Review Article

Aibi research, management and engineering journal. Volume 13, Number 3, Pag 1-10, September - December 2025, ISSN 2346-030X

Embedded low-power machine-learning on microcontrollers: systematic literature review

Sleiter Ramos-Sanchez[1], Jinmi Lezama-Calvo[2], Ronald Paucar-Curasma[3]
[1,2]*Universidad Nacional Tecnológica de Lima Sur, Lima - Perú*
[3]*Universidad Nacional Autónoma de Tayacaja Daniel Hernández Morillo, Huancavelica - Perú*

Received: february 19, 2025.
Accepted: july 17, 2025.
Publicado: september 01, 2025.

***Abstract*—** The increasing adoption of artificial intelligence (AI) and machine learning (ML) in engineering systems has traditionally relied on cloud-based processing and high-performance platforms, generating significant computational, energy, and infrastructure costs. The emergence of embedded low-power machine learning, known as TinyML, has enabled the deployment of ML models directly on resource-constrained microcontrollers (MCUs), achieving near-sensor intelligence, reduced latency, and improved energy efficiency. This article presents a systematic literature review on the use of low-power microcontrollers for implementing machine learning techniques. The objective is to synthesize and analyze existing research on embedded ML in MCUs, emphasizing hardware platforms, application domains, and models, methods, and algorithms adopted under strict resource constraints. The review follows the standard systematic review methodology, using the Scopus database for publications from 2020 to 2025. The analyzed studies reveal that ARM Cortex-M–based microcontrollers and Arduino-class devices are the predominant platforms. Applications span environmental monitoring, energy systems, agriculture, healthcare, industrial safety, biosensing, and smart infrastructure. The most commonly employed models include lightweight convolutional neural networks, compact dense networks, and classical ML algorithms, optimized through quantization, pruning, and compression. This review highlights the growing maturity of machine learning on low-power microcontrollers and identifies key trends, constraints, and design trade-offs shaping current embedded ML implementations.

Keywords: literature-review, low-power, machine-learning, microcontrollers, tinyml.

*Corresponding author: 2419010015@untels.edu.pe (Sleiter Ramos Sanchez).
Peer reviewing is a responsability of the Universidad de Santander.
How to cite this article: S. Ramos-Sanchez, J. Lezama-Calvo and R. Paucar-Curasma, "Embedded low-power machine-learning on microcontrollers: systematic literature review", *Aibi research, management and engineering journal*, vol. 13, no. 3, pp. 1-10 2025, doi: 10.15649/2346030X.4183

I. INTRODUCTION

Artificial intelligence (AI) and machine learning (ML) have advanced significantly in various industries [1], [2]. AI refers to computational systems that imitate and simulate human intelligence, such as the ability to solve problems and learn. On the other hand, ML is a branch of AI that automatically extracts patterns from data [3]. However, the training of Deep Learning (DL) [4] models involve a high computational cost, with high consumption of CPU [5], GPU [6], memory, and time [7], [8]. In addition, the large volume of data to be processed and sent to the cloud [9] implies considerable energy, bandwidth, and storage expenses [10], [11].

A proposed alternative to address these limitations is the processing and analysis of data on edge devices or edge computing [12], [13]. This strategy allows scaling and avoiding the saturation of the cloud, by extracting only the relevant information and metadata to be sent to the cloud. Additionally, edge devices could temporarily mitigate cloud availability problems. Nevertheless, the hardware used as an edge device has restrictions in terms of memory resources, energy consumption, and compatibility [14]. Likewise, the architectures designed for embedded systems depend on the type of hardware and software used [15].

In this context, the field of engineering has been driven by the growing integration of ML techniques [16] in embedded and low-power devices [1], [17], [18]. These intelligent systems have enabled the development of innovative solutions in various areas, from the Internet of Things (IoT) [19] to robotics and industrial automation [3], [20].

The present document aims to conduct a systematic review of the literature on the use of low-power embedded devices that have been implemented with ML techniques, exploring the applications and the models, methods, or algorithms used in this context. For this purpose, the Scopus scientific database [21] was consulted, which indexes a large number of journals in all disciplines, with the area of interest being engineering.

II. METHODOLOGY

In the development of this systematic review research work, which is a "synthesis of the available evidence" [22], the objective is to summarize the existing information regarding the topic of Embedded low power ML on microcontrollers (MCU). The existing methodologies for the systematic review of the literature, such as those used in the work by Manterola [22], Evans [23] y Grant [24], provide a set of phases to be used: search and selection, evaluation, information extraction, and review synthesis. These phases are developed based on formulated questions and inclusion criteria for the selected information sources [22], [23], [24], described in the present work, with the stages shown graphically in Figure 1.

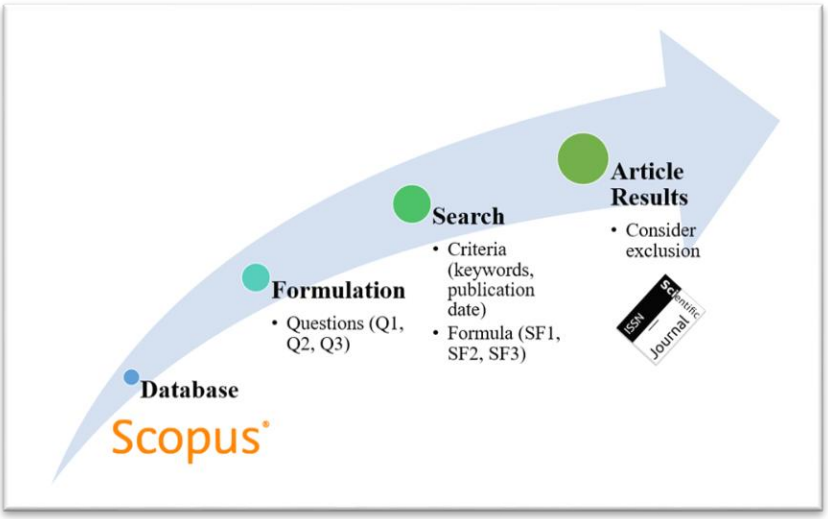

Figura 1: Literature review phases in the selected database.
Fuente: Elaboración propia.

a. Problem Statement

Edge computing has become a prominent research field, as it facilitates the implementation of ML techniques in multiple use cases. However, the hardware deployed at the network edge has severely limited resources, such as memory capacity, energy consumption, and compatibility, which restricts the provision of complex high-level services (Figure 2). In fact, the current edge computing scenario does not exactly reflect the expected paradigm of cloud-to-embedded device [25], [26].

