Jurnal Teknik Informatika (JUTIF)
P-ISSN: 2723-3863
E-ISSN: 2723-3871
Vol. 7, No. 3, June 2026, Page. 2189-2224
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5541

Comprehensive Systematic Review of TinyML Edge Deployment: Optimization Techniques, Application Domains, and Hardware Ecosystems

Very Kurnia Bakti*[1], Arif Setyanto[2], Alva Hendi Muhammad[3], Ferry Wahyu Wibowo[4]

[1]Computer Engineering, Universitas Harkat Negeri, Indonesia
[2,3,4]Informatics, Universitas Amikom Yogyakarta, Indonesia

Email: [1]verykurniabakti@gmail.com

Received: Dec 6, 2025; Revised : Dec 30, 2025; Accepted : Jan 19, 2026; Published : Jun 15, 2026

Abstract

The Internet of Things (IoT) is growing rapidly, making it even more crucial to deploy Machine Learning (ML) models directly on edge devices with limited resources. TinyML fixes this matter by giving microcontroller-class hardware the ability to think for itself. This makes it less reliant on the cloud and better for latency, energy efficiency, and data privacy. This study offers a comprehensive Systematic Literature Review (SLR) of TinyML research published between 2021 and 2025, in accordance with PRISMA principles. We identified 429 records, removed 326 duplicates, and added 83 studies to the final synthesis. The evaluation examines five research inquiries concerning optimization techniques, streamlined architectures, sophisticated learning frameworks, application sectors, and hardware ecosystems. The findings underscore four key themes: enhancing models, utilizing specialized tools and technology, and adapting strategies. Some of the challenges that keep recurring are broken ecosystems, different benchmarking approaches, and on-device learning that isn't compelling when ideas shift. This research presents an open-access taxonomy that categorizes optimization techniques, application trends, and hardware constraints, thereby laying the foundation for a TinyML research agenda within the informatics community. Future directions highlight the importance of adaptive TinyMLOps pipelines, federated learning, LLM-assisted model design, and NVM-based computing to support scalable and sustainable edge intelligence. The results underscore the relevance of TinyML for advancing informatics and computer science, particularly in enabling secure, efficient, and environmentally aligned IoT systems that support SDG 9 and SDG 12.

Keywords: *Edge AI, Hardware Ecosystems, On-Device Learning, TinyML Applications, TinyML Optimization, TinyMLOps*

1. INTRODUCTION

The rapid growth of the Internet of Things (IoT) has created significant demand for developing and deploying Machine Learning (ML) models on resource-constrained edge devices. This framework, termed TinyML, enables intelligent processing directly on devices with limited computing and energy resources, thereby expanding the applicability of machine learning to various real-world scenarios[1], [2]. Machine learning has emerged as a crucial component of modern research, revolutionizing data processing through techniques such as Neural Networks, Deep Learning, and clustering. These approaches enable the extraction of complex patterns and insights from large datasets, thereby improving computational intelligence and supporting diverse scientific applications [3], [4], [5].

Traditionally, machine learning models, particularly advanced Deep Neural Networks (DNNs), require substantial computational power and memory resources. As a result, their implementation has been chiefly limited to cloud servers or high-performance devices, thereby constraining its applicability in resource-constrained environments[6], [7], [8]. The cloud-centric model presents several scalability issues, including communication latency, increased energy consumption, bandwidth constraints, and privacy vulnerabilities. These limitations impede the effective implementation of ML models in real-

time applications and underscore the need for alternative paradigms, such as TinyML and Edge AI, to mitigate resource and security issues[9], [10], [11]. The Generative Pre-trained Transformer (GPT-3) utilizes a network with 175 billion parameters and requires approximately three gigawatt-hours of electricity for training[12], [13].

A trend called TinyML has emerged to address these challenges, connecting microcontroller units (MCUs) with machine learning (ML)[14], [15] TinyML emphasizes the implementation of deep learning inference models on ultra-low-power devices with stringent computational and memory constraints [16], [17]. Typical TinyML devices, including 32-bit microcontrollers (e.g., Raspberry Pi RP2040 or ARM Cortex-M4), exhibit severely constrained resources: Flash memory typically does not exceed 1 MB, SRAM is quantified in kilobytes, and clock speeds are comparatively low [18], [19], [20], [21]. The Arduino Nano 33 BLE Sense offers merely 256 KB of RAM and 1 MB of Flash memory [22], [23]. TinyML enables advanced analytics by processing data locally at the sensor level, thereby reducing reliance on continuous data transmission to the cloud. This approach inherently enhances energy efficiency, extends battery longevity, facilitates real-time responsiveness through reduced latency, and safeguards the confidentiality of sensitive data[24]. TinyML, while it has significant potential, encounters specific challenges in its development. Machine learning models must be rigorously optimized using compression techniques such as quantization and pruning to meet the memory constraints of microcontroller units, potentially reducing accuracy. [25], [26], [27]. The lifecycle of TinyML, commonly known as TinyMLOps, is more complex than conventional machine learning, encompassing additional processes such as model optimization and conversion to formats suitable for TinyML frameworks (e.g., TensorFlow Lite). Furthermore, the TinyML environment is markedly fragmented, exhibiting considerable variability in hardware and software, which obstructs direct comparisons and scalability[28], [29], [30]. The absence of a cohesive framework makes platform-specific implementations unscalable[31], [32].

By the end of 2025, the worldwide network is expected to have more than 21 billion IoT devices. This shows how quickly the network is growing and how sensors and smart things are becoming more common in a large-scale digital ecosystem. This growth needs intelligence at the edge that is faster, safer, and uses less energy. TinyML is a critical technology that makes low-power on-device learning possible, which is in line with global aspirations for innovation and sustainability[33]. The simultaneous development of cloud, edge, and on-device learning has led to the creation of hybrid computing architectures that make real-time processing, privacy, and efficiency better. However, there are still problems with limited resources, security, and scalability[34], [35], [36]. The development of TinyML shows a move away from machine learning that relies on the cloud and toward more flexible intelligence on devices. From 2015 to 2018, when the cloud was popular, ML models relied a lot on data center compute, but they had problems with latency, energy use, and privacy [37]. During the early edge AI phase (2019–2021), optimization methods like quantization and pruning made it possible to make inferences on microcontrollers and single-board computers [38]. During the TinyML expansion period (2022–2024), this momentum picked up speed thanks to lightweight frameworks like TensorFlow Lite Micro and Edge Impulse and a wide range of uses in healthcare, predictive maintenance, and smart city systems. However, there were still problems with benchmarking and ecosystem fragmentation [39] As we move into 2025, research is focusing more and more on adaptive TinyMLOps pipelines, federated learning, and incremental on-device training. This shows that cloud, edge, and local intelligence are all coming together to create scalable and sustainable edge AI [33], [40]. This transition from cloud-based ML to adaptive on-device learning is shown in Figure 1.

Vol. 7, No. 3, June 2026, Page. 2189-2224
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5541

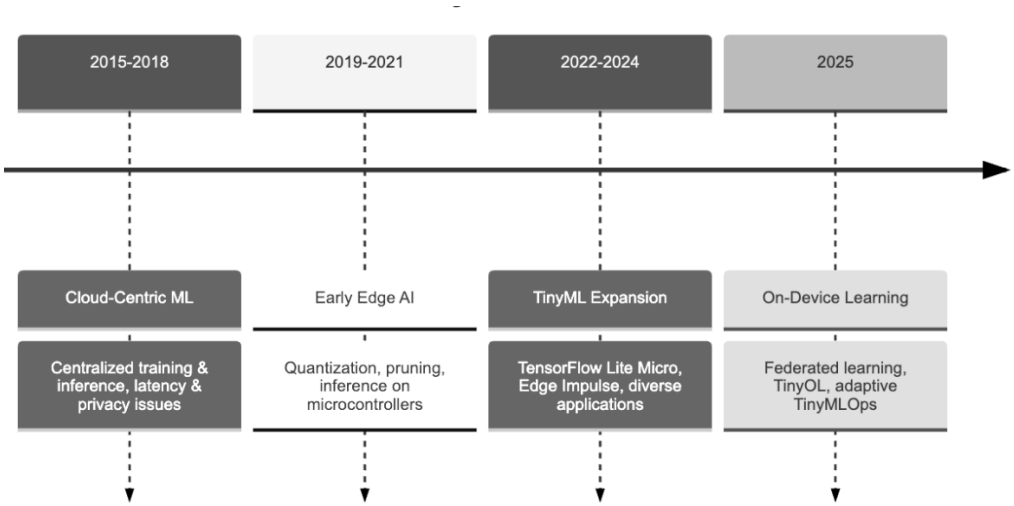

Figure 1. Timeline of TinyML evolution

A Systematic Literature Review (SLR) is necessary for assessing the advancements, tools, and obstacles in the integration of Machine Learning (ML) with embedded systems. Prior studies underscore the necessity for equitable and pragmatic benchmarking standards to facilitate meaningful comparisons among various TinyML implementations, hence ensuring repeatability, scalability, and reliable performance evaluation in this swiftly advancing domain [41]. In response to this demand, the current SLR brings together optimization methods, looks at current hardware and software frameworks, sorts new application areas, and lists the problems with current TinyMLOps practices. This evaluation brings together many points of view to give a full picture of the current state of TinyML, find areas where standardization and scalability are lacking, and suggest ways to make TinyML solutions that are more efficient and work with other systems.

This work offers four significant additions to the TinyML research domain. Initially, it offers the most current and methodologically sound systematic literature review (SLR) of TinyML from 2021 to 2025, incorporating PRISMA 2020, MMAT, and bibliometric analysis to guarantee thorough coverage and analytical rigor. Secondly, it presents a three-dimensional taxonomy that consolidates optimization approaches, application domains, and hardware ecosystems, thereby resolving the fragmentation identified in previous surveys. Third, it conducts a multi-gap analysis that reveals ten unresolved research difficulties, such as benchmarking inconsistencies, hardware heterogeneity, and the increasing necessity for adaptive TinyMLOps and federated TinyML. This review consolidates temporal, thematic, and methodological trends to propose a future-oriented research agenda, providing practical insights for enhancing scalable, efficient, and interoperable TinyML systems.

2. RELATED WORK

Numerous prior TinyML surveys identify significant shortcomings that the current systematic literature review (SLR) addresses. The study by Lamaakal et al. (2025) focuses solely on TinyML for Human Behavior Analysis (HBA), yielding a limited viewpoint that fails to address broader technical issues in TinyML. Key elements such as model optimization, hardware heterogeneity, federated TinyML, memory limitations, and latency issues on ultra-low-power systems are not addressed. Moreover, the study lacks a technical taxonomy and a systematic gap analysis.[42]. The survey by Capogrosso et al. (2024) employs a machine-learning perspective, highlighting ML workflows, co-design methodologies, and optimization strategies. Despite being methodologically organized, it does not constitute a comprehensive systematic literature review and lacks gap analysis, bibliometric mapping, or a multidimensional taxonomy. Moreover, some nascent research avenues, such as RISC-V

Vol. 7, No. 3, June 2026, Page. 2189-2224
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5541

acceleration, external-memory overlaying, model attestation, and federated TinyML, are not addressed, thereby constraining the survey's comprehensiveness.[43]. The preliminary systematic literature review by Han and Siebert (2022) offers a foundational synthesis of TinyML research but is constrained in scope, focusing mainly on hardware, frameworks, datasets, and applications. The review omits model-level optimization methodologies and fails to address current challenges, including concept drift, sub-byte quantization, and distributed inference. Furthermore, its scope is limited to material published from 2019 to 2021, making it inadequate for encompassing recent developments in TinyML[44].

Conversely, the current systematic literature review provides a more thorough and contemporary examination of TinyML research from 2021 to 2025. This study presents a three-dimensional taxonomy encompassing model optimization, application domains, and hardware ecosystems; performs a multi-gap heatmap analysis ten identify 10 unresolved research gaps; and employs PRISMA 2020 and MMAT to ensure methodological rigor. This thorough literature review includes bibliometric mapping and rigorously analyzes new research issues such as federated TinyML, LLM-assisted model creation, RISC-V-based acceleration, external-memory approaches, model attestation, and distributed inference. These contributions establish the current SLR as the most exhaustive and progressive assessment of TinyML to date.

3. RESEARCH QUESTION

The Systematic Literature Review (SLR) technique requires the formulation of precise and focused research questions to guide the inquiry, reduce potential bias, and systematically address the knowledge gaps identified in the pertinent literature, as outlined by Kitchenham and Charters[45], [46]. Research questions determine the study's objectives and ensure that data collection and analysis are methodical, focused, and aligned with the established scope. The research questions (RQs) augment methodological rigor and ensure consistency throughout the investigation by providing explicit direction[47]. This review aims to assess TinyML and Edge AI, as presented in Table 1 five Research Questions (RQs) Established for TinyML and Edge Computing.

Table 1. Research Questions on TinyML and Edge Computing

RQ	Research Question
RQ1	Which TinyML model compression techniques (e.g., quantization, pruning, clustering, distillation, NAS) most effectively balance predictive accuracy with resource limitations, and how are the trade-offs among model size, inference speed, and accuracy quantitatively assessed on microcontroller-class devices?
RQ2	How can lightweight neural network architectures enhance efficiency on resource-constrained edge devices? This study investigates optimization mechanisms that balance accuracy, memory footprint, and latency in TinyML deployment.
RQ3	In what ways are advanced learning paradigms (Federated Learning, Transfer Learning, Online Learning) tailored and enhanced for TinyML to tackle issues of data privacy, device heterogeneity, and concept drift on low-power edge devices?
RQ4	Which application domains (Human Activity Recognition, Computer Vision, Predictive Maintenance, Anomaly Detection) most extensively utilize TinyML, and how is model performance quantitatively evaluated regarding accuracy, inference latency, memory footprint (RAM/Flash), and energy consumption on constrained embedded systems?
RQ5	In what ways does the TinyML hardware ecosystem, characterized by its intrinsic resource limitations and diverse designs, influence the implementation issues and impact the performance reliability of edge AI applications?

Vol. 7, No. 3, June 2026, Page. 2189-2224
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5541

4. METHODS

4.1. Search Queries

The literature search was conducted across two principal scientific archives, IEEE Xplore and Scopus, owing to their comprehensive indexing of peer-reviewed journals, conference proceedings, and technical publications relevant to TinyML and Edge AI. To improve coverage and mitigate database bias, supplementary grey literature was sourced from arXiv, specifically concerning nascent TinyML structures and optimization methodologies not yet defined in academic journals. Search strings were formulated using Boolean operators to combine synonymous terms and narrow the retrieval scope. The OR operator combined synonymous concepts (e.g., "TinyML" OR "Tiny Machine Learning"), whereas the AND operator connected fundamental topics such as TinyML and Edge Computing. The NOT operator was utilized to eliminate extraneous domains, including cloud-centric research that does not pertain to on-device inference, as detailed in the entire list of keywords in Table 2.

Table 2. Search string and detailed search keywords

Database	Basic Keyword	Detailed Keyword
Scopus	"Edge computing" OR "edge intelligence"	"Edge computing" OR "Edge AI" AND "TinyML" AND NOT "Review" AND NOT "Comparative" AND NOT "Survey" AND NOT "Comprehensive"
IEEE	"Edge computing" OR "edge intelligence"	"Edge computing" OR "Edge AI" AND "TinyML" NOT "Review" NOT "Comparative" NOT "Survey "NOT "Comprehensive"
Arxiv	"Edge computing" OR "edge intelligence"	"Edge computing" OR "Edge AI" AND "TinyML" AND NOT "Review" AND NOT "Comparative" AND NOT "Survey" AND NOT "Comprehensive"

4.2. Screening and PRISMA Flow

The study selection method adhered to the PRISMA 2020 principles to guarantee systematic identification, screening, and inclusion of pertinent material. The preliminary search produced 53,087 records from Scopus and 46,480 from IEEE Xplore. After applying comprehensive keyword filters, the dataset was reduced to 429 potentially relevant papers, including 13 grey literature entries from arXiv. A deduplication process eliminated 326 redundant entries, yielding 103 distinct items. Title and abstract screening eliminated 10 studies that were not consistent with the TinyML scope. Ninety-three publications were subjected to full-text evaluation, resulting in the exclusion of ten additional studies for not satisfying the inclusion criteria (e.g., absence of performance measurements, irrelevant context to TinyML, inadequate approach).

