Journal of Technology Research (JTR)
Volume 4, Issue 2, (2026), pp 073-092, ISSN 3005-639X

A Comprehensive Review of Federated Learning and TinyML for Localization

Akpojoto Siemuri [*1,2], Mohammed Elmusrati[1,3]

[1] School of Technology and Innovation, University of Vaasa, Vaasa, Finland.

[2] OneNav Finland Oy, Tampere, Finland.

[3] Libyan Authority of Scientific Research, Tripoli, Libya.

*Corresponding author email: akpo.siemuri@gmail.com

Received: 10-12-2025 | Accepted: 06-04-2026 | Available online: 16-06-2026 | DOI:10.26629/jtr.2026.08

ABSTRACT

The growing demand for intelligent and context-aware Internet of Things (IoT) systems has spurred the advancement of machine learning (ML) solutions for real-time localization. The deployment of ML models on distributed and resource-constrained edge devices offers substantial opportunities while posing unique challenges. This paper presents a structured and comparative review of strategies for adapting and designing ML models for deployment on embedded hardware with limited resources. We emphasize Tiny Machine Learning (TinyML), a paradigm that facilitates ultra-low-power ML execution on microcontrollers and sensors, and explore its synergy with Federated Learning (FL), which enables collaborative model training across devices without centralizing sensitive data. We examine training strategies, deployment workflows from cloud to edge, model optimization techniques, and key applications of TinyML and FL in localization. Benefits such as real-time processing, privacy preservation, and energy efficiency are analysed, along with challenges including computational constraints, interoperability, and security vulnerabilities. The article concludes by identifying research gaps and outlining future directions for integrating TinyML and FL in large-scale localization systems.

Keywords: Edge computing, Federated learning, TinyML, Machine learning, Localization.

مراجعة شاملة للتعلّم الاتحادي والتعلّم الآلي المصغّر لأغراض تحديد الاماكن

أكبويوتو سيموراي[1،2]، محمد سالم المصراتي[1،3]

[1]كلية التقنية والابتكار، جامعة فازا، فنلندا

[2]شركة ون ناف العالمية، تامبري، فنلندا

[3]الهيئة الليبية للبحث العلمي، طرابلس، ليبيا

ملخـــــص البحـــــث

أدى الطلب المتزايد على أنظمة إنترنت الأشياء الذكية والمدركة للسياق إلى تسريع تطوير حلول التعلم الآلي لأغراض تحديد الموقع في الزمن الحقيقي. ويوفر نشر نماذج التعلم الآلي على أجهزة الحافة الموزعة والمحدودة الموارد فرصًا كبيرة، إلا أنه يطرح في الوقت ذاته مجموعة من التحديات الفريدة. تقدم هذه الورقة مراجعة منهجية ومقارنة للاستراتيجيات المستخدمة في تكييف وتصميم نماذج التعلم الآلي بما يلائم نشرها على الأجهزة المضمنة ذات الموارد المحدودة

ويركز هذا البحث على التعلم الآلي المصغر، بوصفه نهجًا يتيح تنفيذ نماذج التعلم الآلي باستهلاك فائق الانخفاض للطاقة على المتحكمات الدقيقة وأجهزة الاستشعار، كما يستعرض أوجه التكامل بينه وبين التعلم الاتحادي، الذي يتيح التدريب التعاوني للنماذج عبر أجهزة متعددة دون الحاجة إلى تجميع البيانات الحساسة في موقع مركزي. كذلك، نناقش استراتيجيات التدريب، وآليات النشر من السحابة إلى الحافة، وتقنيات تحسين النماذج، وأبرز تطبيقات التعلم الآلي المصغر والتعلم الاتحادي في مجال تحديد الموقع كما يتم تحليل عدد من المزايا، من بينها المعالجة في الزمن الحقيقي، والحفاظ على الخصوصية، وكفاءة استهلاك الطاقة، إلى جانب التحديات المرتبطة بالقيود الحاسوبية، وقابلية التشغيل البيني، والثغرات الأمنية. وتختتم المقالة بتحديد الفجوات البحثية القائمة واستعراض الاتجاهات المستقبلية لدمج التعلم الآلي المصغر والتعلم الاتحادي ضمن أنظمة تحديد الموقع واسعة النطاق

الكلمات الدالة: الحوسبة الطرفية، التعلّم الاتحادي، التعلّم الآلي المصغّر، التعلّم الآلي، التمركز وتحديد المواضع

.