For this reason, a new concept called TinyML has emerged, which is the convergence of machine learning and resource-constrained embedded devices [27], [28]. TinyML enables the deployment of deep learning models on Edge devices [26] that have severe resource constraints, such as clock speed, memory, and energy consumption [27], [29]. This technology allows for local data analysis and interpretation on the devices, providing greater responsiveness and privacy, while avoiding the high energy costs typically associated with wireless communication [14], [30].

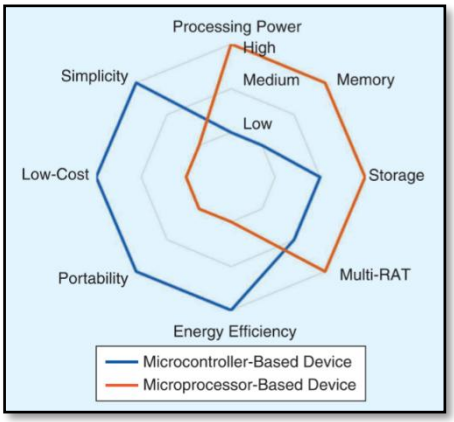

Figure 2: Comparison of characteristics between MCU and microprocessors.
Fuente: Recuperado de [25].

Therefore, the research questions formulated in this literature review-based investigation are as follows:

- Q1: What low-power embedded devices are used in the field of engineering for the implementation of machine learning techniques?
- Q2: What applications use machine learning techniques on low-power microcontrollers?
- Q3: What machine learning models, methods, or algorithms are employed on low-power microcontrollers?

b. Search Strategies

The literature review focuses on publications from 2020 to 2025 due to both technological relevance and methodological consistency with the current state of embedded machine learning. While early attempts to deploy machine learning on embedded systems existed prior to this period, most of these works relied on ad hoc implementations, proprietary toolchains, or simplified classifiers that do not reflect the modern TinyML ecosystem. A key turning point for the field was the emergence and consolidation of TinyML frameworks and toolchains, particularly the public release and widespread adoption of TensorFlow Lite for Microcontrollers (TFLM) around 2019. From 2020 onwards, the availability of standardized, open-source inference engines, along with improved compiler support, quantization workflows, and vendor-backed SDKs, enabled systematic and reproducible deployment of machine learning models on ultra-low-power microcontrollers. Consequently, research published after 2020 reflects a qualitative shift from proof-of-concept demonstrations to deployable, optimized, and application-oriented embedded ML systems.

Moreover, low-power microcontroller hardware has evolved significantly in recent years, with widespread availability of ARM Cortex-M4/M7 devices, integrated DSP instructions, and optional neural accelerators, which directly influences the feasibility and performance of on-device inference. Restricting the review to the last five years ensures that the analyzed works are aligned with current hardware capabilities, software frameworks, and deployment constraints relevant to contemporary engineering practice. Nevertheless, the temporal window is not intended to exclude influential earlier contributions. Seminal works published prior to 2020 are considered when they introduce foundational concepts, algorithms, or architectural principles that remain relevant to modern embedded machine learning. Such works are referenced selectively when they provide necessary historical context or methodological grounding, even if they fall outside the primary time window.

To situate this review within the broader research landscape and ensure methodological rigor, it is important to acknowledge the existence of prior surveys related to embedded machine learning on resource-constrained devices. For example, the work titled “TinyML: Enabling of Inference Deep Learning Models on Ultra-Low-Power IoT Edge Devices for AI Applications” [27] provides a comprehensive overview of inference deployment of deep learning models on ultra-low-power edge hardware, drawing on sources indexed in Google Scholar and Web of Science.

While this and other review articles offer valuable insights into the general capabilities and frameworks for TinyML, the present study is differentiated in several respects: (i) it focuses specifically on low-power microcontrollers (MCUs) in engineering applications, (ii) it systematically characterizes not only inference frameworks but also the actual devices, applications, and machine learning models used in peer-reviewed research, and (iii) it integrates qualitative synthesis organized around research questions tailored to the constraints and opportunities of embedded machine learning in real-world engineering systems. Regarding the use of bibliographic sources, this review was conducted primarily using the Scopus database, selected for its broad and consistent coverage of peer-reviewed journals and conference proceedings across engineering, computer science, and related areas.

Scopus provides structured indexing that facilitates systematic screening and reproducibility of search results, which are essential criteria for rigorous literature reviews. Nonetheless, it is acknowledged that no single database provides exhaustive coverage of all relevant publications. Scopus has limitations in terms of the range of journals and conferences covered, particularly for emerging topics where publication venues may be distributed across multiple domains. While this review is anchored in Scopus for systematic and reproducible retrieval, recognition is given to the broader literature through cross-database validation and citation analysis. Regarding the use of bibliographic sources, this review was conducted primarily using the Scopus database, selected for its broad and consistent coverage of peer-reviewed journals and conference proceedings in engineering, computer science, and related areas. Scopus provides structured indexing that facilitates systematic selection and reproducibility of search results, which are essential criteria for conducting rigorous literature reviews [21]. With the identified database, the following search and inclusion criteria are established, as shown in the following Table 1.

Table 1: Search criteria.

Parameters	Values
Keywords	embedded machine learning low power machine learning mcu tinyml
Language	English
Type	Article
Publication date	From 2020 to 2024
Area	Engineering

Source: Own elaboration.

Table 2 shows the three search formulas that are related to the keywords described in Table 1. The search was conducted in English, the type of material is scientific articles within the field of engineering, with a publication range from 2020 to 2024.

Table 2: Search formulas.

Formula	Value
SF1	TITLE-ABS-KEY ("embedded machine learning" AND "low power") AND PUBYEAR > 2019 AND PUBYEAR < 2025 AND (LIMIT-TO (DOCTYPE , "ar")) AND (LIMIT-TO (LANGUAGE , "English")) AND (LIMIT-TO (SUBJAREA , "ENGI"))
SF2	TITLE-ABS-KEY ("machine learning" AND "mcu" AND "low power") AND PUBYEAR > 2019 AND PUBYEAR < 2025 AND (LIMIT-TO (DOCTYPE , "ar")) AND (LIMIT-TO (LANGUAGE , "English")) AND (LIMIT-TO (SUBJAREA , "ENGI"))
SF3	TITLE-ABS-KEY ("tinyml" AND "mcu" AND "low power") AND PUBYEAR > 2021 AND PUBYEAR < 2025 AND (LIMIT-TO (DOCTYPE , "ar")) AND (LIMIT-TO (LANGUAGE , "English")) AND (LIMIT-TO (SUBJAREA , "ENGI"))

Source: Own elaboration.

c. Execution of the Review

The result of the search criteria and the application of the search formula is shown in Table 3. The search was conducted on June 1, 2024.

Table 3: Search results.

Parameters	SF1	SF2	SF3
2024	2	3	1
2023	4	3	3
2022	5	3	1
2021	3	2	0
2020	0	0	0
Total	**14**	**11**	**5**

Source: Own elaboration.