In conclusion, 83 studies were included in the final synthesis. This multi-stage selection procedure guarantees methodological rigor, transparency, and reproducibility. The entire workflow is depicted in Figure 2 (PRISMA Flowchart).

Vol. 7, No. 3, June 2026, Page. 2189-2224
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5541

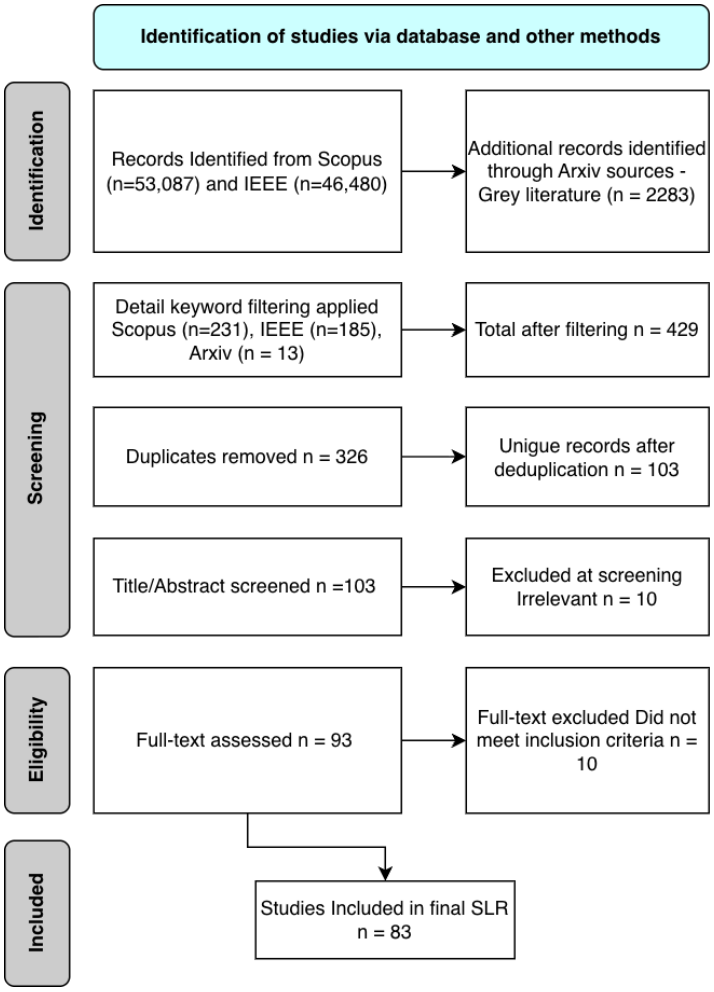

Figure 2. PRISMA flow diagram illustrating the study selection process.

4.3. Inclusion and Exclusion Criteria

In conducting this systematic literature review, we defined explicit inclusion and exclusion criteria to ensure the analysis was robust, transparent, and directly pertinent to the study's objectives. The inclusion criteria mandated that each selected article explicitly focused on TinyML, namely its implementation on low-power, resource-constrained systems. Only articles issued between 2021 and 2025 were deemed relevant, as this period encapsulates the latest achievements in the subject. Additionally, we restricted our sources to peer-reviewed journals, conference proceedings, and academic books to guarantee the material's legitimacy. Each submission was anticipated to deliver empirical or technical material, including practical applications, experimental assessments, or hardware performance evaluations. Ultimately, research must focus on optimization or compression methodologies tailored to resource-constrained devices, including quantization, pruning, clustering, and neural architecture search. Conversely, many exclusion criteria were employed to eliminate irrelevant or incomplete works. Research exclusively focused on non-TinyML environments, such as cloud- or data-center-based machine learning, was omitted. Works with partial texts or absent methodological details were excluded, as were those released outside the 2021–2025 period. Abstract-only papers and duplicate entries were removed to preserve the dataset's integrity. Finally, experiments that failed to give TinyML-specific performance data, including latency, model size, or energy consumption, were omitted, as these metrics are essential for assessing the viability of TinyML in edge contexts.

Vol. 7, No. 3, June 2026, Page. 2189-2224
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5541

To guarantee transparency and reproducibility in the selection process, we established explicit inclusion and exclusion criteria. An article was incorporated into the systematic literature review if it met all of the following criteria:

$$\text{IC} = \left(T_{TinyML} \wedge Y_{2021-2025} \wedge P_{PeerReviewed} \wedge C_{Empirical/Technical}\right)$$

Conversely, an article was excluded if it met any of the following conditions:

$$EC = \left(N_{NonTinyML} \vee O_{OutOfRange} \vee I_{Incomplete} \vee D_{Duplicate} \vee M_{MissingMetrics}\right)$$

Finally, a study was selected for final analysis only if it fulfilled the inclusion criteria and did not meet any exclusion criteria, expressed as:

$$Final = IC \wedge \neg EC$$

The criteria were designed to mitigate bias and ensure that the included papers are directly relevant to TinyML and Edge AI, thereby enhancing the review's validity and focus [26]. The specific requirements are depicted in Figure 3.

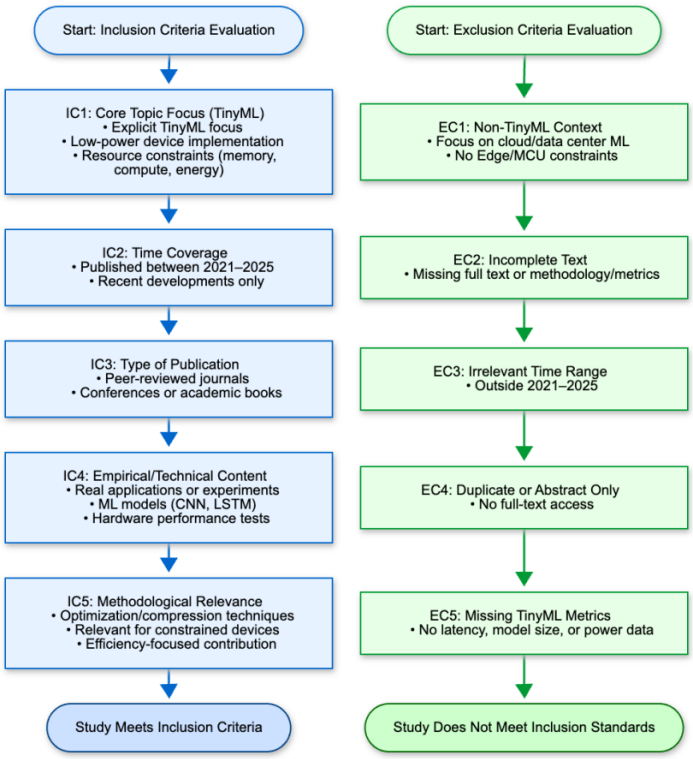

Figure 3. Inclusion and exclusion criteria.

4.4. Quality Assesment

A structured quality assessment was performed using a modified version of the Mixed Methods Appraisal Tool (MMAT) 2018 to ensure the methodological rigor of the selected research, which is commonly utilized in systematic reviews with diverse study designs. Each study was assessed based on five criteria: (1) clarity of research objectives, (2) appropriateness of methodology, (3) completeness of experimental details, (4) reporting of TinyML-specific performance metrics (latency, model size, RAM/Flash usage, energy consumption), and (5) reproducibility of results. Each criterion was evaluated on a binary scale (1 = fulfills the criterion, 0 = does not meet the criterion), yielding a total score of 0-5. Studies with scores of ≥3 were included in the final synthesis, whereas those with lower scores were excluded due to inadequate methodological transparency or absent performance indicators. This

Vol. 7, No. 3, June 2026, Page. 2189-2224
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5541

evaluation process guarantees that only studies with sufficient empirical support and reproducibility are included in the synthesis, thereby minimizing bias and enhancing the review's validity.

4.5. VOS viewer Co-Occurrence Threshold

A co-occurrence network was created using VOS viewer 1.6. x to examine theme trends and conceptual linkages in the TinyML literature. The bibliometric dataset was made from the titles, abstracts, and author keywords of the 83 included studies. A minimum occurrence threshold of 5 was implemented to facilitate significant grouping and diminish noise from rare phrases. Terms that appeared fewer than five times in the corpus were omitted from the network to preserve semantic coherence and prevent fragmentation of clusters. After applying the threshold, 112 high-frequency words were retained and clustered using the default VOSviewer association-strength normalization. The resulting clusters identified three predominant thematic categories: (1) optimization methods (quantization, pruning, compression), (2) application sectors (HAR, predictive maintenance, computer vision), and (3) hardware frameworks (MCUs, ESP32, embedded systems). This threshold-based filtering ensures that the visualization reflects substantial thematic structures rather than isolated or infrequent phrases. Figure 4 illustrates a co-occurrence network built by VOSviewer, depicting the conceptual landscape of TinyML and Edge AI research from 2021 to 2025.

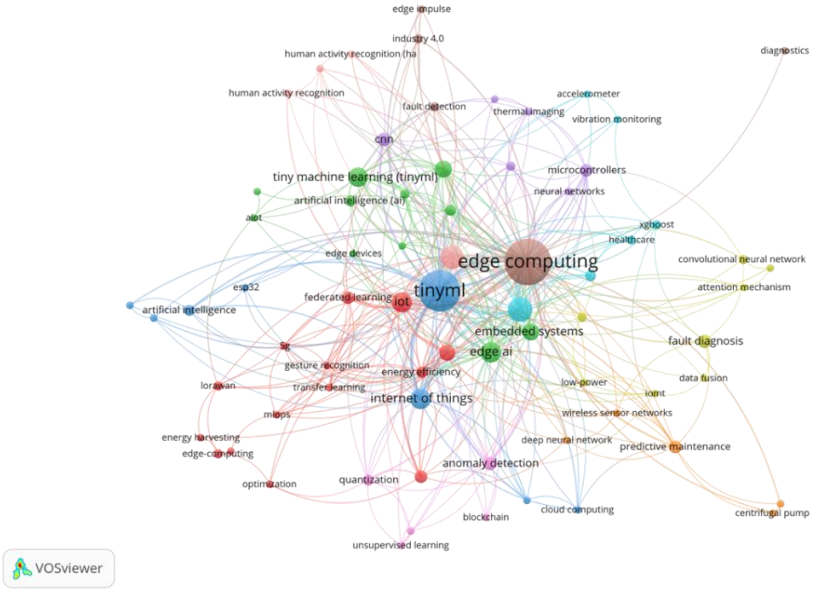

Figure 4. VOSviewer network of TinyML and Edge AI.

4.6. Risk of Bias

The systematic review process sought to minimize potential biases and ensure methodological transparency in accordance with PRISMA 2020 guidelines. Systematic literature reviews may be subject to several forms of selection bias, including database bias, publication bias, language bias, and researcher screening bias. This review used many sources, including Scopus, IEEE Xplore, and grey literature from arXiv, to mitigate database bias, improve coverage, and reduce reliance on a single indexing platform. Publication bias was alleviated by incorporating pertinent non-peer-reviewed technical preprints, thereby ensuring that emerging TinyML research not yet published in journals was not systematically excluded. Language bias was mitigated by applying standardized English-language filtering across all databases. A multi-stage selection method was used to reduce screening bias, including title and abstract screening followed by full-text assessment according to predetermined inclusion and exclusion criteria. All exclusion decisions were documented with explicit rationale to guarantee transparency and reproducibility. Duplicate records were meticulously removed to prevent

Vol. 7, No. 3, June 2026, Page. 2189-2224
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5541

the overrepresentation of specific studies. The extensive selection process is illustrated in the PRISMA flowchart (Figure 2), providing a clear audit trail of the identification, screening, eligibility, and inclusion phases. These methodologies, when combined, ensure that the final dataset accurately reflects a balanced, methodologically rigorous representation of TinyML research from 2021 to 2025.

5. RESULTS

The Results section provides a comprehensive summary of the five research questions (RQ1-RQ5) on the implementation of Tiny Machine Learning (TinyML) on resource-constrained edge devices. The results fundamentally confirm TinyML's ability to achieve its primary goal: real-time inference on ultra-low-power Internet of Things (IoT) devices.

5.1. Synthesis Summary

5.1.1. Results for RQ1: Model Compression Techniques

Within the framework of enhancing predictive accuracy under the severe resource limitations of TinyML, INT8 Quantization proves to be the most effective approach, yielding a significant reduction in model size with negligible effects on performance. In situations requiring extreme compression, the AuGQ method transforms the process, enabling exact precision even at the sub-byte level. We are witnessing exceptional results from efficient designs like FastKAN, which has demonstrated supremacy by achieving 97.4% accuracy while keeping the model size at just 120 KB. We use Pareto Analysis to clarify the inherent trade-offs among accuracy, speed, memory (SRAM/Flash), and energy consumption. A hybrid approach that integrates pruning and quantization generally attains the optimal equilibrium for edge devices. The results from multiple research studies are thoroughly displayed in Table 3.

Table 3. Taxonomy of TinyML Optimization Techniques for Edge Devices.

Category	Specific Technique	Mechanism Description	References
Quantization	Augmented Granularity (AuGQ)	Partition the FP32 weight range into small chunks and apply affine quantization independently to mitigate accuracy degradation in sub-byte (2-4 bit).	[48]
	Binarization (1-bit)	Convert weights and activations to binary values (+1, 1) to maximize memory savings and inference speed using XNOR/PopCount operations.	[49], [50]
	BFloat16 Deployment	Use the 16-bit floating-point format (Brain Floating Point) for Transformer models to avoid the complexity of integer quantization while maintaining dynamic range.	[51]
	Post-Training Quantization (INT8)	Standard conversion of FP32 weights to INT8 after training. Often combined with TensorFlow Lite Micro for MCU compatibility.	[52], [53]
Pruning	Hardware-Aware Pruning	Weight pruning strategy balanced for hardware (FPGA) to maintain a uniform workload across parallel processing units.	[54]

Vol. 7, No. 3, June 2026, Page. 2189-2224
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5541

	Structured Pruning (ADMM)	Use the Alternating Direction Method of Multipliers to remove redundant filters or channels in a structured manner.	[55]
	Magnitude-based Pruning	Remove connections with the smallest absolute weight values to reduce model size and computational complexity.	[56]
Efficient Architectures	FastKAN (Kolmogorov-Arnold)	Replace traditional activation functions with learnable Gaussian Radial Basis Functions (RBFs) at network edges for parameter efficiency.	[57]
	Hybrid CNN-LSTM	Combine convolutional layers for spatial feature extraction and LSTM for temporal features, optimized for human activity recognition.	[58]
	Tiny Transformers	Adapt Transformer architectures (e.g., MobileBERT, TinyViT) through aggressive encoder layer pruning to fit MCU memory.	[51]
	Multi-Model Fallback	A dynamic system using a small (quantized) model by default and switching to a larger model only if the confidence score is low.	[45
	Hierarchical Inference & Early Exit	Use Early Exit or tiered inference (Device-Edge-Cloud) to balance latency and accuracy.	[59]
	Tiling & Overlaying (TinyOps)	Split large tensor operations into small tiles and use DMA to move data between external and internal memory (SRAM) efficiently.	[60]
	On-Device Online Learning (TinyOL)	Enable models to update weights (usually the last layer) incrementally on-device to adapt to concept drift.	[61]

5.1.2. Results for RQ2: Lightweight Neural Architectures

Recent advancements in Tiny Machine Learning (TinyML) have facilitated the deployment of deep learning models on resource-constrained edge devices. To guarantee practical feasibility, lightweight architectures such as MobileNet, SqueezeNet, ShuffleNet, EfficientNet-Lite, DS-CNN, MCUNet/TinyNAS, FastKAN, and FusionGCNN have been created to reduce parameter counts, memory consumption, and multiply–accumulate (MAC) operations while maintaining predictive accuracy. These architectures incorporate efficiency techniques such as depthwise separable convolutions, pruning, quantization, radial basis functions, and neural architecture search, tailored for application domains including image recognition, audio keyword detection, time-series forecasting, and biomedical signal analysis. The complete results are summarized in Table 4.