1. INTRODUCTION

The rapid proliferation of the Internet of Things (IoT) and edge computing has fundamentally transformed how data is generated, processed, and utilized in modern intelligent systems. As IoT deployments scale across domains such as smart cities, industrial automation, healthcare, and autonomous systems, there is an increasing demand for real-time, context-aware localization that operates reliably under stringent constraints of latency, energy consumption, and privacy.Traditional localization approaches often rely on centralized cloud-based machine learning (ML) pipelines, where raw sensor data is transmitted to remote servers for processing. While effective in high-resource environments, this paradigm introduces significant limitations, including communication latency, bandwidth overhead, and privacy risks associated with transmitting sensitive location data. These challenges are particularly pronounced in resource-constrained and privacy-sensitive IoT environments, where continuous connectivity cannot be guaranteed [11].

To address these limitations, recent advances in edge intelligence have enabled the deployment of machine learning models directly on end devices. In this context, Tiny Machine Learning (TinyML) has emerged as a key enabler for executing lightweight ML models on microcontrollers and embedded systems with extremely limited computational and memory resources. TinyML facilitates low-power, real-time inference at the edge, making it well suited for localization tasks in dynamic and connectivity-constrained environments [1,3].

Complementing this paradigm, Federated Learning (FL) introduces a decentralized training framework in which multiple devices collaboratively learn a shared global model without exchanging raw data. Instead, devices compute local updates and transmit only model parameters or gradients to a central aggregator. This approach not only reduces communication overhead but also enhances data privacy and regulatory compliance, making it particularly relevant for localization systems that handle sensitive geospatial information [2]. The integration of TinyML and Federated Learning—often referred to as Federated building scalable, privacy-preserving, and energy-efficient localization systems. By combining on-device inference with distributed collaborative learning, this paradigm enables intelligent decision-making directly at the edge while continuously improving model performance across heterogeneous environments. Despite the growing body of research in this area, existing studies are often fragmented across disciplines, focusing either on model optimization for edge deployment, federated learning protocols, or domain-specific localization techniques. A systematic and

comparative synthesis of these developments is therefore necessary to provide a unified understanding of the design space, trade-offs, and open challenges. This paper presents a comprehensive review of TinyML and Federated Learning for localization in IoT systems. It examines training strategies, deployment pipelines, and application domains, while also addressing critical challenges such as resource constraints, data heterogeneity, communication efficiency, and regulatory compliance. Furthermore, the paper highlights emerging research directions, including hierarchical federated architectures and communication-efficient learning techniques tailored for edge-based localization.OVERVIEW OF TINYML AND FEDERATED LEARNING TinyML refers to the deployment of ML models on extremely resource-constrained devices, typically with memory in the order of kilobytes and power consumption in microwatts. It involves techniques to shrink model size and computational requirements while maintaining acceptable performance. Federated Learning (FL), on the other hand, is a distributed ML approach where multiple devices collaboratively train a shared model under the orchestration of a central server. Instead of transmitting raw data, devices send model updates, preserving data privacy. The integration of FL with TinyML—often termed Federated TinyML—allows resource-limited devices to participate in collaborative learning, making it particularly suitable for IoT scenarios where devices are numerous and heterogeneous [3,4].

This combination is especially promising for localization tasks, as it leverages distributed sensing while addressing privacy and efficiency concerns. For example, in a network of IoT sensors, Federated Learning can aggregate local insights without exposing individual device data, enabling robust and privacy-preserving location estimation [5–7].modern cloud infrastructure may appear capable of handling vast data volumes, practical constraints such as cost, latency, and connectivity limitations often make centralized processing infeasible. Edge machine learning (Edge ML) addresses these challenges by enabling data processing directly at the source on devices located at the network's periphery. This approach is essential for applications requiring low-latency responses, real-time predictions, or operation in environments with poor or non-existent cloud connectivity. Additionally, legal restrictions and the need to pre-process large datasets locally further justify the use of edge computing [8]. Edge ML—also referred to as edge artificial intelligence or edge AI—is defined as the execution of ML algorithms on edge devices to make decisions and predictions as close as possible to the data source [9]. This can be implemented in a distributed manner, forming the basis of distributed edge intelligence—a rapidly evolving research area that enables the deployment of machine learning and deep learning models near the point of data generation [10].