As shown in the table on the left in Table 3, by applying the search formula SF1, a total of 14 articles are obtained, by applying SF2, 11 articles are obtained, and by applying SF3, 5 articles are obtained.

It is necessary to consider excluding some articles to discard those that are not useful for the present research. Regarding the SF1 formula ("embedded machine learning" AND "low power"), 4 articles have been excluded that are not related to the topic, as they are tools, circuits, or frameworks, on which it is proposed to work with artificial intelligence. These are not of interest to us, as we are looking for techniques already implemented in embedded devices.

Regarding SF2 ("machine learning" AND "mcu" AND "low power"), 4 articles have been excluded as they were found in the results of the next query, i.e., duplicates. Additionally, 1 article has been excluded for proposing a framework that uses machine learning. Regarding SF3 ("tinyml" AND "mcu" AND "low power"), no articles have been excluded, as all the articles found are related to the central topic of interest.

Table 4: Search results after applying the exclusions.

Parameters	SF1	SF2	SF3
2024	2	2	1
2023	3	0	3
2022	3	2	1
2021	2	1	0
2020	0	0	0
Total	**10**	**5**	**5**

Source: Own elaboration.

Looking at Table 4, after applying the exclusions of some articles that were part of the results of the SF1 formula, a total of 10 articles are obtained, after the exclusions on SF2, 6 articles are obtained, and for the case of SF3, the 5 articles initially found are maintained.

The quantities of the results are then shown graphically in Figure 3. The first image on the left corresponds to the results obtained by the different search formulas SF1, SF2, and SF3. In the graph on the right, we show the number of results obtained after having excluded some articles for the reasons mentioned above.

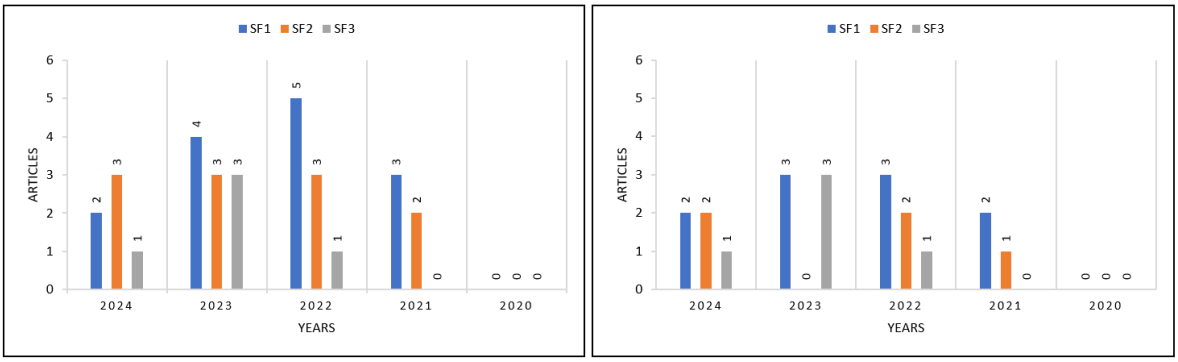

Figure 3: Search results graphically grouped by year.
Source: Own elaboration.

III. RESULTS, ANALYSIS AND INTERPRETATIOM

The selected articles according to the search criteria SF1, SF2, and SF3 employed are presented, and we will review the research advances according to the previously raised topic: artificial intelligence in low-power microcontrollers. The following shows the details of the selected articles, according to the keyword search filters within the selection criteria identified earlier, carried out in the Scopus database.

Tables V, VI, and VII show the tables with the content of the selected articles, displaying in the columns data such as the title, authors, and year of publication, as well as an identification code that represents each article. This identification code is formed by:

Code = "ART-" + "SF(Search Formula)" + "_(Article Number)"

The articles are grouped as follows: 10 articles for the SF1 search (ART-SF1_01, ART-SF1_02, ART-SF1_03, ART-SF1_04, ART-SF1_05, ART-SF1_06, ART-SF1_07, ART-SF1_08, ART-SF1_09, and ART-SF1_10), 5 articles for the SF2 search (ART-SF2_01, ART-SF2_02, ART-SF2_03, ART-SF2_04 and ART-SF2_05), and finally, 5 articles for the SF3 search (ART-SF3_01, ART-SF3_02, ART-SF3_03, ART-SF3_04, and ART-SF3_05).

Table 5: Results according to SF1 search formula.

Code	Authors	Title	Year
ART-SF1_01	Ksira Z.; Mellit A.; Blasuttigh N.; Massi Pavan A.	A Novel Embedded System for Real-Time Fault Diagnosis of Photovoltaic Module [31] s	2024
ART-SF1_02	Martinez-Rau L.S.; Chelotti J.O.; Giovanini L.L.; Adin V.; Oelmann B.; Bader S.	On-Device Feeding Behavior Analysis of Grazing Cattle [32]	2024
ART-SF1_03	de Oliveira Filho J.I.; Faleiros M.C.; Ferreira D.C.; Mani V.; Salama K.N.	Empowering Electrochemical Biosensors with AI: Overcoming Interference for Precise Dopamine Detection in Complex Samples [33]	2023
ART-SF1_04	Cappelli I.; Carli F.; Fort A.; Intravaia M.; Micheletti F.; Peruzzi G.; Vignoli V.	Enhanced Visible Light Localization Based on Machine Learning and Optimized Fingerprinting in Wireless Sensor Networks [34]	2023
ART-SF1_05	Peruzzi G.; Pozzebon A.; Van Der Meer M.	Fight Fire with Fire: Detecting Forest Fires with Embedded Machine Learning Models Dealing with Audio and Images on Low Power IoT Devices [35]	2023
ART-SF1_06	Mongardi A.; Rossi F.; Prestia A.; Ros P.M.; Roch M.R.; Martina M.; Demarchi D.	Hand Gestures Recognition for Human-Machine Interfaces: A Low-Power Bio-Inspired Armband [36]	2022
ART-SF1_07	Roy A.; Dutta H.; Griffith H.; Biswas S.	An On-Device Learning System for Estimating Liquid Consumption from Consumer-Grade Water Bottles and Its Evaluation [37]	2022
ART-SF1_08	Krayden A.; Schohet M.; Shmueli O.; Shlenkevitch D.; Blank T.; Stolyarova S.; Nemirovsky Y.	CMOS-MEMS Gas Sensor Dubbed GMOS for SelectiveAnalysis of Gases with Tiny Edge Machine Learning [38]	2022
ART-SF1_09	Dellragnola F.; Pale U.; Marino R.; Arza A.; Atienza D.	MBioTracker: Multimodal Self-Aware Bio-Monitoring Wearable System for Online Workload Detection [39]	2021
ART-SF1_10	Shabani F.; Philamore H.; Matsuno F.	An energy-autonomous chemical oxygen demand sensor using a microbial fuel cell and embedded machine learning [40]	2021

Source: Own elaboration.