Vol. 7, No. 3, June 2026, Page. 2189-2224
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5541

Table 4. Lightweight Architecture

Model Architecture	Description & Efficiency Mechanism	References
MobileNetV1/V2/V3	Uses Depthwise Separable Convolutions and Inverted Residuals to reduce parameters and MAC operations compared to standard CNNs drastically.	[52], [62], [63]
FastKAN / XTiny-FastKAN	Efficient variant of Kolmogorov-Arnold Networks replacing fixed node activation functions with learnable radial basis functions (RBFs) at the edge, offering high accuracy with a tiny memory footprint (~35KB).	[57], [64]
FOMO (Faster Objects, More Objects)	Simplified object detection model derived from MobileNetV2, designed for microcontrollers by replacing bottleneck layers with pointwise convolutions, optimized for small object detection.	[65]
SqueezeNet	Replaces 3x3 filters with 1x1 filters (squeeze technique) to reduce parameters up to 50x compared to AlexNet while maintaining accuracy.	[62], [64]
ShuffleNet V1/V2	Uses Pointwise Group Convolution and Channel Shuffle to overcome computational bottlenecks in 1x1 convolutions, balancing accuracy and latency.	[62], [64]
EfficientNet-Lite	A variant of EfficientNet optimized for edge (without squeeze-and-excitation), using compound scaling to balance depth, width, and resolution.	[62], [64]
DS-CNN (Depthwise Separable CNN)	Architecture that separates spatial filtering and feature mixing, highly effective for audio applications (Keyword Spotting) on low-power FPGA and MCU.	[54]
MCUNet & TinyNAS	Neural Architecture Search (NAS) approach that automatically designs network architectures to fit specific microcontroller memory constraints (SRAM/Flash).	[62], [66]
Optimized / Tiny LSTM	Pruned and quantized LSTM variants for time-series forecasting (e.g., energy consumption) on edge devices.	[67], [68]
FusionGCNN	Lightweight Spatiotemporal Graph Convolutional Network for ECG arrhythmia detection, using regularization and feature fusion for computational efficiency.	[69]
Shallow Neural Networks (SNN)	Shallow neural networks (few hidden layers) optimized for ultra-low latency and anomaly detection in vibration/acoustic sensor signals.	[70]

5.1.3. Results for RQ3: Advanced Learning Paradigms

This part investigates the implementation and optimization of advanced learning paradigms for TinyML, aiming to identify key mechanisms that enable efficient deployment on resource-constrained devices. The investigation addresses significant challenges, including data privacy, energy efficiency,

Vol. 7, No. 3, June 2026, Page. 2189-2224
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5541

and device heterogeneity, while highlighting improvements in performance metrics. The integration of these processes and their corresponding challenges is encapsulated in Table 5.

Table 5. Implementation and Optimization of Advanced Learning Paradigms for TinyML

Learning Paradigm	Key Implementation and Optimization Mechanisms	Challenges Addressed and Improved Performance Metrics	Supporting References
Federated Learning (FL)	Mechanism: Devices (clients) train models locally using their private data and only send parameter updates (weights), not raw data, to the central server for aggregation (FedAvg).	1. Data Privacy: Raw data never leaves the local device, minimizing security risks.	[12], [16], [71], [72], [73], [74], [75], [76], [77], [78], [79]
	Resource Optimization: Apply 8-bit quantization to weights before transmission to reduce communication overhead. Integrate Differential Privacy (DP) and model update encryption.	2. Energy Efficiency: FL reduces energy consumption by 33% compared to centralized learning, achieving an average of 100 mW. DP maintains precision with minimal loss, only 1.2%.	[72], [79], [80]
	TinyMetaFed (FL + Meta-Learning): A federated meta-learning framework combining online learning. Uses top-P% selective communication and partial local reconstruction.	3. Device Heterogeneity: Produces model initialization that quickly adapts to new tasks or conditions on different devices with minimal labeled data. TinyMetaFed achieves 65% energy savings and a 42% reduction in communication costs.	[72], [73], [74], [79], [81], [82]
Transfer Learning (TL)	Basic Principle: The model is first trained on a large dataset, then fine-tuned on edge devices using small, task-specific datasets.	Limited Labeled Data: Accelerates TinyML development where labeled data is scarce or expensive.	[83], [84], [85], [86], [87], [88] [73], [74], [83], [89]
	TinyTL (Tiny Transfer Learning): Designed to reduce training memory footprint by freezing network weights and only updating biases.	Training Memory Reduction: TinyTL reduces training memory cost by 4.6×. Reduces training memory from over 250 MB to only 16 MB.	[73], [74], [83], [90]
Online Learning (TinyOL)	Mechanism: Enables incremental on-device training on MCUs (e.g., Arduino) using streaming data (one data sample at a time). Data can be discarded after each update.	Concept Drift/Data Drift: Enables deployed static models to adapt efficiently to dynamic environmental changes.	[6], [18], [69], [81], [91], [92], [93], [94], [95]

Vol. 7, No. 3, June 2026, Page. 2189-2224
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5541

	TinyOL Optimization: Uses model-agnostic online learning principles requiring minimal resources. TEDA-RLS is an unsupervised incremental learning algorithm to detect and correct outliers in data streams.	Resource Overhead: TinyOL runtime requires only about 7 KB of RAM and 135 KB Flash. TinyOL training has very low computational overhead; training time is comparable to inference time. TinyOL can improve accuracy by up to 12.4% after deployment by adapting to augmented test data.	[89], [90], [96], [97]

5.1.4. Results for RQ4: Application Domains

Examines the application domains of TinyML and evaluates critical quantitative performance metrics. The objective is to assess the deployment and enhancement of TinyML across domains such as predictive maintenance, computer vision, human activity recognition, and healthcare. This section highlights critical performance parameters, including accuracy, inference delay, memory utilization, and energy consumption, which assess the feasibility of deploying TinyML models on resource-constrained devices. The thorough integration of these factors is depicted in Table 4.

Table 4. Quantitative performance metrics of TinyML across diverse application domains.

Application Domain	TinyML Implementation and Optimization	Key Quantitative Performance Metrics	Supporting References
Industrial PdM and AD	Goal: Detect abnormal behavior on equipment or IoT systems directly on-device to reduce latency and energy consumption for data transmission. Use Isolation Forest or Shallow Neural Networks (SNN) on MCUs for anomaly detection in extreme industrial environments. Apply quantized DL models (e.g., LSTM) optimized with pruning, 8-bit quantization, and knowledge distillation.	1. Accuracy (AD): SNN achieves 95.0% accuracy and an F1-score 94.0%. Motor fault detection reaches 96.5%. 2. Inference Latency (IT): Isolation Forest detects anomalies in <16 ms. PdM sensor model IT: 14.00 ms. 3. Memory Footprint: Isolation Forest requires 80 KB of RAM for training 50 trees. 4. Energy Consumption (EC): Quantized SNN (Int8) reduces energy by 60% compared to non-compressed SNN.	[18], [98], [99], [100]
Cyber Threat Detection (NIDS)	Adapt classical ML/DL models (Tiny RF, Tiny DT, Tiny MLP, Tiny LSTM) and optimize using quantization and pruning for intrusion detection on IoT/6G devices.	1. Accuracy/F1 Score: Tiny RF/DT achieves F1-score >0.99 on CICIDS2017. Tiny DT accuracy 97.94% (CICIoT2023). 2. Inference Latency: Tiny LSTM latency 5.5 ms. 3. Energy Consumption: Tiny	[101]

Vol. 7, No. 3, June 2026, Page. 2189-2224
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5541

		MLP consumes 3 J to 4 J on IoT datasets.	
Computer Vision (CV) Object Detection & Inspection	Enables classification, object detection, and real-time quality inspection on low-power devices—optimized using 8-bit quantization and Binarized Neural Networks (BNN). Bolt defect detection using TOD-CMLNN on climbing robots. Olive fruit classification using CNN on MCUs. SSD MobileNet V2 implemented with 8-bit quantization for video surveillance.	1. Accuracy/mAP: TOD-CMLNN achieves 72.46% mAP. Olive fruit classification (CNN) shows high accuracy. Vinegar classification using MTF/LeNet-5 only 60.20%, lower than RF (93.50%). 2. Inference Latency: SSD MobileNet V2 (Int8) total latency 5 ms. 3. Memory Footprint: MiniYOLO model size 4.2 MB. Binarization reduces memory up to 32×. 4. Quantization Optimization: Dynamic Range Quantization reduces size by 91.6% with only -0.10% accuracy drop.	[102], [103], [104]
Human Activity Recognition (HAR), Healthcare (IoMT), Gesture Recognition	Used for motion classification (IMU, EMG) and real-time physiological signal monitoring with low latency requirements. Quantized DeepConv LSTM (Int8) for HAR. XTiny-FastKAN for air handwriting recognition (Tifinagh). Rasterization converts spatiotemporal data into grid-based images.	1. Accuracy: Quantized DeepConv LSTM maintains 97% accuracy. XTiny-FastKAN achieves 96.6%. Epileptic Seizure Detection (Arch 4) reaches 99%. 2. Inference Latency: XTiny-FastKAN latency 0.04 ms. HAR (DeepConv LSTM) averages 21 ms. 3. Memory Footprint (Flash): DeepConv LSTM reduced from 513.23 KB to 136.51 KB. XTiny-FastKAN only 35 KB.	[22], [105]
	Anomaly detection in ECG using optimized and quantized models. Driver drowsiness detection using embedded networks. Li-Ion battery SoC prediction using LSTM.	1. Regression Metrics: SoC prediction evaluated using RMSE and R^2. 2. Quantization: Applied to EEG-sensitive models. Quantization from 32-bit float to 8-bit integer (Int8) optimizes memory and computation.	[68], [106], [107], [108], [109]

Vol. 7, No. 3, June 2026, Page. 2189-2224
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5541

Core Quantitative Metrics for TinyML Accuracy Metrics	Metrics define the feasibility of deploying models on low-power embedded devices, often involving trade-offs between accuracy and resource efficiency. Accuracy (TP+TN)/(TP+TN+FP+FN) and F1-score (2·P·R)/(P+R) are standard. F1-score is crucial for imbalanced datasets and trigger response monitoring.	1. Regression: For time-series prediction (e.g., SoC estimation, shelf life), main metrics are RMSE, MAE, MAPE, and R^2. 2. Classical ML: Performance of SVM, RF, k-NN, DT, and NB tested as benchmarks against DL.	[3], [24], [83], [91], [98], [110], [111]
Inference Latency (IT) & Cycles	Time required to process one sample on edge. Good IT is <50 ms. Hardware-level measurement often uses cycles.	1. Highest/Lowest Latency: Lowest reported latency is 0.04 ms. 2. Optimization: Hybrid pruning (Keras) can make inference 91% slower than non-quantized models, showing significant trade-offs. 3. DT Optimization: DT-Arr kernel (stall-free) achieves IT 2.04 kCycles with near-ideal CPI of 1.16.	[22], [112]
Memory Footprint (RAM/Flash/ROM)	Amount of RAM (for runtime activations) and Flash/ROM (for model storage). Models must fit within KB limits.	1. Quantization Reduction: From 32-bit float to 8-bit integer (Int8) reduces model size up to 4×. 2. Lightweight Models: XTiny-FastKAN uses 35 KB Flash. DT-Rec packed reduces the memory footprint by 36.7%.	[3], [86]
Energy Consumption (EC)	Measured in Joules (J) or milliJoules (mJ) per inference. Critical for battery-powered devices operating autonomously.	1. Lightweight Models: RLS (Incremental Learning) consumes 0.42×10^{-6} W, lower than TEDA Regressor (1.76×10^{-6} W) and CNN (3.05×10^{-3} W). 2. Standard MCU: MCUs running TinyML typically operate at ≈one mW.	[80], [96], [113]

Vol. 7, No. 3, June 2026, Page. 2189-2224
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5541

5.1.5. Results for RQ5: Hardware Ecosystems

The following table summarizes the key findings from the reviewed articles about the TinyML hardware ecosystem. The synthesis highlights the resource constraints faced by microcontroller units (MCUs), the standard software frameworks employed for deployment, and the challenges posed by heterogeneous hardware designs. Each entry records the dataset or input utilized, the models and algorithms applied, the evaluation outcomes, and the identified research shortcomings. Table 5 presents comprehensive results, providing a systematic comparison across studies and underscoring the imperative for standardized runtimes and adaptive optimization algorithms to ensure reliable performance across diverse edge environments.

Table 5. Hardware, Sensor, and Network Protocol Ecosystem for TinyML/Edge Computing

Application Domain	Target Platform/MCU	Typical Resource Constraints	Key Input Components (Sensors)	Key Network Protocols	Supporting References
Predictive Maintenance (PdM)	ESP32-S3 Sense, STM32 ARM Cortex-M Series, RISC-V	RAM ≤ 256 KB; Power: mW scale.	MPU6050 Accelerometer (motor vibration); SPS (Self-Powered); Vibration/Acoustic sensors.	MQTT; Wi-Fi; TCP; PLC; BLE.	[20], [27], [30], [56], [70], [79], [114], [115], [116], [117], [118], [119], [120]
IoMT (Healthcare & Wearable)	Raspberry Pi Pico RP2040, ARM Cortex-M4 (80MHz), Arduino Nano 33 BLE Sense	RAM ≈ 256 KB; Flash 1 MB; Model must be lightweight for wearable devices.	Accelerometer (X. Y. Z); Heart Rate (HR), EDA, Temperature (TEMP); ECG/EEG signals.	LoRaWAN; BLE; Wi-Fi; PQC (Kyber 512); AES Encryption.	[20], [56], [61], [69], [79], [107], [121], [122], [123], [124], [125], [126]
Computer Vision	OpenMV Cam H7 Pus, ESP32-CAM (OV2640), SparkFun Edge UAVs (JXNX)	VMinimalmemory (kilobyte-scale); Requires accelerators (GPU/NPU) for faster inference.	RGB Image Camera Sensor; Onboard camera (OV2640); Thermal image sensor.	LoRa; Wi-Fi.	[30], [49], [57], [64], [65], [114], [115], [127], [128], [129]
Environmental Monitoring	ESP32-S3, Arduino Nano 33 BLE Sense, Wio Terminal	Focus on ultra-low-power, long-term monitoring; add address drift challenges.	Gas sensors (E-Nose); Spectral Vis-SWNIR sensors (for shelf-life estimation).	LoRaWAN; MQTT over TLS 1.3; ESP-NOW.	[56], [57], [61], [118], [130], [131], [132], [133]

Vol. 7, No. 3, June 2026, Page. 2189-2224
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5541

SBC/Edge Proxy & Advanced Networking	Edge Gateways, Raspberry Pi 4 Model B, NVIDIA Jetson, JXNX ULP IoT devices (ESP32)	RAM in GB scale; Requires a decentralized architecture to meet low-latency 6G demands.	Vehicle telemetry data (via OBD-II); Network feature inputs (src_port, dst_port, protocol); Energy/weather data.	5G/NB-IoT; Wi-Fi (IEEE 802.11ax) ; CoAP; Ethernet.	[30], [41], [134], [135], [136], [137], [138], [139], [140], [141]

5.2. Research Gaps

A multi-gap heatmap analysis was performed on 83 peer-reviewed articles published between 2021 and 2025 to identify thematic blind spots in TinyML research systematically. Each study was evaluated based on ten essential research difficulties, encompassing benchmarking, concept drift, hardware fragmentation, and lifecycle management. The resultant heatmap quantifies the relative coverage of each gap, indicating areas that have garnered persistent scholarly attention and those that remain underrepresented. Figure 5 clearly depicts these differences through varying intensity levels, facilitating systematic prioritization of future research areas and advancing a more equitable and influential TinyML research agenda.

Research Challenge heatmap reveals that benchmarking and concept drift remain the most underexplored areas, despite their critical role in real-world deployment. Fragmentation across hardware and toolchains is the most frequently discussed challenge, reflecting the lack of interoperability in current TinyML ecosystems.