2. TRAINING STRATEGIES FOR EDGE ML MODELS

As edge computing continues to reshape the landscape of intelligent systems, training machine learning models directly on or for edge devices has become a critical area of research and development. Unlike traditional cloud-based workflows, edge ML training strategies must account for constraints such as limited memory, processing power, and energy availability—while still delivering accurate and timely predictions.

This section explores three primary approaches to training edge ML models: on-device training, cloud-to-edge deployment, and federated learning. Each method offers distinct advantages and trade-offs in terms of scalability, security and privacy, latency, and resource efficiency, particularly in applications like IoT localization and real-time inference [12–14].

2.1 On-Device Training

Training ML models directly on edge devices, such as Raspberry Pi, smartphones, or embedded boards, is feasible but resource-intensive. Devices like the Raspberry Pi 4 with at least 4 GB of RAM can handle lightweight training tasks. Performance can be boosted using hardware accelerators, including Google Coral USB, Intel Movidius Neural Compute Stick, or NVIDIA Jetson Nano.

The software stack typically comprises Raspberry Pi OS or lightweight Linux distributions, with Python as the primary language owing to its versatility and extensive libraries. Popular frameworks include Tensor-Flow Lite, PyTorch Mobile, Edge Impulse, and scikit-learn.

For IoT localization, on-device ML facilitates rapid predictions by eliminating the need to transfer voluminous raw data over networks. Everyday objects like smartwatches, appliances, and vehicles can embed ML models to infer context or location autonomously. Figure 1 illustrates the implementation of ML on edge devices.

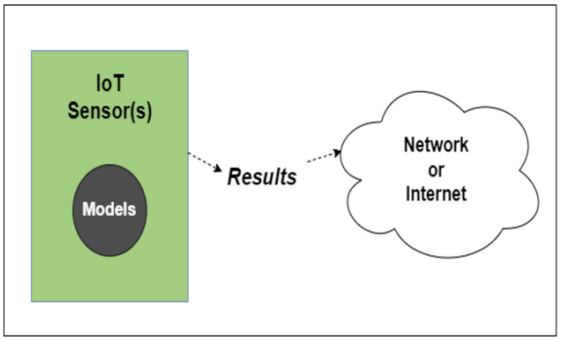

Fig 1. Machine learning model implemented on an edge device, adapted from [20].

However, training intricate models on constrained devices is inefficient. Simple models can be trained using lightweight frameworks, whereas complex ones are typically developed in the cloud or on powerful machines before optimization for edge deployment. Key optimization methods encompass quantization (e.g., int8 or float16), pruning, and knowledge distillation [15,16]. Optimized models are then converted to formats like TensorFlow Lite or ONNX for embedded compatibility [16].

The workflow generally includes: (1) data collection, (2) lightweight pre-processing, (3) training (on-device for simple models, offloaded for complex ones), and (4) deployment. Inference is supported by runtimes such as TensorFlow Lite Interpreter or ONNX Runtime, with remote monitoring for ongoing enhancements [17]. To improve efficiency, hybrid approaches combining local fine-tuning with global models are increasingly adopted [15].

Table 1. Hardware Comparison Between Microcontroller-Class TinyML Devices and Single-Board Computers (SBCs).

Feature	Microcontrollers (TinyML)	Single-Board Computers (SBCs)
Typical Hardware	ARM Cortex-M, ESP32	Raspberry Pi, NVIDIA Jetson
Memory (RAM)	< 1 MB (typically 64–512 KB)	1–32 GB
Software Stack	Bare-metal or RTOS (e.g., FreeRTOS)	Full operating system (e.g., Linux)
On-Device Training Capability	Not feasible; limited to inference or minimal online updates	Feasible for lightweight training and fine-tuning; full training possible with constraints
Primary ML Role	Inference (TinyML models)	Training, fine-tuning, and inference
Federated Learning Role	Client (inference + lightweight updates)	Client or aggregator (training and coordination)

3.1.1 Hardware-Aware Distinction in Edge ML for Localization

In the context of edge-based machine learning, a critical distinction often overlooked in existing literature exists between microcontroller-class TinyML devices and more capable Single-Board Computers (SBCs).