Table 6: Results according to SF2 search formula.

Code	Authors	Title	Year
ART-SF2_01	Mohan N.; Abdelrahman D.; Ali N.F.; Atef M.	An Integrated High-Gain Wide-Dynamic Range Photoplethysmography Sensor for Cardiac Health Monitoring [41]	2024
ART-SF2_02	Zishan M.A.O.; Shihab H.M.; Islam S.S.; Riya M.A.; Rahman G.M.; Noor J.	Dense neural network based arrhythmia classification on low-cost and low-compute micro-controller [42]	2024
ART-SF2_03	Tabanelli E.; Brunelli D.; Acquaviva A.; Benini L.	Trimming Feature Extraction and Inference for MCU-Based Edge NILM: A Systematic Approach [43]	2022
ART-SF2_04	Xiao J.; Liu J.; Yang H.; Liu Q.; Wang N.; Zhu Z.; Chen Y.; Long Y.; Chang L.; Zhou L.; Zhou J.	ULECGNet: An Ultra-Lightweight End-to-End ECG Classification Neural Network [44]	2022
ART-SF2_05	Wang X.; Cavigelli L.; Schneider T.; Benini L.	Sub-100 µW Multispectral Riemannian Classification for EEG-Based Brain–Machine Interfaces [45]	2021

Source: Own elaboration.

Table 7: Results according to SF3 search formula.

Code	Authors	Title	Year
ART-SF3_01	Zanghieri M.; Indirli F.; Latella A.; Puglia G.M.; Tecce F.; Papariello F.; Urlini G.; Benini L.; Conti F.	An Extreme-Edge TCN-Based Low-Latency Collision-Avoidance Safety System for Industrial Machinery [46]	2024
ART-SF3_02	Xu K.; Zhang H.; Li Y.; Zhang Y.; Lai R.; Liu Y.	An Ultra-Low Power TinyML System for Real-Time Visual Processing at Edge [47]	2023
ART-SF3_03	Albanese A.; Nardello M.; Fiacco G.; Brunelli D.	Tiny Machine Learning for High Accuracy Product Quality Inspection [48]	2023
ART-SF3_04	Zhang Y.; Adin V.; Bader S.; Oelmann B.	Leveraging Acoustic Emission and Machine Learning for Concrete Materials Damage Classification on Embedded Devices [49]	2023
ART-SF3_05	Manor E.; Greenberg S.	Custom Hardware Inference Accelerator for TensorFlow Lite for Microcontrollers [50]	2022

Source: Own elaboration.

The following is a summary of the answers to the research questions:

a. Q1: What low-power hardware or devices exist in the field of engineering for the implementation of machine learning techniques?

The literature shows a clear trend toward the use of low-power embedded devices, mainly microcontroller units (MCUs) and MCU-centered heterogeneous platforms, as the primary hardware for implementing machine learning techniques in engineering applications. These devices enable on-device inference under strict constraints of energy, memory, and computational resources, which is essential for real-time, autonomous, and edge-based systems. Several works demonstrate the feasibility of deploying machine learning models on commercial low-power MCUs. For instance, ART-SF1_01 reports the implementation of a TinyCNN on an Arduino Nano 33 BLE Sense, enabling real-time photovoltaic fault diagnosis directly on the device without cloud dependence. Similarly, ART-SF1_02 employs a Raspberry Pi Pico (ARM Cortex-M0+) to analyze grazing cattle feeding behavior using embedded ML, achieving sub-21 mW power consumption. These studies highlight that ARM Cortex-M–based platforms are widely adopted due to their favorable balance between performance and energy efficiency. Beyond environmental and agricultural monitoring, embedded ML on low-power devices is increasingly used in biosensing and healthcare engineering. In ART-SF1_03, TinyML models are integrated into low-power portable embedded systems for electrochemical biosensors, improving selectivity and noise rejection. This trend continues in ART-SF1_06, ART-SF1_09, and ART-SF2_04, where wearable systems rely on low-power embedded processors—such as MSP432 and similar MCUs—to perform real-time gesture recognition, cognitive workload monitoring, and ECG classification with millijoule-level energy consumption per inference. These results demonstrate that MCUs are sufficiently capable of executing lightweight neural networks and classical ML algorithms in medical and human–machine interface applications.

Another relevant group of studies focuses on energy-autonomous and ultra-low-power systems, where the choice of embedded hardware is critical. In ART-SF1_10, a low-power microcontroller powered solely by a microbial fuel cell runs a support vector regression algorithm for water quality monitoring, illustrating that embedded ML can operate under extreme energy constraints. Likewise, ART-SF2_05 shows that EEG-based brain–machine interfaces can be implemented on low-power MCUs with power consumption below 100 µW, reinforcing the suitability of these devices for long-term, near-sensor intelligence. Low-power embedded devices are also widely used in industrial and infrastructure engineering. In ART-SF1_04 and ART-SF3_01, low-cost MCUs execute neural network regressors and temporal convolutional networks for indoor localization and industrial collision avoidance, respectively, achieving low latency and high robustness. Structural health monitoring applications further confirm this trend: ART-SF3_04 deploys a lightweight CNN on an nRF52840 (ARM Cortex-M4) microcontroller to classify concrete damage with high accuracy and minimal energy usage. These works demonstrate that MCUs can meet the reliability and real-time requirements of industrial environments while maintaining low power consumption. In parallel, several studies highlight the integration of low-power IoT devices and sensor-centric platforms with embedded ML. Forest fire detection systems (ART-SF1_05) and gas sensing solutions (ART-SF1_08) leverage embedded processors tightly coupled with sensors to perform local inference and transmit only high-level results, significantly reducing communication energy costs. This sensor–MCU coupling is a key architectural choice in low-power engineering systems.

Finally, recent research points toward heterogeneous embedded architectures as an evolution of low-power ML platforms. While MCUs remain the core processing units, some works incorporate dedicated neural accelerators or co-processors to improve efficiency. For example, ART-SF3_02 and ART-SF3_05 combine MCUs with neural processing units (NPUs) or custom hardware accelerators to support more demanding ML workloads while preserving low power operation. This indicates a growing interest in hybrid solutions where low-power embedded devices are augmented with specialized hardware to extend their machine learning capabilities.