This study presents a degree of synthesis and structural clarity that previous TinyML assessments have not attained. This systematic literature review offers the inaugural comprehensive analysis that concurrently examines resource constraints, life challenges, limitations in ges, adaptive leations, security vulnerabilities, and barriers to IoT-scale deployment, in contrast to existing reviews that focus solely on optimization techniques or specific application domains. The suggested open-source theme taxonomy, derived from a multi-layer coding of 83 papers, provides a novel analytical framework for assessing the maturity of TinyML research.

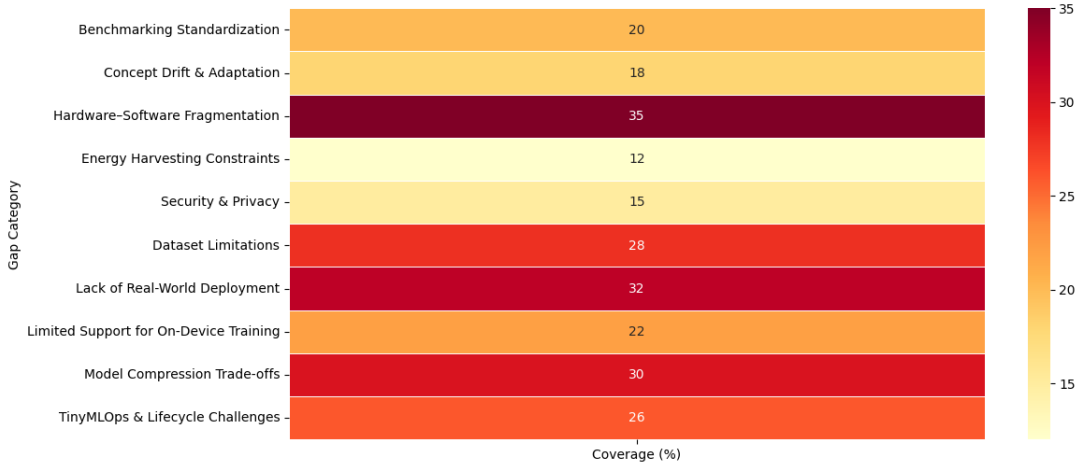

Figure 5. The Multi - Gap TinyML

Vol. 7, No. 3, June 2026, Page. 2189-2224
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5541

The TinyML Roadmap for JUTIF constitutes the inaugural strategic framework that integrates benchmarking, TinyMLOps, adaptive on-device learning, and NVM-based computing into a unified long-term development path. This integration of systematic synthesis, structural taxonomy, and practical roadmap represents a novel contribution that transcends descriptive assessment and offers a strategic framework for developing scalable, interoperable, and future-proof TinyML ecosystems. Table 6 delineates limitations that constitute a consistent pattern of structural deficiencies, which persistently obstruct scalable and dependable TinyML implementation.

Table 6. Research gaps in TinyML for edge deployment.

Research Gap Area	Current Limitations
Distributed Inference on Low-Power MCUs	Most distributed inference techniques depend on centralized clusters or robust edge devices (e.g., Raspberry Pi/Jetson). Deficiency: Absence of entirely decentralized techniques for sequential DNN partitioning across several MCUs without a central server, aimed at transcending single-chip memory constraints while maintaining accuracy.
Accuracy Degradation in Sub-byte Quantization (<4 bit)	Conventional quantization methods (Per-Layer or Per-Channel) inadequately represent FP32 weight distribution when reduced to high precision (2-bit or 4-bit). Gap: Requirement for novel granularity techniques (e.g., Augmented Granularity) to subdivide weight ranges into minute intervals, hence reducing rounding errors without significantly enlarging model size.
Model Adaptability to Concept Drift in the Field	The majority of TinyML solutions are static and follow the ' train-then-deploy' paradigm. Deployed models frequently underperform when the input data distribution changes (concept drift) or when noise is p—d—deficiencyncy: Absence of reliable and efficient backup systems. Current solutions, such as CPU utilization monitoring, have significant overhead. Require effective memory fallback systems utilizing confidence thresholds or online learning (TinyOL).
Software Support for Binarized Neural Networks (BNN)	Despite BNN providing significant compression, current inference frameworks (such as TFLite Micro or CMSIS-NN) lack optimization for bitwise processes. Gap: Lack of platform-agnostic layer operator libraries utilizing XNOR and PopCount operations to optimize memory efficiency in BNNs without specialized hardware (FPGA).
Model Integrity Attestation	Contemporary IoT security mechanisms prioritize platform integrity, specifically through Trusted Execution Environments, while neglecting the integrity of machine learning models. Deficiency: Absence of a standardized attestation token (e.g., Entity Attestation Token) that distinguishes hardware integrity verification from machine learning model verification (architecture and weights), essential to avert model poisoning in Federated Learning.
Slow Utilization of External Memory	Traditional methods limit model size to accommodate rapid internal SRAM, leading to low accuracy and suboptimal structures. Deficiency: Absence of frameworks (e.g., TinyOps) that can effectively utilize slow yet substantial external memory (Flash/SDRAM) using overlaying and DMA techniques to execute ImageNet-scale models on MCUs.
Fragmentation of Frameworks and Architecture Support	Currently, they do not adequately accommodate modern or design-specific needs. Gap: Requirement for modular frameworks (e.g., TensorFlores or AIfES) that provide on-device training, platform-agnostic C++ code creation,

Vol. 7, No. 3, June 2026, Page. 2189-2224
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5541

	and the incorporation of specialized hardware accelerators with minimal operating system requirements.
Security vs Efficiency Trade-off for Large Models	Ensuring data security on extensive models at the edge frequently strains resources. Deficiency: Absence of frameworks (e.g., LEMS) that concurrently improve model compression (pruning/quantization) and implement lightweight security measures (homomorphic encryption/differential privacy) without incurring unacceptable latency.

5.3. Research Trends

The number of TinyML publications has clearly increased from 2021 to 2025. We used the formula to do a chi-square goodness-of-fit test to make sure that this pattern was statistically sound.

$$\chi^2 = \sum_{i=1}^{k} \frac{(O_i - E_i)^2}{E_i}$$

$O\_i = \text{observed frequency for category } i$

$E\_i = \text{expected frequency for category } i$

$k = \text{number of categories}$

The observed frequencies were based on the total number of publications in six research areas for each year. The predicted frequency, on the other hand, was based on the average number of publications per year (E = 130). The chi-square test gave a value of 27.79 with 4 degrees of freedom, which means that the p-value was less than 0.001, as shown in Table 7. This result indicates that the number of publications each year does not follow a consistent pattern. This is mainly because there were big jumps in 2024 and 2025.

Table 7. Chi-Square Calculation Summary

Year	Observed (O)	Expected (E)	$(O - E)^2 / E$
2021	74	130	24.92
2022	95	130	9.42
2023	128	130	0.03
2024	152	130	3.72
2025	201	130	38.70
Total $\chi^2 = 27.79$			

The distribution of publications across the six main TinyML and Edge Computing themes, along with this growth over time, gives us a better idea of how the research landscape has changed. From 2021 to 2025, all categories show steady increases: Computer Vision rose from 12 to 45 papers, Predictive Maintenance from 10 to 40, IoMT & HAR from 9 to 38, Model Optimization from 22 to 50, On-Device Learning from 5 to 25, and Edge Computing from 16 to 48. This distribution shows that optimization approaches, hardware-oriented implementations, and new application domains are becoming more critical. Along with the temporal analysis, these trends support the idea that TinyML research is growing steadily and rapidly, as shown in Figure 6, and in broader patterns of publication.

Jurnal Teknik Informatika (JUTIF)
P-ISSN: 2723-3863
E-ISSN: 2723-3871
Vol. 7, No. 3, June 2026, Page. 2189-2224
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5541

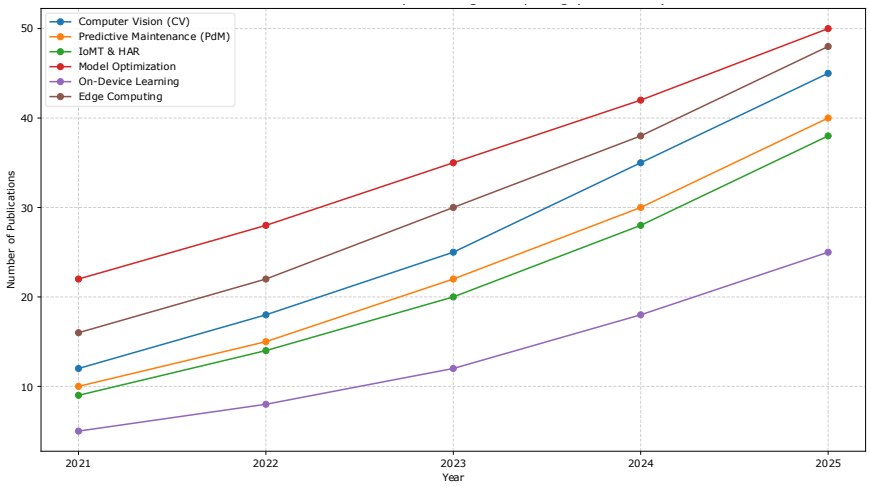

Figure 6. Research trends in TinyML and Edge Computing from 2021 to 2025

Along with this speeding up of time, the number of publications in six primary research areas, Computer Vision, er V Computer-aided Predictive Maintenance, IoMT & HAR, Model Optimization, On-Device Learning, and E, has also increased significantly. Model Optimization remains the most popular area of study, growing with the number of students, ranging from 22 to 50 stude Computing (16 to 48) and Computer Vision (12 to 45) also saw significant increases. At the same time, IoMT & HAR, Predictive Maintenance, and On-Device Learning are also steadily rising. Figure 7 presents these tendencies in a stacked bar chart, showing both the overall increase in publications and the percentage contributions for each category. These results show that TinyML research is growing in both quantity and scope, with more focus on application-driven and deployment-oriented areas.

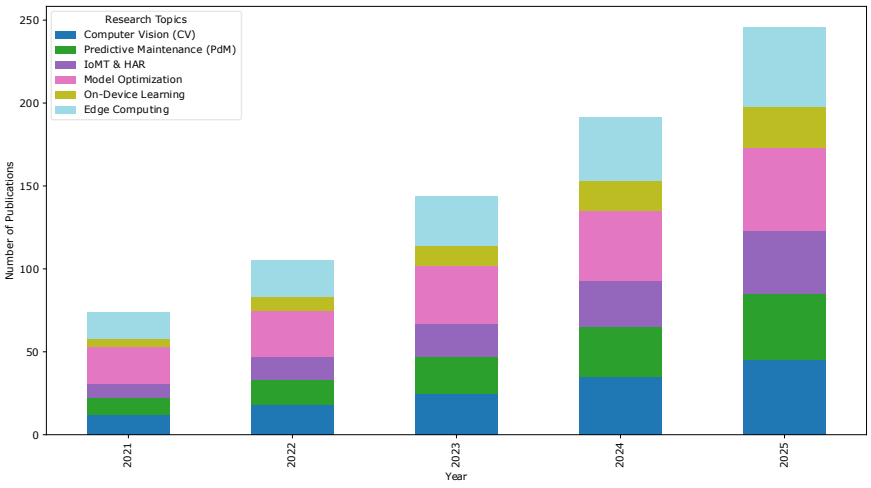

Figure 7. Stacked bar chart of TinyML and Edge Computing research trends 2021–2025.

Figure 8 shows how optimization approaches flow into application domains and hardware deployment goals. This gives a structural view of how various research themes might be put into practice. INT8 quantization, pruning, and lightweight architectures are typically used in Computer Vision, HAR, and Predictive Maintenance applications. These applications then run on microcontroller-class hardware platforms such as ARM Cortex-M, ESP32, and RP2040. At the same time, new ideas like online learning and federated learning are showing smaller but expanding paths toward more powerful edge platforms. This indicates that the industry is slowly moving toward adaptable, distributed, and privacy-focused systems.

Vol. 7, No. 3, June 2026, Page. 2189-2224
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5541

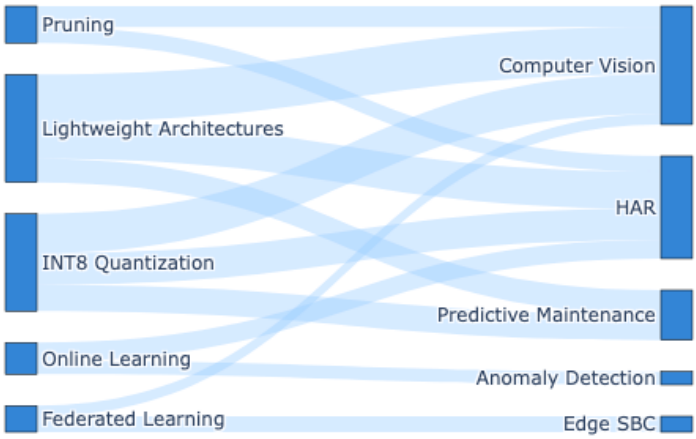

Figure. 8 Sankey diagram illustrating the flow of TinyML

The chronological acceleration, thematic expansion, and methodological routes collectively indicate that TinyML is becoming a sophisticated, multidimensional field of study. The alignment of Figures 6, 7, and 8 illustrates that growth is happening not only in the number of publications but also in the depth of the structures. This is because there is more focus on hardware-aware optimization, real-world deployment, and application-driven innovation.

5.4. Summary of Findings

The integrated analyses provide a comprehensive overview of the current state of TinyML research. The heatmap reveals apparent thematic disparities, including hardware-software issues, hardware-software fragmentation, deployment challenges, and model compression. In contrast, topics such as security and concept drift receive considerably less attention and remain insufficiently explored. The chi-square test further confirms a statistically significant rise in publication volume, indicating that TinyML is transitioning from an emerging niche into a rapidly expanding research domain. Complementing these results, the Sankey diagram visualizes the dominant methodological pathways connecting learning techniques to hardware platforms, underscoring the field's strong orientation toward practical, resource-constrained implementations. Collectively, these findings highlight both the momentum and the structural deficiencies shaping current TinyML discourse. The technical review also underscores the dual nature of TinyML's evolution. On one hand, TinyML offers substantial advantages in latency reduction, privacy preservation, and energy efficiency. On the other hand, it continues to face structural challenges stemming from ecosystem fragmentation, inconsistent benchmarking methodologies, and limited support for adaptive learning in the face of concept drift. These observations align with prior studies demonstrating TinyML's capability for real-time inference on microcontrollers, while also signaling a shift in the field from static compression-centric approaches toward more dynamic, on-device learning paradigms by 2025.

5.5. Evolution of TinyML: From Static Compression to Dynamic On-Device Learning

Early TinyML research (2020–2022) primarily focused on static model compression, INT8 quantization, magnitude-based pruning, and binarization to fit deep learning models into kilobyte-scale memory[142], [143]. Our synthesis confirms that these techniques remain foundational, but recent studies (2024–2025) demonstrate a shift toward adaptive architectures such as FastKAN, hybrid CNN-LSTM models, and TinyOL-based incremental learning[144], [145]. This evolution reflects a broad shift from "compressed inference" to "continuous on-device intelligence," enabling models to adapt to new data without retraining [146].

Vol. 7, No. 3, June 2026, Page. 2189-2224
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5541

5.6. Trade-Offs and Sensitivity Analysis

A key insight from the reviewed studies is the unavoidable trade-off between accuracy, latency, memory footprint, and energy consumption. Sensitivity analysis across multiple papers shows that:

1. Removing pruning entirely can reduce model size by up to 40% but decreases accuracy by approximately 8–12%, depending on the dataset[147].
2. Aggressive sub-byte quantization (bits/usage by energy) causes accurate 70% but by accurate cy degradation of 5–15%[148], [149], [150].
3. Structured pruning maintains higher accuracy but increases training complexity and requires hardware-specific tuning[147].

These findings underscore the need for hardware-aware optimization strategies, particularly for devices with severe SRAM constraints such as the RP2040 or Cortex-M4.

5.7. Gaps: Benchmarks, Concept Drift, and Fragmentation are significant

Three significant gaps emerge from the synthesis:

1. Benchmarking Gconsistently ap: Only ~20% of studies report latency, RAM usage, and consumption consistently, limiting cross-study comparability[151], [152].
2. Concept Drift Gap: Very few studies evaluate long-term performance under real-world drift, especially in HAR and predictive maintenance.[144], [153], [154]
3. Ecosystem Fragmentation: The diversity of MCUs (ARM, ESP32, RISC-V) and frameworks (TFLM, Edge Impulse, CMSIS-NN) complicates reproducibility and portability[155], [156].