As shown in Table 1, these device classes differ substantially in memory, compute, and energy availability, which directly dictates the feasibility of on-device training [62].

This hardware heterogeneity has direct implications for FL in localization. Microcontroller-based nodes are typically limited to inference and minimal local updates, whereas more capable edge devices or gate-ways can perform local training and aggregation, enabling hierarchical FL architectures. Recognizing these distinctions is therefore critical for designing realistic, scalable, and energy-efficient TinyML–FL localization systems.

3.2 Cloud-to-Edge Deployment

A prevalent strategy is the cloud-to-edge pipeline, where training occurs on robust cloud infrastructure, and optimized models are deployed to edge devices. Platforms like Databricks, Google Cloud AI, AWS SageMaker, and Microsoft Azure offer GPUs and TPUs for scalable training [18].

Edge-collected data undergoes cloud-based pre-processing, including normalization, augmentation, and filtering. Training employs frameworks like TensorFlow, PyTorch, or Keras [17]. Subsequently, optimizations such as pruning and distillation minimize model complexity without substantial accuracy loss [17]. Models are converted to edge-friendly formats like TensorFlow Lite, ONNX, or TensorRT [19]. Figure 2 demonstrates the deployment of ML model at the edge.

Deployment leverages interpreters like TensorFlow Lite or PyTorch Mobile, augmented by accelerators such as Coral USB or Jetson Nano. This approach harmonizes scalability with efficiency, delivering potent training capabilities alongside low-latency edge inference. Exemplary applications include on-camera object detection for surveillance, predictive maintenance in industrial IoT, and real-time speech recognition in voice assistants [21]. Enhancements in this area include automated model conversion tools to streamline the deployment process.

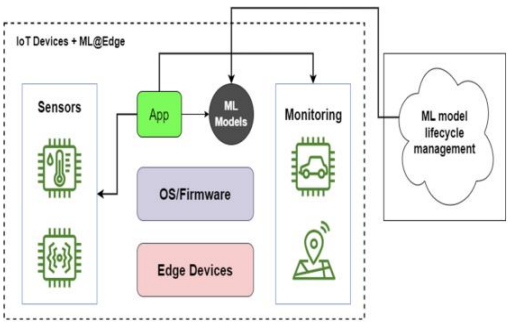

Fig 2. Cloud-based training with optimized edge deployment, adapted from [20].

3.3 Federated Learning Approaches

Federated Learning enables collaborative training across edge devices without centralizing data. In localization contexts, devices can aggregate insights from local RF signals or sensor data to build global models [24].

This is particularly advantageous for TinyML, as it distributes computational load and enhances model robustness through diverse data sources [20].

FL workflows involve local training on devices, followed by aggregation of model updates on a central server. Techniques like FedAvg (Federated Averaging) are commonly used [22]. For resource-constrained settings, optimizations such as secure aggregation and differential privacy are integrated to bolster security and efficiency [23]. Recent advancements include personalized FL, where models are adapted to individual device contexts for better performance [24].

3. APPLICATIONS OF TINYML AND FL IN LOCALIZATION

The convergence of TinyML and Federated Learning (FL) has enabled a new class of intelligent localization systems capable of operating in constrained environments where power, latency, and privacy are critical [4]. By shifting computation to the edge and enabling collaborative learning across devices, these technologies support a wide range of applications—from smart homes to industrial automation—without relying on centralized infrastructure [25]. Below, we explore key use cases with expanded examples and supporting research [3]. Recent studies have demonstrated the effectiveness of TinyML and FL in localization tasks. Avellaneda et al. [32] proposed a deep learning-based TinyML solution for indoor asset tracking, achieving 94% accuracy after post-processing. M´endez et al. [28] compared Wi-Fi RSSI and CSI fingerprinting, showing that CSI outperformed RSSI by 5–6% in single-sample classification and over 50% in time-series scenarios. Emerging research also explores the integration of TinyML and FL with 5G networks, enabling ultra-low-latency and high-precision localization through edge-native architectures and enhanced spectrum sensing [33]. Some applications covered by the literature are presented below.