Overall, the reviewed literature clearly indicates that low-power microcontrollers (e.g., Arduino-class MCUs, ARM Cortex-M series, MSP432, nRF52, STM32) are the dominant embedded devices used in engineering for implementing machine learning techniques. These devices, sometimes enhanced with energy harvesting, sensor-integrated designs, or lightweight hardware accelerators, enable efficient, real-time, and autonomous machine learning at the edge, directly addressing the constraints and requirements of modern engineering applications.

b. Q2: What applications use machine learning techniques in low-power microcontrollers?

The reviewed literature demonstrates that machine learning techniques deployed on low-power microcontrollers are used across a wide range of engineering applications, where local intelligence, real-time response, and energy efficiency are critical requirements. These applications span environmental monitoring, healthcare and biomedical systems, industrial automation, agriculture, smart infrastructure, and wearable technologies, highlighting the versatility of embedded machine learning at the edge. A major application domain is environmental monitoring and sustainability. In ART-SF1_01, machine learning is applied to the real-time fault diagnosis of photovoltaic modules using infrared imagery, enabling predictive maintenance and improved energy production. Similarly, ART-SF1_10 applies embedded machine learning to water quality monitoring, where a low-power microcontroller estimates chemical oxygen demand using data from a microbial fuel

cell. Forest fire detection represents another key environmental application, as shown in ART-SF1_05, where embedded ML models process audio and image data locally to enable early fire warning while minimizing energy consumption and communication overhead.

Agriculture and livestock monitoring also benefit significantly from machine learning on low-power microcontrollers. In ART-SF1_02, embedded ML techniques are used to analyze grazing cattle feeding behavior through acoustic signals, supporting precision livestock farming and autonomous field deployment. These applications rely on continuous monitoring in outdoor environments, where low power consumption and robustness are essential.

Another prominent application area is biomedical and healthcare engineering, where embedded ML enables real-time physiological signal analysis directly on wearable or portable devices. Hand gesture recognition for human–machine interfaces is addressed in ART-SF1_06, where machine learning models run on a low-power armband using sEMG signals. Cognitive workload detection in ART-SF1_09 further illustrates how embedded ML supports adaptive systems in high-risk operations. Cardiac health monitoring and diagnosis are recurrent applications: ART-SF2_01, ART-SF2_02, and ART-SF2_04 employ machine learning on low-power microcontrollers for blood pressure estimation, arrhythmia classification, and ECG signal classification, respectively. Additionally, ART-SF2_05 demonstrates the use of embedded ML in EEG-based brain–machine interfaces, enabling near-sensor classification with ultra-low power consumption.

Biosensing and chemical analysis constitute another relevant application domain. In ART-SF1_03, machine learning techniques are embedded into portable electrochemical biosensors to improve selectivity and interference rejection in complex biological samples. Similarly, ART-SF1_08 applies TinyML to CMOS-MEMS gas sensors, enabling selective gas identification and concentration estimation directly at the sensor node.

In the context of industrial systems and smart manufacturing, machine learning on low-power microcontrollers is used for safety, inspection, and monitoring tasks. ART-SF3_01 presents a collision-avoidance safety system for industrial machinery, where a temporal convolutional network running on an MCU ensures low-latency human detection. Quality inspection of manufactured components is addressed in ART-SF3_03, where tinyML camera systems perform visual defect and anomaly detection locally on MCU-based platforms. Structural health monitoring is another industrial application, as shown in ART-SF3_04, where embedded ML classifies damage in concrete materials using acoustic emission signals.

Low-power microcontrollers also support smart sensing and infrastructure applications. Indoor localization based on visible light is explored in ART-SF1_04, where embedded machine learning regressors estimate position within wireless sensor networks. Energy monitoring and smart metering applications are addressed in ART-SF2_03, where embedded ML enables nonintrusive load monitoring on MCU-based smart meters, supporting energy efficiency and demand analysis.

Finally, several works illustrate that computer vision and advanced perception tasks can also be supported at the extreme edge. ART-SF3_02 and ART-SF3_05 show that, when combined with lightweight models or hardware accelerators, low-power microcontrollers can be used for real-time visual processing and general ML inference acceleration, expanding the range of feasible applications beyond traditional sensing.

Overall, the literature indicates that machine learning techniques on low-power microcontrollers are applied to environmental and energy monitoring, agriculture, biomedical and wearable systems, biosensing, industrial safety and inspection, smart infrastructure, and edge perception tasks. These applications exploit the ability of low-power MCUs to perform local inference efficiently, enabling autonomous, real-time, and scalable engineering solutions without reliance on cloud-based processing.

c. Q3: What machine learning models, methods, or algorithms are used in low-power microcontrollers?

The reviewed literature indicates that machine learning models deployed on low-power microcontrollers are carefully selected and designed to balance accuracy, memory footprint, computational complexity, and energy consumption. As a result, the dominant approaches include lightweight neural networks, classical machine learning algorithms, and optimized or compressed deep learning models, often combined with quantization, pruning, and feature reduction techniques.

A large portion of the studies rely on lightweight neural networks, particularly convolutional neural networks (CNNs) tailored for embedded execution. In ART-SF1_01, a TinyCNN is used for photovoltaic fault classification from infrared images, demonstrating that compact CNN architectures can be executed directly on Arduino-class microcontrollers. Similar CNN-based approaches are reported in ART-SF3_03 and ART-SF3_04, where compressed versions of MobileNetV2, SqueezeNet, and custom lightweight CNNs are deployed on MCUs for industrial quality inspection and structural health monitoring. These works show that CNNs with a limited number of parameters (on the order of tens of thousands) are practical for real-time inference on low-power devices. Beyond spatial feature extraction, temporal and sequential models are also employed. In ART-SF3_01, a Temporal Convolutional Network (TCN) is implemented on an MCU for collision-avoidance systems, enabling low-latency processing of time-series data from ultrasonic sensors. This highlights the feasibility of temporal deep learning models when carefully optimized for embedded platforms.

In addition to deep learning, several studies adopt classical machine learning algorithms, which are often preferred for their low computational cost and interpretability. Support Vector Regression (SVR) is used in ART-SF1_10 to estimate chemical oxygen demand in an energy-autonomous water quality sensor. Support Vector Machines (SVMs), K-nearest neighbors (KNN), and related classifiers are discussed or compared in applications such as nonintrusive load monitoring (ART-SF2_03) and EEG-based brain–machine interfaces (ART-SF2_05), where feature engineering enables efficient execution on MCUs. Dense (fully connected) neural networks with a small number of layers and neurons are another common choice. In ART-SF2_02, a compact dense neural network with two hidden layers is implemented on an ATmega328 microcontroller for arrhythmia classification, achieving high accuracy with a model size of only 1.267 kB. Similarly, ART-SF1_07 employs lightweight on-device neural networks for gesture-based liquid consumption estimation, emphasizing accuracy–energy trade-offs.