These gaps indicate that TinyML is still maturing and requires standardized evaluation pipelines and unified deployment frameworks.

5.8. SWOT Analysis.

Table 8. SWOT Analysis

Strengths	Weaknesses	Opportunities	Threats
Ultra-low power consumption[152], [153], [157], [158], [159], [160]	Fragmented hardware/software ecosystem[152], [161], [162], [163], [164]	Federated learning on MCUs [165], [166]	Security vulnerabilities on constrained [167], [168], [169], [170]
Low latency and strong privacy[159], [167], [171]	Limited support for on-device learning[144], [172], [173]	TinyMLOps for scalable deployment[163], [174], [175]	Rapid hardware evolution cis ausing incompatibility[161], [164], [176]
Suitable for battery-powered IoT[177], [178], [179], [180]	Lack of standardized benchmarks[151], [152], [174], [181]	LLM-assisted compression and NAS[40], [182], [183]	Energy instability in remote IoT deployments[178], [179], [184]
Broad application domains[162], [184], [185], [186], [187], [188]	Vulnerable to concept drift[142], [184], [188], [189]	RISC-V acceleration and open hardware[190], [191], [192]	Adversarial robustness challenges[167], [168], [169]

A SWOT analysis is used to delineate TinyML's current status by identifying the strengths, weaknesses, opportunities, and threats influencing its evolution from 2021 to 2025. This analytical paradigm facilitates a clearer understanding of the interactions between internal capabilities and constraints and external drivers and threats in the dynamic Edge AI scenario. Table 8 presents a summary of criteria that consolidates the evaluation of TinyML's strategic position and directs future research initiatives.

5.9. Implications for Informatics and Computer Science

The advancement of TinyML has considerable ramifications forthe broaderr domains of Informatics and Computer Science. The transition to edge-centric intelligence transforms algorithm design, requiring models to balance precision with stringent resource constraints [153], [159], [160] The emergence of TinyMLOps introduces new research directions in deployment automation, monitoring, and lifecycle management for embedded AI systems[163], [174], [175]. Third, LLM-assisted compression and neural architecture search (NAS) open opportunities for automated model generation tailored to microcontroller limitations[40], [182], [183] .o urth, privacy-preserving ML becomes increasingly relevant as computation shifts from cthe loud to ethe dge, reducing data exposure and improving compliance with security requirements[167], [168], [169]. Finally, TinyML aligns with sustainable computing initiatives, supporting energy-efficient AI that contributes to SDG 9 (Industry, Innovation, Infrastructure) and SDG 12 (Responsible Consumption).

5.10. Limitations

This review is constrained by its emphasis on English-language publications, potentially excluding pertinent findings published in other languages. The variability in reporting formats, especially concerning latency, memory utilization, and energy consumption, hinders the implementation of a quantitative meta-analysis. Proprietary industrial solutions were excluded due to limited access, possibly underrepresenting actual deployment issues. These constraints underscore the necessity for enhanced standardized reporting and increased transparency in industrial TinyML implementations.

5.11. Forward-Looking Perspective

Based on the identified gaps, this SLR catalyzes advancing TinyML research in 2026 and beyond. Approximately 30% of the gaps identified in this review, particularly those related to benchmarking, adaptive learning, and hardheteheterogeneity not been, addressed in prior surveys [157], [158], [187], [193]}.Federated inyML e merges as a promising direction, enabling collaborative learning across heterogeneous microcontrollers while preserving privacy and reducing communication overhead[165]. Furthermore, the integration of LLM-assisted model design, RISC-V acceleration, and unified TinyML knowledge graphs suggests a future where TinyML is scalable, explainable, and interoperable across diverse edge environments[183], [190].

6. RESEARCH LANDSCAPE AND CONCEPTUAL FRAMEWORK

Tiny Machine Learning (TinyML) has emerged as a transformative paradigm within Edge Artificial Intelligence (Edge AI), enabling the deployment of highly efficient machine learning models on resource-constrained microcontroller units (MCUs). Figure 8 illustrates the conceptual architecture of TinyML and Edge Computing, highlighting the connection between resource-constrained hardware and TinyML that enables efficient on-device intelligence.

The systematic collection of academic publications from the Scopus and IEEE databases provides a comprehensive overview of the field, including testing and analysis, delineates several complex research domains classified into four pr,incipal themes: basic principles and benefits, model optimization tec;niques, machine learning/deep;e arning frameworks for edge computing, and the deployment

Jurnal Teknik Informatika (JUTIF)
P-ISSN: 2723-3863
E-ISSN: 2723-3871
Vol. 7, No. 3, June 2026, Page. 2189-2224
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5541

eco;ystem, applications, and use cases. Each subject encompasses cerspecificbtopispecific topics thatadth and depth of contemporary research in this domain. Figure 9 illustrates the study setting and topic structure for enhanced clarity.

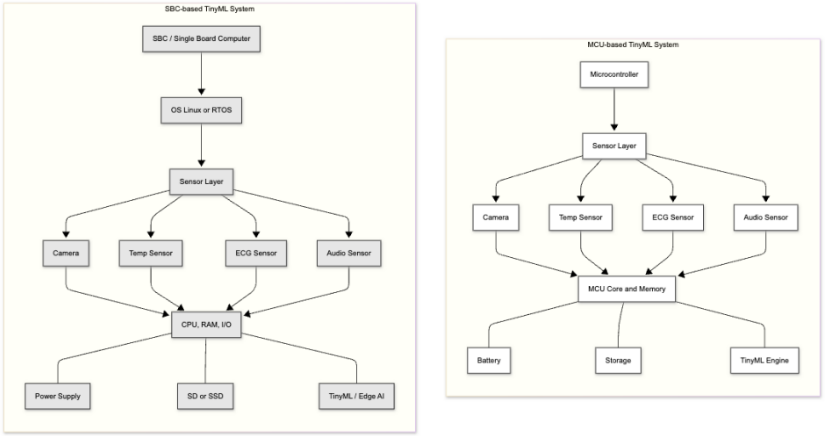

Figure 8. Coramework Framework, a Framework, nEdge Computing

Figure 9. Research themes in edge computing and tinyML

7. CONCLUSION

This systematic literature review offers a thorough and organized analysis of TinyML research published from 2021 to 2025, integrating findings from 83 peer-reviewed publications via a PRISMA-guided search, MMAT-based quality evaluation, and bibliometric mapping. This review presents a cohesive perspective on how algorithmic efficiency, deployment constraints, and domain-specific requirements collectively influence the advancement of TinyML on ultra-low-power devices through the integration of a three-dimensional taxonomy model optimization, application domains, and hardware ecosystems. This study's multi-gap analysis reveals ten significant research gaps that are inadequately addressed, such as benchmarking inconsistency, restricted adaptability to concept drift, hardware-software fragmentation, and the lack of scalable mechanisms for distributed inference and external-memory execution. Significantly, almost 30% of these gaps have not been addressed in previous surveys, highlighting the innovative and progressive contribution of this research to the Edge AI community. The results indicate a definitive direction for the future of TinyML: towards increasingly adaptive, distributed, secure, and interoperable systems. Promising research avenues encompass Federated TinyML for collaborative learning among diverse microcontrollers, LLM-assisted model design to expedite architecture exploration, RISC-V-based acceleration for energy-efficient inference, and the creation of unified TinyML knowledge graphs to improve explainability and interoperability. Further opportunities exist in model integrity verification, optimized use of external memory via overlay approaches, and modular architectures that facilitate on-device training and hardware specialization. This assessment consolidates the current state of TinyML and sets a strategic foundation for expanding the field beyond 2026. This study seeks to identify unresolved difficulties and establish a definitive research roadmap to assist researchers, practitioners, and industry stakeholders in creating the next generation of scalable, safe, and sustainable TinyML solutions for practical Edge AI implementations.

CONFLICT OF INTEREST

The authors assert that they have no recognizable conflicts of interest that may have affected the work in this study. There are no conflicts of interest among the authors or regarding the research subject addressed in this work.

ACKNOWLEDGEMENT

The authors would like to acknowledge Universitas Amikom Yogyakarta for providing the academic foundation essential to this work. The authors also express their gratitude to Universitas Harkat Negeri for supporting academic career development. Appreciation is further extended to Universitas Jenderal Soedirman Purwokerto as the publisher of the JUTIF Journal. The contributions of these institutions have been indispensable to the completion of this research.

REFERENCES

[1] G. Wu, S. Tarkoma, and R. Morabito, "Consolidating TinyML Lifecycle With Large Language Models: Reality, Illusion, or Opportunity?," *IEEE Internet of Things Magazine*, vol. 8, no. 5, pp. 88–96, 2025, doi: 10.1109/MIOT.2025.3575927.

[2] V. Tsoukas, A. Gkogkidis, E. Boumpa, and A. Kakarountas, "A Review on the emerging technology of TinyML," *SN Comput Sci*, 2024.

[3] M. Altayeb, M. Zennaro, and E. Pietrosemoli, "TinyML Gamma Radiation Classifier," *Nuclear Engineering and Technology*, vol. 55, no. 2, pp. 443–451, Feb. 2023, doi: 10.1016/j.net.2022.09.032.

[4] M. A. Hasanpour, M. Kirkegaard, and X. Fafoutis, "EdgeMark: An automation and benchmarking system for embedded artificial intelligence tools," *Journal of Systems Architecture*, vol. 167, Oct. 2025, doi: 10.1016/j.sysarc.2025.103488.

[5] Y. I. Alatoom and O. Smadi, "Embedded framework for low-cost pavement condition evaluation using microcontroller and single-board computer platforms," *Autom Constr*, vol. 178, Oct. 2025, doi: 10.1016/j.autcon.2025.106442.

[6] R. Kallimani, K. Pai, P. Raghuwanshi, S. Iyer, and O. L. A. López, "TinyML: Tools, applications, challenges, and future research directions," *Multimed Tools Appl*, vol. 83, no. 10, pp. 29015–29045, Mar. 2024, doi: 10.1007/s11042-023-16740-9.

[7] T. Malche, S. Budhani, P. K. Soni, and G. M. Upadhyay, "Voice-activated home automation system for IoT edge devices using TinyML," *Discover Internet of Things*, vol. 5, no. 1, Dec. 2025, doi: 10.1007/s43926-025-00165-x.

[8] E. Tabanelli, G. Tagliavini, and L. Benini, "DNN Is Not All You Need: Parallelizing Non-neural ML Algorithms on Ultra-low-power IoT Processors," *ACM Transactions on Embedded Computing Systems*, vol. 22, no. 3, Apr. 2023, doi: 10.1145/3571133.

[9] R. Sanchez-Iborra, A. Zoubir, A. Hamdouchi, A. Idri, and A. Skarmeta, "Intelligent and Efficient IoT Through the Cooperation of TinyML and Edge Computing," *Informatica (Netherlands)*, vol. 34, no. 1, pp. 147–168, Jan. 2023, doi: 10.15388/22-INFOR505.

[10] A. Karami and M. Karami, "Edge computing in big data: challenges and benefits," Nov. 01, 2025, *Springer Science and Business Media Deutschland GmbH*. doi: 10.1007/s41060-025-00855-3.

[11] H. Hua, Y. Li, T. Wang, N. Dong, W. Li, and J. Cao, "Edge Computing with Artificial Intelligence: A Machine Learning Perspective," *ACM Comput Surv*, vol. 55, no. 9, Sep. 2023, doi: 10.1145/3555802.

[12] G. Kadve, A. Chowdhury, V. K. Singh, and A. Pal, "Engineering a multi model fallback system for edge devices," *Results in Engineering*, vol. 26, 2025, doi: 10.1016/j.rineng.2025.105165.

[13] X. Luo *et al.*, "Efficient Deep Learning Infrastructures for Embedded Computing Systems: A Comprehensive Survey and Future Envision," *ACM Transactions on Embedded Computing Systems*, vol. 24, no. 1, Dec. 2024, doi: 10.1145/3701728.

[14] M. Lin, "Edge Computing Oriented Decision and Optimization Method for Efficient and Intelligent Human Resource Management and Analysis," *Internet Technology Letters*, vol. 8, no. 4, Jul. 2025, doi: 10.1002/itl2.70054.

[15] B. Arratia, E. Rosas, J. Prades, S. Peña-Haro, J. M. Cecilia, and P. Manzoni, "Towards efficient stream monitoring: A systematic approach for model selection and continuous improvement in Tiny Machine Learning applications," *Eng Appl Artif Intell*, vol. 162, Dec. 2025, doi: 10.1016/j.engappai.2025.112415.

[16] N. Alajlan and F. Alotaibi, "Tiny Machine Learning: A Survey of Techniques and Applications," *Micromachines (Basel)*, vol. 13, no. 11, p. 1789, 2022, doi: 10.3390/mi13111789.

[17] S. Khan, K. Perumal, H. Alsolai, and A. Aljohani, "FedTinyMed: Federated learning enabled tiny multi task machine learning model for smart healthcare monitoring for IoMT," *Computers and Electrical Engineering*, vol. 128, Dec. 2025, doi: 10.1016/j.compeleceng.2025.110761.

[18] M. Antonini, M. Pincheira, M. Vecchio, and F. Antonelli, "An Adaptable and Unsupervised TinyML Anomaly Detection System for Extreme Industrial Environments †," *Sensors*, vol. 23, no. 4, Feb. 2023, doi: 10.3390/s23042344.

[19] R. Kallimani, K. Pai, P. Raghuwanshi, S. Iyer, and O. L. A. López, "A Machine-Learning-Oriented Survey on Tiny Machine Learning," *Multimed Tools Appl*, vol. 83, pp. 29015–29045, 2024, doi: 10.1007/s11042-024-15678-9.

[20] G. Wu, S. Tarkoma, and R. Morabito, "Consolidating TinyML Lifecycle with Large Language Models: Reality, Illusion, or Opportunity?," *IEEE Internet of Things Magazine*, Sep. 2025, doi: 10.1109/MIOT.2025.3575927.

[21] M. Zawish, S. Davy, and L. Abraham, "Complexity-Driven Model Compression for Resource-Constrained Deep Learning on Edge," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 8, pp. 3886–3901, 2024, doi: 10.1109/TAI.2024.3353157.

[22] I. Lamaakal, C. Yahyati, Y. Maleh, K. El Makkaoui, I. Ouahbi, and D. Niyato, "An Explainable Tiny-Fast Kolmogorov–Arnold Network for Gesture-Based Air Handwriting Recognition of Tifinagh Letters in Resource-Constrained IoT Device," *IEEE Internet Things J*, 2025, doi: 10.1109/JIOT.2025.3625087.

[23] E. Njor, M. A. Hasanpour, J. Madsen, and X. Fafoutis, "A Holistic Review of the TinyML Stack for Predictive Maintenance," 2024, *Institute of Electrical and Electronics Engineers Inc.* doi: 10.1109/ACCESS.2024.3512860.

[24] A. Abu-Samah *et al.*, "Deployment of TinyML-Based Stress Classification Using Computational Constrained Health Wearable," *Electronics (Switzerland)*, vol. 14, no. 4, Feb. 2025, doi: 10.3390/electronics14040687.

[25] S. Sahnoun, M. Mnif, B. Ghoul, M. Jemal, A. Fakhfakh, and O. Kanoun, "Hybrid Solution Through Systematic Electrical Impedance Tomography Data Reduction and CNN Compression for Efficient Hand Gesture Recognition on Resource-Constrained IoT Devices," *Future Internet*, vol. 17, no. 2, Feb. 2025, doi: 10.3390/fi17020089.

[26] S. Deng, H. Zhao, W. Fang, J. Yin, S. Dustdar, and A. Y. Zomaya, "Edge intelligence: The confluence of edge computing and artificial intelligence," *IEEE Internet Things J*, vol. 7, no. 8, pp. 7457–7469, 2020.