4.1 Zone Classification

TinyML models can classify physical zones (e.g., kitchen, living room, office) using ambient wireless signals such as Wi-Fi RSSI, Bluetooth Low Energy (BLE), or generic RF patterns [26, 27]. This enhances context awareness in smart environments, allowing systems to adapt behaviour based on user location. For instance, entering a kitchen could trigger recipe suggestions on a smart display, while presence in a bedroom might activate sleep mode on connected devices. FL enables these models to be trained across multiple homes without compromising user privacy, improving generalization across diverse layouts and signal conditions [28].

4.2 Activity-Based Localization

TinyML can leverage inertial measurement unit (IMU) data to infer user activities and movement patterns, such as walking, climbing stairs, or standing still [29]. These insights support pedestrian dead reckoning and indoor navigation, particularly in fitness tracking and elderly care applications [4]. FL allows models to adapt to individual gait patterns and device placements, improving accuracy over time without centralized data collection [3].

4.3 RF Fingerprinting

RF fingerprinting enables anchor-free localization by learning unique signal characteristics at different locations. TinyML models can be trained to recognize these patterns using Wi-Fi RSSI, Channel State Information (CSI), or BLE signals, eliminating the need for fixed infrastructure [28, 30]. FL enhances this approach by aggregating fingerprints from multiple devices, allowing the system to adapt to environmental changes such as furniture rearrangement or signal interference [4]. This is especially useful in dynamic public spaces like airports, shopping malls, or exhibition centers [3].

4.4 Asset Tracking

TinyML-powered asset trackers can monitor the location of equipment, inventory, or medical devices in real time using wireless signals [25, 28]. In industrial settings, this supports logistics optimization, predictive maintenance, and theft prevention [4]. FL enables collaborative learning across devices deployed in different zones, allowing the system to adapt to layout changes, signal obstructions, or usage patterns [3].

4.5 Seamless Indoor–Outdoor Positioning

Combining GNSS (e.g., GPS), Ultra-Wideband (UWB), Wi-Fi, and IMU data with TinyML

allows devices to maintain accurate positioning across indoor and outdoor environments [25, 31]. TinyML models can detect transitions between signal domains and adjust inference strategies accordingly [3]. FL supports the fusion of data from multiple users and devices, refining global positioning models and improving accuracy in challenging environments such as urban canyons, underground facilities, or multi-level buildings [4].

4.6 Sensor Modalities for TinyML and Federated Learning-Based Localization

The effectiveness of the above localization approaches is strongly influenced by the underlying sensing modalities. Therefore, a comparative analysis of commonly used sensor technologies is presented in this subsection.

Different sensing modalities used in localization systems exhibit distinct trade-offs in terms of accuracy, power consumption, and suitability for deployment on resource-constrained edge devices. In the context of TinyML and federated learning (FL), these trade-offs become even more critical, as models must operate under strict memory and energy constraints while supporting distributed training. Table 2 provides a comparative overview of common localization modalities, including Wi-Fi RSSI, channel state information (CSI), Bluetooth Low Energy (BLE), ultra-wideband (UWB), and inertial measurement units (IMU), highlighting their suitability for TinyML and FL-based systems [60,62].

Wi-Fi RSSI and BLE-based approaches are widely adopted due to their low power consumption and compatibility with lightweight models, making them well suited for TinyML deployment [63,64]. In contrast, CSI and UWB-based methods offer higher localization accuracy but require increased computational resources and more complex signal processing pipelines, which may limit their applicability on microcontroller-class devices [64, 66]. IMU-based approaches, while energy-efficient, suffer from drift over time, often requiring sensor fusion or periodic correction [65]. These differences highlight the importance of selecting sensing modalities based on system constraints and application requirements.

Table 2. Comparison of Sensor Modalities for TinyML and Federated Learning-Based Localization under Accuracy, Power, and Privacy Constraints.