Several applications rely on feature-based machine learning pipelines, where signal processing precedes classification or regression. In ART-SF1_06, hand gesture recognition is achieved using the Average Threshold Crossing (ATC) feature combined with a lightweight classifier, reducing computational complexity. Feature extraction followed by ML inference is also central in ART-SF1_02, ART-SF1_09, and ART-SF2_03, where acoustic, physiological, or electrical signals are processed to extract compact feature vectors before classification. Optimization techniques play a crucial role across almost all studies. Quantization (e.g., 8-bit fixed-point inference) is explicitly reported in ART-SF1_03 and ART-SF2_05, enabling significant memory and energy savings with minimal accuracy loss. Pruning, model compression, and feature space reduction are employed in ART-SF2_03, ART-SF3_03, and ART-SF3_05 to fit models within the tight constraints of low-power microcontrollers. Some works further exploit hardware–software co-design, as in ART-SF3_02 and ART-SF3_05, where custom neural accelerators or co-processors are used to speed up inference while keeping power consumption low.

Overall, the literature shows that machine learning on low-power microcontrollers predominantly relies on TinyCNNs, lightweight dense neural networks, temporal convolutional networks, support vector machines and regressors, KNN-based classifiers, and feature-driven ML pipelines, all heavily optimized through quantization, pruning, and model compression. These models and methods enable efficient, real-time inference on resource-constrained embedded devices, making them suitable for a wide range of engineering applications at the extreme edge.

d. Training Constraints and Deployment Limitations in Low-Power Microcontrollers

Although low-power microcontrollers are increasingly capable of executing machine learning inference, training machine learning models directly on these devices remains highly constrained and is generally impractical for most applications. The reviewed literature consistently adopts an off-device training and on-device inference paradigm, where models are trained on high-performance computing platforms and later deployed to microcontrollers in optimized form. The primary limitations affecting on-device training are related to memory, computational capacity, and energy availability. Typical low-power MCUs offer only a few tens or hundreds of kilobytes of RAM and operate at clock frequencies in the order of tens of megahertz, which is insufficient for gradient-based training of neural networks. Backpropagation requires storing intermediate activations, gradients, and weight updates, leading to memory footprints that exceed the available resources of most microcontrollers used in engineering applications. Energy constraints further restrict on-device training. Several studies in the reviewed literature highlight energy budgets in the range of microjoules to millijoules per inference, such as in ART-SF1_10, ART-SF2_05, and ART-SF3_04. Training workloads would require orders of magnitude more energy than inference, making them incompatible with battery-powered or energy-harvesting systems, especially in long-term or autonomous deployments. From a technical implementation perspective, most reported systems rely on static, pre-trained models, which are then adapted to embedded execution through quantization, pruning, and model compression. For example, ART-SF1_03 and ART-SF2_05 demonstrate that 8-bit quantization enables efficient inference with minimal accuracy degradation, while ART-SF2_03 and ART-SF3_03 emphasize feature reduction and architectural simplification to meet memory constraints. These optimization steps are typically performed during the deployment phase, not during training on the microcontroller. Only limited forms of on-device adaptation are occasionally feasible, such as threshold tuning, incremental updates of lightweight classifiers, or rule-based adjustments, as suggested in energy-aware and self-aware systems like ART-SF1_09. However, full model retraining or deep neural network optimization on-device is not reported in the reviewed works, reinforcing the current separation between training and inference in low-power embedded ML.

Future research directions include federated learning, incremental learning, and ultra-lightweight on-device adaptation techniques; however, these approaches are still in early stages and fall outside the scope of this review.

IV. CONCLUSIONS

This paper presented a comprehensive synthesis of recent research on the use of machine learning techniques in low-power microcontrollers, focusing on embedded devices, application domains, and algorithmic approaches. By reviewing works published between 2020 and 2025, the study captured the period in which embedded machine learning matured into a practical and deployable engineering paradigm, driven by the emergence of standardized TinyML frameworks and improved low-power hardware platforms. The analysis shows that low-power microcontrollers, particularly those based on the ARM Cortex-M family, Arduino-class devices, and similar ultra-low-power MCUs, constitute the dominant hardware platforms for embedded machine learning. These devices enable on-device inference under strict energy, memory, and computational constraints, supporting autonomous and real-time operation in a wide range of engineering systems. From an application perspective, embedded machine learning on low-power microcontrollers is widely used in environmental monitoring, energy systems, agriculture, healthcare and wearable devices, biosensing, industrial safety, structural health monitoring, and smart infrastructure. Across these domains, local inference reduces communication overhead, improves latency, and enhances system robustness, making low-power embedded intelligence particularly suitable for edge and near-sensor deployments. Regarding algorithmic choices, the literature reveals a strong preference for lightweight neural networks, such as TinyCNNs, compact dense networks, and temporal convolutional networks, as well as classical machine learning methods including support vector machines and regression models. These approaches are systematically optimized through quantization, pruning, feature reduction, and model compression, enabling their execution within the limited resources of microcontrollers. The review also highlights important practical limitations, particularly concerning model training. Due to severe constraints in memory, computation, and energy, training is almost exclusively performed off-device, while microcontrollers are used for optimized inference. This separation between training and deployment remains a defining characteristic of low-power embedded machine learning systems. Overall, the findings confirm that low-power microcontrollers are not merely peripheral components but key enablers of modern machine learning at the edge. Future research is expected to further explore adaptive learning, federated approaches, and hardware–software co-design strategies; however, current engineering practice remains centered on efficient inference-driven solutions tailored to the constraints of resource-limited embedded platforms.

V. ACKNOWLEDGMENTS

The authors would like to thank the CONCYTEC-PROCIENCIA. This work was part of the project “E041-2023-02 Proyectos de Investigación Aplicada” [PE501083603-2023].

VI. REFERENCES

[1] R. Kallimani, K. Pai, P. Raghuwanshi, S. Iyer, and O. L. A. López, "TinyML: Tools, Applications, Challenges, and Future Research Directions," Multimed. Tools Appl., Mar. 2023, doi: 10.1007/s11042-023-16740-9.

[2] R. I. Mukhamediev et al., "Review of Artificial Intelligence and Machine Learning Technologies: Classification, Restrictions, Opportunities and Challenges," Mathematics 2022, Vol. 10, Page 2552, vol. 10, no. 15, p. 2552, Jul. 2022, doi: 10.3390/MATH10152552.

[3] M. Soori, B. Arezoo, and R. Dastres, "Artificial intelligence, machine learning and deep learning in advanced robotics, a review," Jan. 01, 2023, KeAi Communications Co. doi: 10.1016/j.cogr.2023.04.001.