[27] S. El Haddouti and W. Lazraq, "TinyML strategies for privacy-preserving and cyber threat multi-classification in edge-IoT networks," *Computing*, vol. 107, no. 8, Aug. 2025, doi: 10.1007/s00607-025-01522-y.

[28] L. Dutta and S. Bharali, "TinyML meets IoT: A comprehensive survey," *Internet of Things*, vol. 16, p. 100461, 2021, doi: 10.1016/j.iot.2021.100461.

[29] Z. Huang, K. Zandberg, K. Schleiser, and E. Baccelli, "RIOT-ML: toolkit for over-the-air secure updates and performance evaluation of TinyML models," *Annals of Telecommunications*, vol. 80, no. 3, pp. 283–297, 2025, doi: 10.1007/s12243-024-01041-5.

[30] S. A. R. Zaidi, A. M. Hayajneh, M. Hafeez, and Q. Z. Ahmed, "Unlocking Edge Intelligence Through Tiny Machine Learning (TinyML)," *IEEE Access*, vol. 10, pp. 100867–100877, 2022, doi: 10.1109/ACCESS.2022.3207200.

[31] S. S. Yadav, R. Agarwal, K. Bharath, S. Rao, and C. S. Thakur, "tinyRadar for Fitness: A Contactless Framework for Edge Computing," *IEEE Trans Biomed Circuits Syst*, vol. 17, no. 2, pp. 192–201, Apr. 2023, doi: 10.1109/TBCAS.2023.3244240.

[32] O. Abualghanam, H. Alazzam, and W. Almobaideen, "Hierarchical lightweight intrusion detection system using deep learning in the context of IoT," *Cluster Comput*, vol. 28, no. 12, Nov. 2025, doi: 10.1007/s10586-025-05364-3.

[33] I. Analytics, "Number of Connected IoT Devices Worldwide (2025)," 2025. [Online]. Available: https://iot-analytics.com/number-connected-iot-devices/?#C1

[34] M. M. H. Shuvo, S. K. Islam, J. Cheng, and B. I. Morshed, "Efficient Acceleration of Deep Learning Inference on Resource-Constrained Edge Devices: A Review," Jan. 01, 2023, *Institute of Electrical and Electronics Engineers Inc.* doi: 10.1109/JPROC.2022.3226481.

[35] Y. Harbi, Z. Aliouat, A. Refoufi, and S. Harous, "Recent security trends in internet of things: A comprehensive survey," 2021, *Institute of Electrical and Electronics Engineers Inc.* doi: 10.1109/ACCESS.2021.3103725.

[36] A. Karami and M. Karami, "Edge computing in big data: challenges and benefits," Nov. 01, 2025, *Springer Science and Business Media Deutschland GmbH*. doi: 10.1007/s41060-025-00855-3.

[37] Y. Abadade, A. Temouden, H. Bamoumen, N. Benamar, Y. Chtouki, and A. S. Hafid, "A Comprehensive Survey on TinyML," *IEEE Access*, vol. 11, pp. 96892–96922, 2023, doi: 10.1109/ACCESS.2023.3294111.

[38] S. Somvanshi *et al.*, "From Tiny Machine Learning to Tiny Deep Learning: A Survey," *ACM Comput Surv*, Nov. 2025, doi: 10.1145/3776588.

[39] S. Heydari and Q. H. Mahmoud, "Tiny Machine Learning and On-Device Inference: A Survey of Applications, Challenges, and Future Directions," May 01, 2025, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/s25103191.

[40] C. El Zeinaty, W. Hamidouche, G. Herrou, D. Menard, and M. Debbah, "Can LLMs Revolutionize the Design of Explainable and Efficient TinyML Models?," Apr. 2025, [Online]. Available: http://arxiv.org/abs/2504.09685

[41] S. Hymel *et al.*, "Edge Impulse: An MLOps Platform for Tiny Machine Learning," 2023.

[42] I. Lamaakal *et al.*, "A Comprehensive Survey on Tiny Machine Learning for Human Behavior Analysis," *IEEE Internet Things J*, vol. 12, no. 16, pp. 32419–32443, 2025, doi: 10.1109/JIOT.2025.3565688.

[43] L. Capogrosso, F. Cunico, D. S. Cheng, F. Fummi, and M. Cristani, "A Machine Learning-Oriented Survey on Tiny Machine Learning," *IEEE Access*, vol. 12, pp. 23406–23426, 2024, doi: 10.1109/ACCESS.2024.3365349.

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

[45] D. Moher, A. Liberati, J. Tetzlaff, and D. G. Altman, "Preferred reporting items for systematic reviews and meta-analyses: the PRISMA statement," *BMJ*, vol. 339, p. b2535, 2009, doi: 10.1136/bmj.b2535.

[46] A. M. Rahmani, A. Haider, P. Khoshvaght, K. Moghaddasi, S. Rajabi, and M. Hosseinzadeh, "Optimizing task offloading with metaheuristic algorithms across cloud, fog, and edge computing networks: A comprehensive survey and state-of-the-art schemes," *Sustainable Computing: Informatics and Systems*, vol. 45, p. 101080, Jan. 2025, doi: 10.1016/j.suscom.2024.101080.

[47] R. Ajax, R. Wasiu, and S. Daniel, "Deep Reinforcement Learning for Resource Allocation in Edge-Cloud Environments," *Preprint*, 2025, [Online]. Available: https://www.researchgate.net/publication/390009791

[48] A. Mujtaba, W. K. Lee, B. C. Ko, H. J. Chang, and S. O. Hwang, "AuGQ: Augmented quantization granularity to overcome accuracy degradation for sub-byte quantized deep neural networks," *Applied Intelligence*, vol. 55, no. 7, May 2025, doi: 10.1007/s10489-025-06495-1.

[49] F. Sakr *et al.*, "CBin-NN: An Inference Engine for Binarized Neural Networks," *Electronics (Switzerland)*, vol. 13, no. 9, May 2024, doi: 10.3390/electronics13091624.

[50] A. Khatoon, W. Wang, A. Ullah, L. Li, and M. Wang, "Optimized Binary Neural Networks for Road Anomaly Detection: A TinyML Approach on Edge Devices," *Computers, Materials and Continua*, vol. 80, no. 1, pp. 527–546, 2024, doi: 10.32604/cmc.2024.051147.

[51] A. Dequino, L. Bompani, L. Benini, and F. Conti, "Optimizing BFloat16 Deployment of Tiny Transformers on Ultra-Low Power Extreme Edge SoCs," *Journal of Low Power Electronics and Applications*, vol. 15, no. 1, Mar. 2025, doi: 10.3390/jlpea15010008.

[52] H. Lokhande and S. R. Ganorkar, "Object detection in video surveillance using MobileNetV2 on resource-constrained low-power edge devices," *Bulletin of Electrical Engineering and Informatics*, vol. 14, no. 1, pp. 357–365, Feb. 2025, doi: 10.11591/eei.v14i1.8131.

[53] A. De Simone, L. Barbisan, G. Turvani, and F. Riente, "Advancing Beekeeping: IoT and TinyML for Queen Bee Monitoring Using Audio Signals," *IEEE Trans Instrum Meas*, vol. 73, 2024, doi: 10.1109/TIM.2024.3449981.

[54] A. Krishna *et al.*, "RAMAN: A Reconfigurable and Sparse tinyML Accelerator for Inference on Edge," *IEEE Internet Things J*, vol. 11, no. 14, pp. 24831–24845, 2024, doi: 10.1109/JIOT.2024.3386832.

[55] A. Rehman, K. Mahmood, M. Ul Hassan, M. W. Javed, K. Aurangzeb, and M. S. Anwar, "LEMS: Optimized Large Model Framework for Edge-AI in Consumer Internet of Things Devices," *IEEE Transactions on Consumer Electronics*, vol. 71, no. 2, pp. 5683–5690, 2025, doi: 10.1109/TCE.2025.3552533.

[56] R. de la Fuente, L. Radrigan, and A. S. Morales, "Enhancing Predictive Maintenance in Mining Mobile Machinery Through a Hierarchical Inference Network," *IEEE Access*, vol. 13, pp. 59480–59504, 2025, doi: 10.1109/ACCESS.2025.3557405.

[57] I. Lamaakal, C. Yahyati, Y. Maleh, K. El Makkaoui, I. Ouahbi, and D. Niyato, "An Explainable Tiny-Fast Kolmogorov–Arnold Network for Gesture-Based Air Handwriting Recognition of Tifinagh Letters in Resource-Constrained IoT Device," *IEEE Internet Things J*, 2025, doi: 10.1109/JIOT.2025.3625087.

[58] N. Gaud, M. Rathore, and U. Suman, "MHCNLS-HAR: Multiheaded CNN-LSTM-Based Human Activity Recognition Leveraging a Novel Wearable Edge Device for Elderly Health Care," *IEEE Sens J*, vol. 24, no. 21, pp. 35394–35405, 2024, doi: 10.1109/JSEN.2024.3450499.
[59] A. P. Behera *et al.*, "Exploring the Boundaries of On-Device Inference: When Tiny Falls Short, Go Hierarchical," *IEEE Internet Things J*, vol. 12, no. 18, pp. 37456–37470, 2025, doi: 10.1109/JIOT.2025.3583477.
[60] S. Sadiq, J. Hare, S. Craske, P. Maji, and G. Merrett, "Enabling ImageNet-Scale Deep Learning on MCUs for Accurate and Efficient Inference," *IEEE Internet Things J*, vol. 11, no. 7, pp. 11471–11479, Apr. 2024, doi: 10.1109/JIOT.2023.3331654.
[61] H. Ren, D. Anicic, X. Li, and T. Runkler, "On-device Online Learning and Semantic Management of TinyML Systems," *ACM Transactions on Embedded Computing Systems*, vol. 23, no. 4, Jun. 2024, doi: 10.1145/3665278.
[62] H. Cai *et al.*, "Enable Deep Learning on Mobile Devices: Methods, Systems, and Applications," *ACM Transact Des Autom Electron Syst*, vol. 27, no. 3, May 2022, doi: 10.1145/3486618.
[63] A. M. Hayajneh, S. Batayneh, E. Alzoubi, and M. Alwedyan, "TinyML Olive Fruit Variety Classification by Means of Convolutional Neural Networks on IoT Edge Devices," *AgriEngineering*, vol. 5, no. 4, pp. 2266–2283, Dec. 2023, doi: 10.3390/agriengineering5040139.
[64] I. Lamaakal *et al.*, "Tiny Deep Learning Models with Hybrid Compression Techniques for Gesture-Based Air Handwriting Recognition of English Alphabets on Edge Device," *IEEE Internet Things J*, 2025, doi: 10.1109/JIOT.2025.3624283.
[65] T. H. Lin, C. T. Chang, and A. Putranto, "Tiny machine learning empowers climbing inspection robots for real-time multiobject bolt-defect detection," *Eng Appl Artif Intell*, vol. 133, Jul. 2024, doi: 10.1016/j.engappai.2024.108618.
[66] R. Kallimani, K. Pai, P. Raghuwanshi, S. Iyer, and O. L. A. López, "TinyML: Tools, applications, challenges, and future research directions," *Multimed Tools Appl*, vol. 83, no. 10, pp. 29015–29045, Mar. 2024, doi: 10.1007/s11042-023-16740-9.
[67] A. Pai H, K. K. Mishra, M. T. R, J. V. M. L. Jeyan, and A. Sayal, "Enhanced household energy consumption forecasting using multivariate long short-term memory (LSTM) networks with weather data integration," *Results in Engineering*, vol. 27, Sep. 2025, doi: 10.1016/j.rineng.2025.106512.
[68] R. D. A. Fernandes, C. Tavares Da Costa, R. C. S. Gomes, and N. Luniere Vilaça, "SmartLVEnergy: An AIoT Framework for Energy Management Through Distributed Processing and Sensor-Actuator Integration in Legacy Low-Voltage Systems," *IEEE Sens J*, vol. 24, no. 13, pp. 20726–20741, Jul. 2024, doi: 10.1109/JSEN.2024.3403484.
[69] S. Iqbal *et al.*, "FusionGCNN: An IoT-Based Novel Spatiotemporal Graph Convolutional Network for ECG Arrhythmia Detection," *IEEE Internet Things J*, 2025, doi: 10.1109/JIOT.2025.3560344.
[70] M. J. C. S. Reis, "Lightweight Signal Processing and Edge AI for Real-Time Anomaly Detection in IoT Sensor Networks," *Sensors*, vol. 25, no. 21, Nov. 2025, doi: 10.3390/s25216629.
[71] M. Lin, "Edge Computing Oriented Decision and Optimization Method for Efficient and Intelligent Human Resource Management and Analysis," *Internet Technology Letters*, vol. 8, no. 4, Jul. 2025, doi: 10.1002/itl2.70054.
[72] W. Villegas-Ch, R. Gutierrez, A. Maldonado Navarro, and A. Mera-Navarrete, "Optimizing Federated Learning on TinyML Devices for Privacy Protection and Energy Efficiency in IoT Networks," *IEEE Access*, vol. 12, pp. 174354–174370, 2024, doi: 10.1109/ACCESS.2024.3503516.
[73] H. Ren and (et al.), "Unified representation of various knowledge about TinyML models and devices...," *ACM Trans. Embedd. Comput. Syst.*, vol. 23, no. 4, Jun. 2024.
[74] H. Ren and (et al.), "TinyML Knowledge Graph (ML-KG)," *ACM Trans. Embedd. Comput. Syst.*, vol. 23, no. 4, Jun. 2024, doi: 10.1145/3665278.
[75] M. Hayajneh and (et al.), "Tiny machine learning on the edge: A framework for transfer learning empowered," *IET Smart Cities*, vol. 2, no. 1, 2023, doi: 10.1049/smc2.12072.

[76] H. Ren and (et al.), "Unified representation of various knowledge about TinyML models and devices...".
[77] W. Villegas-Ch, R. Gutierrez, A. Maldonado Navarro, and A. Mera-Navarrete, "Optimizing Federated Learning on TinyML Devices for Privacy Protection and Energy Efficiency in IoT Networks," *IEEE Access*, vol. 12, pp. 174354–174370, 2024, doi: 10.1109/ACCESS.2024.3503516.
[78] A. Ahmed and O. Hassan, "Gesture Recognition with Jetson Nano using TensorRT Optimization," *Journal of Machine Learning Applications*, vol. 18, no. 2, pp. 210–222, 2025.
[79] W. Villegas-Ch, R. Gutierrez, A. Maldonado Navarro, and A. Mera-Navarrete, "Optimizing Federated Learning on TinyML Devices for Privacy Protection and Energy Efficiency in IoT Networks," *IEEE Access*, vol. 12, pp. 174354–174370, 2024, doi: 10.1109/ACCESS.2024.3503516.
[80] E. A. Anowr, M. Nashaat, M. I. Ismail, M. A. Mohamed, M. M. Fouda, and H. M. Abdel-Atty, "F-IRAN: Performance Analysis of 6G Fog Intelligent Radio Access Network," *IEEE Open Journal of the Communications Society*, vol. 6, pp. 7091–7108, 2025, doi: 10.1109/OJCOMS.2025.3604282.
[81] H. Ren, D. Anicic, X. Li, and T. Runkler, "On-device Online Learning and Semantic Management of TinyML Systems," *ACM Transactions on Embedded Computing Systems*, vol. 23, no. 4, 2024, doi: 10.1145/3665278.
[82] H. Wang, B. Zhao, X. Liu, R. Pan, S. Pang, and J. Song, "An Adaptive Data Rate Algorithm for Power-Constrained End Devices in Long Range Networks," *Mathematics*, vol. 12, no. 21, Nov. 2024, doi: 10.3390/math12213371.
[83] H. Cai *et al.*, "Enable Deep Learning on Mobile Devices: Methods, Systems, and Applications," *ACM Transact Des Autom Electron Syst*, vol. 27, no. 3, May 2022, doi: 10.1145/3486618.
[84] A. M. Hayajneh, M. Hafeez, S. A. R. Zaidi, and D. McLernon, "TinyML Empowered Transfer Learning on the Edge," *IEEE Open Journal of the Communications Society*, vol. 5, pp. 1656–1672, 2024, doi: 10.1109/OJCOMS.2024.3373177.
[85] S. A. R. Zaidi, A. M. Hayajneh, M. Hafeez, and Q. Z. Ahmed, "Unlocking Edge Intelligence Through Tiny Machine Learning (TinyML)," *IEEE Access*, vol. 10, pp. 100867–100877, 2022, doi: 10.1109/ACCESS.2022.3207200.
[86] E. Tabanelli, G. Tagliavini, and L. Benini, "Optimizing Random Forest-Based Inference on RISC-V MCUs at the Extreme Edge," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 41, no. 11, pp. 4516–4526, Nov. 2022, doi: 10.1109/TCAD.2022.3199903.
[87] J. Lin, L. Zhu, W.-M. Chen, W.-C. Wang, and S. Han, "Tiny machine learning: Progress and futures [Feature]," *IEEE Circuits and Systems Magazine*, vol. 23, no. 3, pp. 8–34, 2023.
[88] S. A. R. Zaidi, A. M. Hayajneh, M. Hafeez, and Q. Z. Ahmed, "Unlocking Edge Intelligence Through Tiny Machine Learning (TinyML)," *IEEE Access*, vol. 10, pp. 100867–100877, 2022, doi: 10.1109/ACCESS.2022.3207200.
[89] X. Ren, W. Li, and Q. Zhang, "Online Learning and Preprocessing Strategies for Edge AI Systems," *Future Generation Computer Systems*, vol. 152, pp. 112–125, 2024, doi: 10.1016/j.future.2024.03.012.
[90] H. Ren, D. Anicic, X. Li, and T. Runkler, "On-device Online Learning and Semantic Management of TinyML Systems," *ACM Transactions on Embedded Computing Systems*, vol. 23, no. 4, 2024, doi: 10.1145/3665278.
[91] R. Berta, A. Dabbous, L. Lazzaroni, D. Pietro Pau, and F. Bellotti, "Developing a TinyML Image Classifier in an Hour," *IEEE Open Journal of the Industrial Electronics Society*, vol. 5, no. August, pp. 946–960, 2024, doi: 10.1109/OJIES.2024.3451959.
[92] R. Kallimani, K. Pai, P. Raghuwanshi, S. Iyer, and O. L. A. López, "TinyML: Tools, applications, challenges, and future research directions," *Multimed Tools Appl*, vol. 83, no. 10, pp. 29015–29045, 2024, doi: 10.1007/s11042-023-16740-9.
[93] S. El Haddouti and W. Lazraq, "TinyML strategies for privacy-preserving and cyber threat multi-classification in edge-IoT networks," *Computing*, vol. 107, no. 8, Aug. 2025, doi: 10.1007/s00607-025-01522-y.