Modality	Accuracy	Power	TinyML Suitability	FL Suitability	Privacy
Wi-Fi -RSSI	Medium	Low	High	High	Moderate (location inference risk)
Wi-Fi -CSI	High	Medium	Moderate	High	Low–Moderate (fine-grained signal leakage)
BLE	Medium	Very Low	High	High	Moderate–High (coarse proximity data)
UWB	Very High	Medium	Moderate	High	Low (precise tracking risk)
IMU	Medium (drift-prone)	Very Low	High	High	High (no external signal exposure)

In addition to accuracy and energy considerations, privacy implications vary significantly across sensing modalities, with externally observable signals (e.g., Wi-Fi and UWB) posing higher localization inference risks compared to self-contained sensing such as IMU.

4. BENEFITS OF TINYML AND FL IN LOCALIZATION

The integration of TinyML and FL presents a transformative approach to localization, particularly in resource-constrained and privacy-sensitive environments [4,25]. Their combined strengths enable intelligent, distributed decision-making at the edge, offering a range of benefits:

- ***Low Power Consumption:*** TinyML models are optimized for ultra-low-power devices, allowing inference to run on microcontrollers and sensors with minimal energy usage [21]. This is crucial for battery-operated IoT devices deployed in remote or mobile settings, where frequent recharging or replacement is impractical.
- ***Real-Time Processing:*** By performing inference locally, edge devices can make immediate decisions without waiting for cloud responses [3]. This is essential for time-critical localization tasks such as autonomous vehicle navigation, drone flight control, or emergency response coordination.
- ***Reduced Latency:*** Eliminating the need to transmit raw data to centralized servers significantly reduces communication delays [32]. In localization, this translates to faster position updates and smoother tracking, especially in dynamic environments like smart cities or industrial automation.
- ***Privacy and Security:*** FL ensures that sensitive location data remains on-device, with only model updates shared during training [22]. This decentralized approach aligns with data protection regulations such as GDPR and enhances user trust in systems that handle personal or geospatial information.
- ***Robustness in GNSS-Denied Areas:*** In environments where GNSS signals are weak or unavailable—such as indoors, underground, or in urban canyons—TinyML can leverage sensor fusion (e.g., IMU, Wi-Fi, BLE) to maintain accurate localization [31]. FL further strengthens this by aggregating diverse data sources across devices [4].
- ***Adaptive Learning:*** FL enables models to continuously learn from local conditions without centralized retraining [24]. This adaptability is vital for localization systems operating in changing environments, such as seasonal variations, infrastructure updates, or evolving user behaviour.
- ***Scalability:*** As the number of edge devices grows, FL scales naturally by distributing computation and learning across the network [33]. This decentralized architecture supports large-scale deployments in smart homes, logistics networks, and environmental monitoring systems, without overwhelming central infrastructure.

Collectively, these benefits position TinyML and FL as a powerful combination for building efficient, secure, and context-aware localization systems. Their adoption not only enhances technical performance but also reduces infrastructure costs and regulatory risks, making them ideal for next-generation IoT applications.

5.1 Advances in domain-specific federated learning (FL) for localization

Recent advances in domain-specific federated learning (FL) for localization have focused on architectures tailored to wireless signal characteristics and spatial inference tasks. For instance, hierarchical deep neural network (DNN) models for Wi-Fi RSSI-based localization leverage multi-level feature extraction to improve robustness across heterogeneous indoor environments, while maintaining compatibility with federated training paradigms [60]. In parallel, split and encoder–decoder-based FL approaches, such as FedWiLoc, decompose localization models into shared feature extractors and task-specific decoders, enabling efficient collaboration while preserving device-level personalization [61].

Comparatively, hierarchical DNN-based methods emphasize spatial feature learning and scalability across environments, whereas encoder–decoder FL frameworks prioritize model modularity and communication efficiency by limiting shared parameters. These approaches highlight a broader trend toward architecture-aware FL design, where model structure is explicitly adapted to the constraints and objectives of localization systems.

5.2 Benchmarking of TinyML and Federated Learning for Localization

This section provides a consolidated comparison of representative studies in TinyML and federated learning for localization. This section consolidates reported performance metrics from representative studies cited in this paper.