[4] M. Ali, A. Dewan, A. K. Sahu, and M. M. Taye, "Understanding of Machine Learning with Deep Learning: Architectures, Workflow, Applications and Future Directions," Computers 2023, Vol. 12, Page 91, vol. 12, no. 5, p. 91, Apr. 2023, doi: 10.3390/COMPUTERS12050091.

[5] P. Zhou, "Study on CPU and FPGA in Artificial Intelligence and Machine Learning," Proceedings - 2024 13th International Conference of Information and Communication Technology, ICTech 2024, pp. 462–466, 2024, doi: 10.1109/ICTECH63197.2024.00090.

[6] T. Baji, "Evolution of the GPU Device widely used in AI and Massive Parallel Processing," 2018 IEEE Electron Devices Technology and Manufacturing Conference, EDTM 2018 - Proceedings, pp. 7–9, Jul. 2018, doi: 10.1109/EDTM.2018.8421507.

[7] L. Alzubaidi et al., "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," Journal of Big Data 2021 8:1, vol. 8, no. 1, pp. 1–74, Mar. 2021, doi: 10.1186/S40537-021-00444-8.

[8] D. Gyawali, "Comparative Analysis of CPU and GPU Profiling for Deep Learning Models," Sep. 2023, [Online]. Available: http://arxiv.org/abs/2309.02521.

[9] A. S. Al-Ahmad and H. Kahtan, "Cloud Computing Review: Features and Issues," 2018 International Conference on Smart Computing and Electronic Enterprise, ICSCEE 2018, Nov. 2018, doi: 10.1109/ICSCEE.2018.8538387.

[10] Y. Zhao, H. Zhang, L. An, and Q. Liu, "Improving the approaches of traffic demand forecasting in the big data era," Cities, vol. 82, pp. 19–26, Dec. 2018, doi: 10.1016/j.cities.2018.04.015.

[11] T. Staal, "The impact of the Internet of Things on the demand of cloud resources," 2022. Accessed: Jun. 01, 2024. [Online]. Available: https://purl.utwente.nl/essays/91318.

[12] M. Satyanarayanan, "The Emergence of Edge Computing," Computer (Long Beach. Calif)., vol. 50, no. 1, pp. 30–39, Jan. 2017, doi: 10.1109/MC.2017.9.

[13] F. C. Andriulo, M. Fiore, M. Mongiello, E. Traversa, and V. Zizzo, "Edge Computing and Cloud Computing for Internet of Things: A Review," Informatics 2024, Vol. 11, Page 71, vol. 11, no. 4, p. 71, Sep. 2024, doi: 10.3390/INFORMATICS11040071.

[14] J. Lin, L. Zhu, W.-M. Chen, W.-C. Wang, and S. Han, "Tiny Machine Learning: Progress and Futures," IEEE Circuits and Systems Magazine, pp. 8–34, Mar. 2024, doi: 10.1109/MCAS.2023.3302182.

[15] C. R. Banbury et al., "Benchmarking TinyML Systems: Challenges and Direction," Mar. 2020, [Online]. Available: http://arxiv.org/abs/2003.04821.

[16] J. Sen et al., "Machine Learning: Algorithms, Models, and Applications," Jan. 2022, doi: 10.5772/intechopen.94615.

[17] L. Heim, A. Biri, Z. Qu, and L. Thiele, "Measuring what Really Matters: Optimizing Neural Networks for TinyML," Apr. 2021, [Online]. Available: http://arxiv.org/abs/2104.10645.

[18] S. S. Saha, S. S. Sandha, and M. Srivastava, "Machine Learning for Microcontroller-Class Hardware: A Review," IEEE Sens. J., vol. 22, no. 22, pp. 21362–21390, Dec. 2022, doi: 10.1109/JSEN.2022.3210773.

[19] R. A. Mouha and R. A. Mouha, "Internet of Things (IoT)," Journal of Data Analysis and Information Processing, vol. 9, no. 2, pp. 77–101, Mar. 2021, doi: 10.4236/JDAIP.2021.92006.

[20] C. Willson Joseph and G. J. Willsie Kathrine, "Intelligent System with the IoT: A survey on techniques of Artificial Intelligence over the field of Internet of Things," 8th International Conference on Advanced Computing and Communication Systems, ICACCS 2022, pp. 347–351, 2022, doi: 10.1109/ICACCS54159.2022.9785282.

[21] J. F. Burnham, "Scopus database: A review," Mar. 08, 2006. doi: 10.1186/1742-5581-3-1.

[22] C. Manterola, P. Astudillo, E. Arias, and N. Claros, "Revisiones sistemáticas de la literatura. Qué se debe saber acerca de ellas," Cir. Esp., vol. 91, no. 3, pp. 149–155, Mar. 2013, doi: 10.1016/J.CIRESP.2011.07.009.

[23] D. Evans, "The systematic review report," Collegian, vol. 11, no. 2, pp. 8–11, Jan. 2004, doi: 10.1016/S1322-7696(08)60448-5.

[24] M. J. Grant and A. Booth, "A typology of reviews: An analysis of 14 review types and associated methodologies," Health Info. Libr. J., vol. 26, no. 2, pp. 91–108, Jun. 2009, doi: 10.1111/J.1471-1842.2009.00848.X.

[25] R. Sanchez-Iborra and A. F. Skarmeta, "TinyML-Enabled Frugal Smart Objects: Challenges and Opportunities," IEEE Circuits and Systems Magazine, vol. 20, no. 3, pp. 4–18, Jul. 2020, doi: 10.1109/MCAS.2020.3005467.

[26] P. P. Ray, "A review on TinyML: State-of-the-art and prospects," Apr. 01, 2022, King Saud bin Abdulaziz University. doi: 10.1016/j.jksuci.2021.11.019.

[27] N. N. Alajlan and D. M. Ibrahim, "TinyML: Enabling of Inference Deep Learning Models on Ultra-Low-Power IoT Edge Devices for AI Applications," Micromachines (Basel)., vol. 13, no. 6, Jun. 2022, doi: 10.3390/mi13060851.

[28] H. Han and J. Siebert, "TinyML: A Systematic Review and Synthesis of Existing Research," 4th International Conference on Artificial Intelligence in Information and Communication, ICAIIC 2022 - Proceedings, pp. 269–274, 2022, doi: 10.1109/ICAIIC54071.2022.9722636.

[29] B. Moons, D. Bankman, L. Yang, B. Murmann, M. Verhelst, and K. Leuven, "BinarEye: An Always-On Energy-Accuracy-Scalable Binary CNN Processor With All Memory On Chip In 28nm CMOS," in IEEE Custom Integrated Circuits Conference (CICC), 2018. doi: 10.1109/CICC.2018.8357071.

[30] G. Signoretti, M. Silva, P. Andrade, I. Silva, E. Sisinni, and P. Ferrari, "An evolving tinyml compression algorithm for iot environments based on data eccentricity," Sensors, vol. 21, no. 12, Jun. 2021, doi: 10.3390/s21124153.