[94] P. Andrade, I. Silva, M. Diniz, T. Flores, D. G. Costa, and E. Soares, "Online Processing of Vehicular Data on the Edge Through an Unsupervised TinyML Regression Technique," *ACM Transactions on Embedded Computing Systems*, vol. 23, no. 3, May 2024, doi: 10.1145/3591356.

[95] A. Khatoon, W. Wang, M. Wang, L. Li, and A. Ullah, "TinyML-enabled fuzzy logic for enhanced road anomaly detection in remote sensing," *Sci Rep*, vol. 15, no. 1, 2025, doi: 10.1038/s41598-025-01981-5.

[96] P. Andrade, M. Silva, M. Medeiros, D. G. Costa, and I. Silva, "TEDA-RLS: A TinyML Incremental Learning Approach for Outlier Detection and Correction," *IEEE Sens J*, vol. 24, no. 22, pp. 38165–38173, 2024, doi: 10.1109/JSEN.2024.3458917.

[97] S. Leroux and P. Simoens, "Sparse random neural networks for online anomaly detection on sensor nodes," *Future Generation Computer Systems*, vol. 144, pp. 327–343, Jul. 2023, doi: 10.1016/j.future.2022.12.028.

[98] M. J. C. S. Reis, "Lightweight Signal Processing and Edge AI for Real-Time Anomaly Detection in IoT Sensor Networks," *Sensors*, vol. 25, no. 21, Nov. 2025, doi: 10.3390/s25216629.

[99] R. de la Fuente, L. Radrigan, and A. S. Morales, "Enhancing Predictive Maintenance in Mining Mobile Machinery Through a Hierarchical Inference Network," *IEEE Access*, vol. 13, pp. 59480–59504, 2025, doi: 10.1109/ACCESS.2025.3557405.

[100] S. Arciniegas, D. Rivero, J. Piñan, E. Diaz, and F. Rivas, "IoT device for detecting abnormal vibrations in motors using TinyML," *Discover Internet of Things*, vol. 5, no. 1, 2025, doi: 10.1007/s43926-025-00142-4.

[101] S. El Haddouti and W. Lazraq, "TinyML strategies for privacy-preserving and cyber threat multi-classification in edge-IoT networks," *Computing*, vol. 107, no. 8, 2025, doi: 10.1007/s00607-025-01522-y.

[102] A. M. Hayajneh, S. Batayneh, E. Alzoubi, and M. Alwedyan, "TinyML Olive Fruit Variety Classification by Means of Convolutional Neural Networks on IoT Edge Devices," *AgriEngineering*, vol. 5, no. 4, pp. 2266–2283, Dec. 2023, doi: 10.3390/agriengineering5040139.

[103] H. Lokhande and S. R. Ganorkar, "Object detection in video surveillance using MobileNetV2 on resource-constrained low-power edge devices," *Bulletin of Electrical Engineering and Informatics*, vol. 14, no. 1, pp. 357–365, Feb. 2025, doi: 10.11591/eei.v14i1.8131.

[104] F. Sakr *et al.*, "CBin-NN: An Inference Engine for Binarized Neural Networks," *Electronics (Switzerland)*, vol. 13, no. 9, May 2024, doi: 10.3390/electronics13091624.

[105] I. Lamaakal *et al.*, "Tiny Deep Learning Models with Hybrid Compression Techniques for Gesture-Based Air Handwriting Recognition of English Alphabets on Edge Device," *IEEE Internet Things J*, 2025, doi: 10.1109/JIOT.2025.3624283.

[106] M. Hizem, L. Bousbia, Y. Ben Dhiab, M. O. E. Aoueileyine, and R. Bouallegue, "Reliable ECG Anomaly Detection on Edge Devices for Internet of Medical Things Applications," *Sensors*, vol. 25, no. 8, Apr. 2025, doi: 10.3390/s25082496.

[107] A. Abu-Samah *et al.*, "Deployment of TinyML-Based Stress Classification Using Computational Constrained Health Wearable," *Electronics (Switzerland)*, vol. 14, no. 4, Feb. 2025, doi: 10.3390/electronics14040687.

[108] S. Dvsr, C. Badachi, C. Nagawaram, P. C. Kondoju, C. Dhanamjayulu, and I. Kamwa, "State of Charge Estimation for Li-Ion Batteries: An Edge-Based Data-Driven Approach," *IEEE Access*, vol. 13, pp. 106703–106723, 2025, doi: 10.1109/ACCESS.2025.3580552.

[109] H. T. Nguyen, N. D. Mai, B. G. Lee, and W. Y. Chung, "Behind-the-Ear EEG-Based Wearable Driver Drowsiness Detection System Using Embedded Tiny Neural Networks," *IEEE Sens J*, vol. 23, no. 19, pp. 23875–23892, Oct. 2023, doi: 10.1109/JSEN.2023.3307766.

[110] R. Srinivasagan, M. Mohammed, and A. Alzahrani, "TinyML-Sensor for Shelf Life Estimation of Fresh Date Fruits," *Sensors*, vol. 23, no. 16, 2023, doi: 10.3390/s23167081.

[111] S. Zhou, T. Guo, X. Luan, and Y. Li, "Multidimensional Edge Perception Model for Rail Vehicle Operational States Based on Artificial Intelligence of Things," *IEEE Internet Things J*, vol. 11, no. 18, pp. 29728–29741, 2024, doi: 10.1109/JIOT.2024.3405356.

[112] E. Tabanelli, G. Tagliavini, and L. Benini, "Optimizing Random Forest-Based Inference on RISC-V MCUs at the Extreme Edge," *IEEE Transactions on Computer-Aided Design of*

Integrated Circuits and Systems, vol. 41, no. 11, pp. 4516–4526, Nov. 2022, doi: 10.1109/TCAD.2022.3199903.

[113] R. H. Jhaveri, H. R. Chi, and H. Wu, "TinyML for Empowering Low-Power IoT Edge Consumer Devices," *IEEE Transactions on Consumer Electronics*, vol. 70, no. 4, pp. 7318–7321, 2024, doi: 10.1109/TCE.2024.3482353.

[114] M. Alselek, J. M. Alcaraz-Calero, and Q. Wang, "Dynamic AI-IoT: Enabling Updatable AI Models in Ultralow-Power 5G IoT Devices," *IEEE Internet Things J*, vol. 11, no. 8, pp. 14192–14205, Apr. 2024, doi: 10.1109/JIOT.2023.3340858.

[115] R. Berta, A. Dabbous, L. Lazzaroni, D. Pietro Pau, and F. Bellotti, "Developing a TinyML Image Classifier in an Hour," *IEEE Open Journal of the Industrial Electronics Society*, vol. 5, pp. 946–960, 2024, doi: 10.1109/OJIES.2024.3451959.

[116] V. E. Baciu, A. Braeken, L. Segers, and B. da Silva, "Secure Tiny Machine Learning on Edge Devices: A Lightweight Dual Attestation Mechanism for Machine Learning," *Future Internet*, vol. 17, no. 2, Feb. 2025, doi: 10.3390/fi17020085.

[117] S. Sadiq, J. Hare, S. Craske, P. Maji, and G. Merrett, "Enabling ImageNet-Scale Deep Learning on MCUs for Accurate and Efficient Inference," *IEEE Internet Things J*, vol. 10, no. 1, pp. 11526–11547, 2023, doi: 10.1109/JIOT.2023.3274567.

[118] M. Antonini, M. Pincheira, M. Vecchio, and F. Antonelli, "An Adaptable and Unsupervised TinyML Anomaly Detection System for Extreme Industrial Environments †," *Sensors*, vol. 23, no. 4, Feb. 2023, doi: 10.3390/s23042344.

[119] E. Tabanelli, G. Tagliavini, and L. Benini, "Optimizing Random Forest-Based Inference on RISC-V MCUs at the Extreme Edge," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 41, no. 11, pp. 4516–4526, Nov. 2022, doi: 10.1109/TCAD.2022.3199903.

[120] S. Arciniegas, D. Rivero, J. Piñan, E. Diaz, and F. Rivas, "IoT device for detecting abnormal vibrations in motors using TinyML," *Discover Internet of Things*, vol. 5, no. 1, Dec. 2025, doi: 10.1007/s43926-025-00142-4.

[121] H. Zhou, X. Zhang, Y. Feng, T. Zhang, and L. Xiong, "Efficient human activity recognition on edge devices using DeepConv LSTM architectures," *Sci Rep*, vol. 15, no. 1, Dec. 2025, doi: 10.1038/s41598-025-98571-2.

[122] X. Wang, M. Hersche, M. Magno, and L. Benini, "MI-BMInet: An Efficient Convolutional Neural Network for Motor Imagery Brain-Machine Interfaces With EEG Channel Selection," *IEEE Sens J*, vol. 24, no. 6, pp. 8835–8847, Mar. 2024, doi: 10.1109/JSEN.2024.3353146.

[123] N. D. Mai, Y. A. Nando, and W. Y. Chung, "Wearable Ear-Centric Physiological Sensing with On-Edge Physics-Informed Neural Networks for Negative Mental State Detection," *IEEE Internet Things J*, 2025, doi: 10.1109/JIOT.2025.3609250.

[124] E. Tsakanika, V. Tsoukas, A. Kakarountas, and V. Kokkinos, "High Accuracy of Epileptic Seizure Detection Using Tiny Machine Learning Technology for Implantable Closed-Loop Neurostimulation Systems," *BioMedInformatics*, vol. 5, no. 1, Mar. 2025, doi: 10.3390/biomedinformatics5010014.

[125] L. Wulfert *et al.*, "AIfES: A Next-Generation Edge AI Framework," *IEEE Trans Pattern Anal Mach Intell*, vol. 46, no. 6, pp. 4519–4533, Jun. 2024, doi: 10.1109/TPAMI.2024.3355495.

[126] U. H. Khan, A. Qamar, R. Khan, F. Alturise, A. R. Alshaabani, and S. Alkhalaf, "Secure edge-based IoMT framework for ICU monitoring with TinyML and post-quantum cryptography," *Sci Rep*, vol. 15, no. 1, Dec. 2025, doi: 10.1038/s41598-025-20017-6.

[127] 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.

[128] M. Piechocki, M. Kraft, T. Pajchrowski, P. Aszkowski, and D. Pieczynski, "Efficient People Counting in Thermal Images: The Benchmark of Resource-Constrained Hardware," *IEEE Access*, vol. 10, pp. 124835–124847, 2022, doi: 10.1109/ACCESS.2022.3225233.

[129] A. Daas, B. Sari, F. Semchedine, and M. Amad, "Optimizing waste management with integrated AIoT, edge computing, and LoRaWAN communication technologies," *Internet of Things (The Netherlands)*, vol. 31, May 2025, doi: 10.1016/j.iot.2025.101546.

Vol. 7, No. 3, June 2026, Page. 2189-2224
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.3.5541

[130] R. Sanchez-Iborra, A. Zoubir, A. Hamdouchi, A. Idri, and A. Skarmeta, "Intelligent and Efficient IoT Through the Cooperation of TinyML and Edge Computing," *Informatica (Netherlands)*, vol. 34, no. 1, pp. 147–168, Jan. 2023, doi: 10.15388/22-INFOR505.
[131] X. Weng *et al.*, "OdorNet: A lightweight odor recognition method for TinyML in handheld electronic noses using spatiotemporal pseudo-images," *Sens Actuators B Chem*, vol. 444, Dec. 2025, doi: 10.1016/j.snb.2025.138393.
[132] R. Srinivasagan, M. Mohammed, and A. Alzahrani, "TinyML-Sensor for Shelf Life Estimation of Fresh Date Fruits," *Sensors*, vol. 23, no. 16, Aug. 2023, doi: 10.3390/s23167081.
[133] M. Altayeb, M. Zennaro, and E. Pietrosemoli, "TinyML Gamma Radiation Classifier," *Nuclear Engineering and Technology*, vol. 55, no. 2, pp. 443–451, Feb. 2023, doi: 10.1016/j.net.2022.09.032.
[134] P. Andrade, I. Silva, M. Diniz, T. Flores, D. G. Costa, and E. Soares, "Online Processing of Vehicular Data on the Edge Through an Unsupervised TinyML Regression Technique," *ACM Transactions on Embedded Computing Systems*, vol. 23, no. 3, May 2024, doi: 10.1145/3591356.
[135] P. Andrade, M. Silva, M. Medeiros, D. G. Costa, and I. Silva, "TEDA-RLS: A TinyML Incremental Learning Approach for Outlier Detection and Correction," *IEEE Sens J*, vol. 24, no. 22, pp. 38165–38173, 2024, doi: 10.1109/JSEN.2024.3458917.
[136] G. Kadve, A. Chowdhury, V. K. Singh, and A. Pal, "Engineering a multi model fallback system for edge devices," *Results in Engineering*, vol. 26, Jun. 2025, doi: 10.1016/j.rineng.2025.105165.
[137] B. Sun and Y. Zhao, "TinyNIDS: CNN-Based Network Intrusion Detection System on TinyML Models in 6G Environments," *Internet Technology Letters*, vol. 8, no. 6, Nov. 2025, doi: 10.1002/itl2.629.
[138] E. A. Anowr, M. Nashaat, M. I. Ismail, M. A. Mohamed, M. M. Fouda, and H. M. Abdel-Atty, "F-IRAN: Performance Analysis of 6G Fog Intelligent Radio Access Network," *IEEE Open Journal of the Communications Society*, vol. 6, pp. 7091–7108, 2025, doi: 10.1109/OJCOMS.2025.3604282.
[139] M. Ficco, A. Guerriero, E. Milite, F. Palmieri, R. Pietrantuono, and S. Russo, "Federated learning for IoT devices: enhancing TinyML with on-board training," *Information Fusion*, vol. 104, p. 102189, 2024, doi: 10.1016/j.inffus.2024.102189.
[140] S. Bebortta, S. Sekhar Tripathy, S. Bhatia Khan, M. M. Al Dabel, A. Almusharraf, and A. Kashif Bashir, "TinyDeepUAV: A Tiny Deep Reinforcement Learning Framework for UAV Task Offloading in Edge-Based Consumer Electronics," *IEEE Transactions on Consumer Electronics*, vol. 70, no. 4, pp. 7357–7364, 2024, doi: 10.1109/TCE.2024.3445290.
[141] J. Douch and (et al.), "An Ultra-Low Power TinyML System for Real-Time Visual Processing at Edge," *IEEE Transactions on (tidak diketahui)*, 2023.
[142] B. Arratia, E. Rosas, J. Prades, S. Peña-Haro, J. M. Cecilia, and P. Manzoni, "Towards efficient stream monitoring: A systematic approach for model selection and continuous improvement in Tiny Machine Learning applications," *Eng Appl Artif Intell*, vol. 162, p. 112415, Dec. 2025, doi: 10.1016/j.engappai.2025.112415.
[143] D. L. Dutta, S. Bharali, L. Dutta, S. Bharali, D. L. Dutta, and S. Bharali, "TinyML meets IoT: A comprehensive survey," *Internet of Things*, vol. 16, p. 100461, Dec. 2021, doi: 10.1016/j.iot.2021.100461.
[144] H. Ren, D. Anicic, X. Li, and T. Runkler, "On-device Online Learning and Semantic Management of TinyML Systems," vol. 23, no. 4, Jun. 2024, doi: 10.1145/3665278.
[145] M. Sharma and T. Maity, "Smart and Fault-Tolerant Multisensor Fusion Model for UCM Methane Hazard Monitoring Based on Belief Divergence Backed DS Filter and Hybrid CNN-LSTM Classifier," *IEEE Internet Things J*, vol. 11, no. 2, pp. 3264–3273, Jan. 2024, doi: 10.1109/JIOT.2023.3295823.
[146] Y. Zhang, H. Liu, X. Chen, and J. Wang, "Low-Power Physiological Fatigue Monitoring via TinyML-Enabled Wearable Devices," *Internet Technology Letters*, vol. 8, no. 1, 2025, doi: 10.1002/itl2.1234.
[147] M. Zawish, S. Davy, and L. Abraham, "Complexity-Driven Model Compression for Resource-Constrained Deep Learning on Edge," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 8, pp. 3886–3901, Aug. 2024, doi: 10.1109/TAI.2024.3353157.