To systematically connect communication-efficient and heterogeneity-aware federated learning (FL) to edge-based localization, Table 3 summarizes representative methods with respect to their communication reduction capabilities, computational cost, ability to handle device heterogeneity, and overall suitability for edge deployment. These dimensions are critical in IoT localization scenarios, where devices operate under strict bandwidth, energy, and hardware constraints, and where frequent model updates are required to adapt to dynamic environments.

As shown in Table 3, communication-efficient FL methods primarily rely on compression and sparsification techniques to reduce the overhead of model updates. Approaches such as FedSparQ and TEASQ-Fed achieve substantial communication savings while maintaining acceptable computational complexity, making them well suited for resource-constrained edge devices. In contrast, over-the-air aggregation methods, such as AirComp FL [58], significantly reduce latency and bandwidth usage by exploiting the wireless medium, although they offer limited flexibility in handling heterogeneous data and device capabilities.

Notably, methods such as TIFeD and hierarchical FL explicitly address system heterogeneity, which is a defining characteristic of real-world IoT deployments. While these approaches may introduce additional coordination or structural complexity, they enable more robust and scalable training across diverse devices. Overall, the results highlight a fundamental trade-off between communication efficiency, computational over-head, and adaptability to heterogeneous environments, underscoring the need for carefully selecting FL strategies based on the target localization application and deployment constraints.

Table 4 consolidates reported performance metrics from representative studies cited in this paper. The comparison focuses on key deployment-oriented indicators, including model size, inference latency, and energy consumption, which are critical for TinyML-based localization in resource-constrained environments. Where exact values are not explicitly reported, ranges or qualitative estimates are provided based on the described system configurations.

As shown in Table 4, TinyML-based approaches achieve significantly lower latency and energy consumption compared to conventional edge or cloud-based models, often operating within sub-100 ms inference times and sub-megabyte model sizes. However, this efficiency comes at the cost of reduced model capacity, which may impact localization accuracy in complex environments. Federated learning-based approaches mitigate this limitation by enabling collaborative model improvement, although they introduce additional communication overhead.

Table 3. Benchmark Comparison of Communication-Efficient Federated Learning Methods for Edge Systems.

Method	Key Technique	Communication Reduction	Computation Cost	Heterogeneity Handling	Edge Suitability
Quantized Compressed Sensing [55]	Gradient compression + reconstruction	High	Moderate	Limited	Suitable for bandwidth-constrained IoT
FedSparQ [56]	Sparse + quantized updates with error feedback	Very High	Low-Moderate	Moderate	Highly suitable for edge devices
TEASQ-Fed [57]	Adaptive quantization + sparsification	High	Moderate	Moderate	Suitable for heterogeneous TinyML systems
AirComp FL [58]	Over-the-air aggregation	Extremely High	Low (per device)	Limited	Ideal for dense wireless IoT
TIFeD [59]	Integer-only training + feedback alignment	Moderate	Very Low	High	Suitable for microcontroller-class TinyML
Hierarchical FL (Localization) [61]	Multi-level aggregation	High	Moderate	High	Scalable for large IoT deployments

Table 4. Benchmark Summary of TinyML-Based Localization Approaches from Reviewed Literature

Study	Modality	Model	Accuracy	Model Size	Latency	Energy
Avellaneda et al. [32]	IMU / RF	TinyML DNN	~85–90%	< 300 KB	< 50 ms	Low
Suwannaphong et al. [26	Wi-Fi RSSI / CSI	Quantized Transformer	~88–92%	< 500 KB (optimized)	< 100 ms	Low
Liu [31]	UWB + IMU	Sensor fusion ML	< 30 cm error	~1 MB	Low	Medium
Bregar et al. [48]	Wi-Fi RSSI	CNN	~80–90%	~1 MB	Moderate	Moderate
Roy et al. [64]	RSSI vs CSI	TinyML CNN	CSI > RSSI	< 500 KB	Low	Low
Zamzam et al. [63]	Wi-Fi / BLE	ML models	Comparable accuracy	Not reported	BLE < Wi-Fi latency	Low
AlHajri et al. [60]	RF signals	Transfer learning DNN	80–88%	Not reported	Moderate	Moderate
Heydari et al. [62]	Multi-modal	TinyML (CNN/RNN)	70–90%	< 500 KB	< 100 ms	Low