[31] Z. Ksira, A. Mellit, N. Blasuttigh, and A. Massi Pavan, "A Novel Embedded System for Real-Time Fault Diagnosis of Photovoltaic Modules," IEEE J. Photovolt., vol. 14, no. 2, pp. 354–362, Mar. 2024, doi: 10.1109/JPHOTOV.2024.3359462.

[32] L. S. Martinez-Rau, J. O. Chelotti, L. L. Giovanini, V. Adin, B. Oelmann, and S. Bader, "On-Device Feeding Behavior Analysis of Grazing Cattle," IEEE Trans. Instrum. Meas., vol. 73, pp. 1–13, 2024, doi: 10.1109/TIM.2024.3376013.

[33] J. I. de Oliveira Filho, M. C. Faleiros, D. C. Ferreira, V. Mani, and K. N. Salama, "Empowering Electrochemical Biosensors with AI: Overcoming Interference for Precise Dopamine Detection in Complex Samples," Advanced Intelligent Systems, vol. 5, no. 10, p. 2300227, Oct. 2023, doi: 10.1002/AISY.202300227.

[34] I. Cappelli et al., “Enhanced Visible Light Localization Based on Machine Learning and Optimized Fingerprinting in Wireless Sensor Networks,” IEEE Trans. Instrum. Meas., vol. 72, 2023, doi: 10.1109/TIM.2023.3240220.
[35] G. Peruzzi, A. Pozzebon, and M. Van Der Meer, “Fight Fire with Fire: Detecting Forest Fires with Embedded Machine Learning Models Dealing with Audio and Images on Low Power IoT Devices,” Sensors 2023, Vol. 23, Page 783, vol. 23, no. 2, p. 783, Jan. 2023, doi: 10.3390/S23020783.
[36] A. Mongardi et al., “Hand Gestures Recognition for Human-Machine Interfaces: A Low-Power Bio-Inspired Armband,” IEEE Trans. Biomed. Circuits Syst., vol. 16, no. 6, pp. 1348–1365, Dec. 2022, doi: 10.1109/TBCAS.2022.3211424.
[37] A. Roy, H. Dutta, H. Griffith, and S. Biswas, “An On-Device Learning System for Estimating Liquid Consumption from Consumer-Grade Water Bottles and Its Evaluation,” Sensors 2022, Vol. 22, Page 2514, vol. 22, no. 7, p. 2514, Mar. 2022, doi: 10.3390/S22072514.
[38] A. Krayden et al., “CMOS-MEMS Gas Sensor Dubbed GMOS for SelectiveAnalysis of Gases with Tiny Edge Machine Learning,” Engineering Proceedings 2022, Vol. 27, Page 81, vol. 27, no. 1, p. 81, Nov. 2022, doi: 10.3390/ECSA-9-13316.
[39] F. Dellragnola, U. Pale, R. Marino, A. Arza, and D. Atienza, “MBioTracker: Multimodal Self-Aware Bio-Monitoring Wearable System for Online Workload Detection,” IEEE Trans. Biomed. Circuits Syst., vol. 15, no. 5, pp. 994–1007, Oct. 2021, doi: 10.1109/TBCAS.2021.3110317.
[40] F. Shabani, H. Philamore, and F. Matsuno, “An energy-autonomous chemical oxygen demand sensor using a microbial fuel cell and embedded machine learning,” IEEE Access, vol. 9, pp. 108689–108701, 2021, doi: 10.1109/ACCESS.2021.3101496.
[41] N. Mohan, D. Abdelrahman, N. F. Ali, and M. Atef, “An Integrated High-Gain Wide-Dynamic Range Photoplethysmography Sensor for Cardiac Health Monitoring,” IEEE Sens. J., vol. 24, no. 7, pp. 10375–10384, Apr. 2024, doi: 10.1109/JSEN.2024.3367898.
[42] M. A. O. Zishan, H. M. Shihab, S. S. Islam, M. A. Riya, G. M. Rahman, and J. Noor, “Dense neural network based arrhythmia classification on low-cost and low-compute micro-controller,” Expert Syst. Appl., vol. 239, p. 122560, Apr. 2024, doi: 10.1016/J.ESWA.2023.122560.
[43] E. Tabanelli, D. Brunelli, A. Acquaviva, and L. Benini, “Trimming Feature Extraction and Inference for MCU-Based Edge NILM: A Systematic Approach,” IEEE Trans. Industr. Inform., vol. 18, no. 2, pp. 943–952, Feb. 2022, doi: 10.1109/TII.2021.3078186.
[44] J. Xiao et al., “ULECGNet: An Ultra-Lightweight End-to-End ECG Classification Neural Network,” IEEE J. Biomed. Health Inform., vol. 26, no. 1, pp. 206–217, Jan. 2022, doi: 10.1109/JBHI.2021.3090421.
[45] X. Wang, L. Cavigelli, T. Schneider, and L. Benini, “Sub-100 µW Multispectral Riemannian Classification for EEG-Based Brain–Machine Interfaces,” IEEE Trans. Biomed. Circuits Syst., vol. 15, no. 6, pp. 1149–1160, Dec. 2021, doi: 10.1109/TBCAS.2021.3137290.
[46] M. Zanghieri et al., “An Extreme-Edge TCN-Based Low-Latency Collision-Avoidance Safety System for Industrial Machinery,” IEEE Access, vol. 12, pp. 16009–16021, 2024, doi: 10.1109/ACCESS.2024.3357510.
[47] K. Xu, H. Zhang, Y. Li, Y. Zhang, R. Lai, and Y. Liu, “An Ultra-Low Power TinyML System for Real-Time Visual Processing at Edge,” IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 70, no. 7, pp. 2640–2644, Jul. 2023, doi: 10.1109/TCSII.2023.3239044.
[48] A. Albanese, M. Nardello, G. Fiacco, and D. Brunelli, “Tiny Machine Learning for High Accuracy Product Quality Inspection,” IEEE Sens. J., vol. 23, no. 2, pp. 1575–1583, Jan. 2023, doi: 10.1109/JSEN.2022.3225227.
[49] Y. Zhang, V. Adin, S. Bader, and B. Oelmann, “Leveraging Acoustic Emission and Machine Learning for Concrete Materials Damage Classification on Embedded Devices,” IEEE Trans. Instrum. Meas., vol. 72, 2023, doi: 10.1109/TIM.2023.3307751.
[50] E. Manor and S. Greenberg, “Custom Hardware Inference Accelerator for TensorFlow Lite for Microcontrollers,” IEEE Access, vol. 10, pp. 73484–73493, 2022, doi: 10.1109/ACCESS.2022.3189776.