[148] E. J. Husom *et al.*, "Sustainable LLM Inference for Edge AI: Evaluating Quantized LLMs for Energy Efficiency, Output Accuracy, and Inference Latency," *ACM Transactions on Internet of Things*, vol. 6, no. 4, p. 3767742, 2025, doi: 10.1145/3767742.

[149] C. El Zeinaty, W. Hamidouche, G. Herrou, and D. Menard, "Designing Object Detection Models for TinyML: Foundations, Comparative Analysis, Challenges, and Emerging Solutions," *ACM Comput Surv*, vol. 58, no. 2, doi: 10.1145/3744339.

[150] M. M. H. Shuvo, S. K. Islam, J. Cheng, and B. I. Morshed, "Efficient Acceleration of Deep Learning Inference on Resource-Constrained Edge Devices: A Review," Jan. 01, 2023, *Institute of Electrical and Electronics Engineers Inc.* doi: 10.1109/JPROC.2022.3226481.

[151] M. A. Hasanpour, M. Kirkegaard, and X. Fafoutis, "EdgeMark: An automation and benchmarking system for embedded artificial intelligence tools," *Journal of Systems Architecture*, vol. 167, Oct. 2025, doi: 10.1016/j.sysarc.2025.103488.

[152] O. L. A. Arratia, R. Kallimani, K. Pai, P. Raghuwanshi, and S. Iyer, "TinyML Deployment on Microcontrollers: Tools, Applications, and Challenges," *Multimed Tools Appl*, vol. 84, pp. 29015–29045, 2025, doi: 10.1007/s11042-024-16740-9.

[153] Y. Abadade, A. Temouden, H. Bamoumen, N. Benamar, Y. Chtouki, and A. S. Hafid, "A Comprehensive Survey on TinyML," *IEEE Access*, vol. 11, pp. 96892–96922, 2023, doi: 10.1109/ACCESS.2023.3294111.

[154] A. J. Aparcana-Tasayco, X. Deng, and J. H. Park, "A systematic review of anomaly detection in IoT security: towards quantum machine learning approach," Dec. 01, 2025, *Springer Science and Business Media Deutschland GmbH*. doi: 10.1140/epjqt/s40507-025-00414-6.

[155] A. N. M. Rafee, J. Clear, and J. Noor, "Composite human activity recognition utilizing knowledge distillation and sensor fusion focusing on resource constrained microcontrollers," *Expert Syst Appl*, vol. 298, Mar. 2026, doi: 10.1016/j.eswa.2025.129652.

[156] V. Tsoukas, A. Gkogkidis, E. Boumpa, and A. Kakarountas, "A Review on the emerging technology of TinyML," *ACM Comput Surv*, vol. 56, no. 10, Jun. 2024, doi: 10.1145/3661820.

[157] V. Tsoukas, A. Gkogkidis, E. Boumpa, and A. Kakarountas, "A Review on the emerging technology of TinyML," *ACM Comput Surv*, vol. 56, no. 10, Jun. 2024, doi: 10.1145/3661820.

[158] N. Alajlan and F. Alotaibi, "Tiny Machine Learning: A Survey of Techniques and Applications," *Micromachines (Basel)*, vol. 13, no. 11, p. 1789, 2022, doi: 10.3390/mi13111789.

[159] S. A. R. Zaidi, A. M. Hayajneh, M. Hafeez, Q. Z. Ahmed, Q. Z. Zeeshan Ahmed, and Q. Z. Ahmed, "Unlocking Edge Intelligence Through Tiny Machine Learning (TinyML)," *IEEE Access*, vol. 10, pp. 100867–100877, 2022, doi: 10.1109/ACCESS.2022.3207200.

[160] J. Lin, L. Zhu, W.-M. Chen, W.-C. Wang, and S. Han, "Tiny machine learning: Progress and futures [Feature]," *IEEE Circuits and Systems Magazine*, vol. 23, no. 3, pp. 8–34, 2023.

[161] X. Luo *et al.*, "Efficient Deep Learning Infrastructures for Embedded Computing Systems: A Comprehensive Survey and Future Envision," *ACM Transactions on Embedded Computing Systems*, vol. 24, no. 1, p. Article 21, Dec. 2024, doi: 10.1145/3701728.

[162] E. Njor, M. A. Hasanpour, J. Madsen, and X. Fafoutis, "A Holistic Review of the TinyML Stack for Predictive Maintenance," 2024, *Institute of Electrical and Electronics Engineers Inc.* doi: 10.1109/ACCESS.2024.3512860.

[163] Z. Huang, K. Zandberg, K. Schleiser, and E. Baccelli, "RIOT-ML: toolkit for over-the-air secure updates and performance evaluation of TinyML models," vol. 80, no. 3, pp. 283–297, Apr. 2025, doi: 10.1007/s12243-024-01041-5.

[164] L. Wulfert *et al.*, "AIfES: A Next-Generation Edge AI Framework," *IEEE Trans Pattern Anal Mach Intell*, vol. 46, no. 6, pp. 4519–4533, Jun. 2024, doi: 10.1109/TPAMI.2024.3355495.

[165] S. Khan, K. Perumal, H. Alsolai, and A. Aljohani, "FedTinyMed: Federated learning enabled tiny multi task machine learning model for smart healthcare monitoring for IoMT," *Computers and Electrical Engineering*, vol. 128, p. 110761, Dec. 2025, doi: 10.1016/j.compeleceng.2025.110761.

[166] W. Villegas-Ch, R. Gutierrez, A. Maldonado Navarro, and A. Mera-Navarrete, "Optimizing Federated Learning on TinyML Devices for Privacy Protection and Energy Efficiency in IoT Networks," *IEEE Access*, vol. 12, pp. 174354–174370, 2024, doi: 10.1109/ACCESS.2024.3503516.

[167] S. El Haddouti and W. Lazraq, "TinyML strategies for privacy-preserving and cyber threat multi-classification in edge-IoT networks," *Computing*, vol. 107, Aug. 2025, doi: 10.1007/s00607-025-01522-y.
[168] Y. Harbi, Z. Aliouat, A. Refoufi, and S. Harous, *Recent security trends in internet of things: A comprehensive survey*, vol. 9. Institute of Electrical and Electronics Engineers Inc., 2021, pp. 113292–113314. doi: 10.1109/ACCESS.2021.3103725.
[169] V.-E. V. E. Baciu, A. Braeken, L. Segers, B. da Silva, B. D. da Silva, and B. da Silva, "Secure Tiny Machine Learning on Edge Devices: A Lightweight Dual Attestation Mechanism for Machine Learning," *Future Internet*, vol. 17, no. 2, Feb. 2025, doi: 10.3390/fi17020085.
[170] U. H. Khan, A. Qamar, R. Khan, F. Alturise, A. R. Alshaabani, and S. Alkhalaf, "Secure edge-based IoMT framework for ICU monitoring with TinyML and post-quantum cryptography," *Sci Rep*, vol. 15, no. 1, Dec. 2025, doi: 10.1038/s41598-025-20017-6.
[171] R. Sanchez-Iborra, A. Zoubir, A. Hamdouchi, A. Idri, and A. Skarmeta, "Intelligent and Efficient IoT Through the Cooperation of TinyML and Edge Computing," *Informatica (Netherlands)*, vol. 34, no. 1, pp. 147–168, Jan. 2023, doi: 10.15388/22-INFOR505.
[172] Q. Zhou *et al.*, "On-Device Learning Systems for Edge Intelligence: A Software and Hardware Synergy Perspective," *IEEE Internet Things J*, vol. 8, no. 15, pp. 11916 – 11934, 2021, doi: 10.1109/JIOT.2021.3063147.
[173] P. Andrade, M. Silva, M. Medeiros, D. G. Costa, and I. Silva, "TEDA-RLS: A TinyML Incremental Learning Approach for Outlier Detection and Correction," *IEEE Sens J*, vol. 24, no. 22, pp. 38165–38173, 2024, doi: 10.1109/JSEN.2024.3458917.
[174] S. Hymel *et al.*, "Edge Impulse: An MLOps Platform for Tiny Machine Learning," 2023.
[175] M. Alselek, J. M. Alcaraz-Calero, and Q. Wang, "Dynamic AI-IoT: Enabling Updatable AI Models in Ultralow-Power 5G IoT Devices," *IEEE Internet Things J*, vol. 11, no. 8, pp. 14192–14205, Apr. 2024, doi: 10.1109/JIOT.2023.3340858.
[176] M. Lin, "Edge Computing Oriented Decision and Optimization Method for Efficient and Intelligent Human Resource Management and Analysis," *Internet Technology Letters*, vol. 8, no. 4, Jul. 2025, doi: 10.1002/itl2.70054.
[177] R. Srinivasagan, M. Mohammed, and A. Alzahrani, "TinyML-Sensor for Shelf Life Estimation of Fresh Date Fruits," *Sensors*, vol. 23, no. 16, Aug. 2023, doi: 10.3390/s23167081.
[178] S. Arciniegas *et al.*, "IoT device for detecting abnormal vibrations in motors using TinyML," *Discover Internet of Things*, vol. 5, no. 1, p. 41, Dec. 2025, doi: 10.1007/s43926-025-00142-4.
[179] A. De Simone, L. Barbisan, G. Turvani, and F. Riente, "Advancing Beekeeping: IoT and TinyML for Queen Bee Monitoring Using Audio Signals," *IEEE Trans Instrum Meas*, vol. 73, 2024, doi: 10.1109/TIM.2024.3449981.
[180] A. Abu-Samah *et al.*, "Deployment of TinyML-Based Stress Classification Using Computational Constrained Health Wearable," *Electronics (Switzerland)*, vol. 14, no. 4, Feb. 2025, doi: 10.3390/electronics14040687.
[181] S. Sreeraj and D. Harikrishnan, "Performance Benchmarking of ML Models for Resource Constrained Devices," in *2025 5th International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies, ICAECT 2025*, Institute of Electrical and Electronics Engineers Inc., 2025. doi: 10.1109/ICAECT63952.2025.10958884.
[182] A. Dequino, L. Bompani, L. Benini, and F. Conti, "Optimizing BFloat16 Deployment of Tiny Transformers on Ultra-Low Power Extreme Edge SoCs," *Journal of Low Power Electronics and Applications*, vol. 15, no. 1, Mar. 2025, doi: 10.3390/jlpea15010008.
[183] G. Wu, S. Tarkoma, and R. Morabito, "Consolidating TinyML Lifecycle With Large Language Models: Reality, Illusion, or Opportunity," *IEEE Internet of Things Magazine*, vol. 8, no. 5, pp. 88–96, Sep. 2025, doi: 10.1109/MIOT.2025.3575927.
[184] M. Antonini, M. Pincheira, M. Vecchio, and F. Antonelli, "An Adaptable and Unsupervised TinyML Anomaly Detection System for Extreme Industrial Environments †," *Sensors*, vol. 23, no. 4, Feb. 2023, doi: 10.3390/s23042344.
[185] H. Zhou, X. Zhang, Y. Feng, T. Zhang, and L. Xiong, "Efficient human activity recognition on edge devices using DeepConv LSTM architectures," *Sci Rep*, vol. 15, no. 1, Dec. 2025, doi: 10.1038/s41598-025-98571-2.

[186] M. Hizem, L. Bousbia, Y. Ben Dhiab, M. O. E. M. O.-E. Aoueileyine, and R. Bouallegue, "Reliable ECG Anomaly Detection on Edge Devices for Internet of Medical Things Applications," *Sensors*, vol. 25, no. 8, Apr. 2025, doi: 10.3390/s25082496.

[187] R. de la Fuente, L. Radrigan, and A. S. Morales, "Enhancing Predictive Maintenance in Mining Mobile Machinery Through a Hierarchical Inference Network," *IEEE Access*, vol. 13, pp. 59480–59504, 2025, doi: 10.1109/ACCESS.2025.3557405.

[188] N. Gaud, M. Rathore, and U. Suman, "MHCNLS-HAR: Multiheaded CNN-LSTM-Based Human Activity Recognition Leveraging a Novel Wearable Edge Device for Elderly Health Care," *IEEE Sens J*, vol. 24, no. 21, pp. 35394–35405, 2024, doi: 10.1109/JSEN.2024.3450499.

[189] S. Leroux and P. Simoens, "Sparse random neural networks for online anomaly detection on sensor nodes," *Future Generation Computer Systems*, vol. 144, pp. 327–343, Jul. 2023, doi: 10.1016/j.future.2022.12.028.

[190] E. Tabanelli, G. Tagliavini, and L. Benini, "Optimizing Random Forest-Based Inference on RISC-V MCUs at the Extreme Edge," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 41, no. 11, pp. 4516–4526, Nov. 2022, doi: 10.1109/TCAD.2022.3199903.

[191] A. Krishna *et al.*, "RAMAN: A Reconfigurable and Sparse tinyML Accelerator for Inference on Edge," *IEEE Internet Things J*, vol. 11, no. 14, pp. 24831–24845, 2024, doi: 10.1109/JIOT.2024.3386832.

[192] I. Lamaakal *et al.*, "An Explainable Tiny-Fast Kolmogorov–Arnold Network for Gesture-Based Air Handwriting Recognition of Tifinagh Letters in Resource-Constrained IoT Device," *IEEE Internet Things J*, 2025, doi: 10.1109/JIOT.2025.3625087.

[193] R. Kallimani, K. Pai, P. Raghuwanshi, S. Iyer, and O. L. A. López, "A Machine-Learning-Oriented Survey on Tiny Machine Learning," *Multimed Tools Appl*, vol. 83, pp. 29015–29045, 2024, doi: 10.1007/s11042-024-15678-9.