tomato leaf disease detection. AgriEngineering, 6(4), 3563-3585. https://doi.org/10.3390/agriengineering6040203
[20] Bhushan, B., Negi, P., Nayak, A., & Goyal, S. (2025). Graphene composites for water remediation: an overview of their advanced performance with focus on challenges and future prospects. Advanced Composites and Hybrid Materials, 8(1), 55. https://doi.org/10.1007/s42114-024-01088-x
[21] R. Shelke, K., & K. Shinde, S. (2025). SAOA:: Skill archimedes optimization algorithm based privacy enhancement for blockchain storage optimization in medical IoT environment. https://doi.org/10.1016/j.compeleceng.2025.110270
[22] Wang, T., Guo, J., Zhang, B., Yang, G., & Li, D. (2025). Deploying AI on edge: Advancement and challenges in edge intelligence. Mathematics, 13(11), 1878. https://doi.org/10.3390/math13111878