5. CHALLENGES OF TINYML AND FL IN LOCALIZATION

Despite their transformative potential, the deployment of TinyML and Federated Learning (FL) in localization systems faces several technical and operational challenges. These limitations stem from the inherent constraints of edge devices, fragmented development ecosystems, and evolving regulatory

landscapes. To promote practical adoption, this section outlines key challenges along with mitigation strategies that address both engineering and compliance concerns.

- ***Limited Computational Resources:*** Edge devices often operate with constrained memory, processing power, and energy budgets [21]. Running complex models or performing frequent updates can overwhelm these systems. Mitigation includes model compression techniques such as quantization, pruning, and knowledge distillation.
- ***Heterogeneous Hardware and Data:*** Devices differ in architecture, sensor quality, and environmental conditions, leading to inconsistent data distributions [24]. FL must account for non-IID (non-independent and identically distributed) data and device heterogeneity through personalized models and adaptive aggregation strategies.
- ***Communication Overhead:*** FL requires periodic transmission of model updates, which can strain bandwidth and battery life, especially in mobile or remote deployments [22]. Solutions include update sparsification, asynchronous communication, and federated dropout.
- ***Privacy and Security Risks:*** Although FL avoids raw data sharing, model updates can still leak sensitive information [23]. Techniques like secure aggregation, differential privacy, and homomorphic encryption help mitigate these risks.
- ***Lack of Standardization:*** The TinyML and FL ecosystems are fragmented, with diverse frameworks, protocols, and deployment pipelines [33]. This complicates integration and scalability. Emerging standards and cross-platform toolkits aim to unify development practices.
- ***Regulatory Compliance:*** Localization systems must adhere to data protection laws such as GDPR, especially when handling geospatial or biometric data [25]. FL supports compliance by keeping data local, but auditability and transparency remain critical.
- ***Model Drift and Environmental Dynamics:*** Changing layouts, signal interference, and user behaviour can degrade model performance over time [4]. Continuous learning and federated adaptation mechanisms help maintain accuracy in dynamic settings.

5.1 Computational and Memory Limitations

Edge devices such as microcontrollers and embedded sensors often operate with limited memory, processing power, and storage capacity [21]. These constraints restrict the deployment of large or complex machine learning models, which are typically designed for cloud-scale environments. Optimization techniques such as quantization (e.g., int8 or float16), pruning, and knowledge distillation are essential to reduce model size and computational load. However, these techniques may introduce trade-offs between efficiency and accuracy, particularly in high-precision localization tasks [3].

Mitigation: Employ model compression frameworks such as the TensorFlow Model Optimization Toolkit [34] to systematically reduce model complexity while preserving performance. Lightweight architectures like MobileNet and Tiny-YOLO can also be adapted for localization use cases [33].

5.2 Energy Management

Battery-operated edge devices must balance computational demands with energy efficiency to ensure longterm usability, especially in remote or mobile deployments [21, 25]. High-frequency inference or continuous data collection can rapidly deplete power reserves, limiting the practicality of real-time localization [33].

Mitigation: Implement energy-aware techniques such as dynamic voltage and frequency scaling (DVFS), duty cycling, and event-triggered inference [3]. These strategies reduce power consumption by adjusting processing intensity based on contextual needs.

5.3 Toolchain and Standardization Issues

The edge ML ecosystem is fragmented, with a wide variety of hardware platforms, operating systems, and machine learning frameworks [33]. This diversity complicates model portability, integration, and maintenance across devices. Developers often face compatibility issues when deploying models trained in one environment to another [25].

Mitigation: Adopt open standards such as ONNX (Open Neural Network Exchange) to facilitate cross-platform model interoperability [35]. Frameworks like TensorFlow Lite and PyTorch Mobile also support deployment across heterogeneous edge environments [34].

5.4 Communication-Efficient and Heterogeneity-Aware Federated Learning for Edge Localization

Federated Learning (FL) has emerged as a promising paradigm for collaborative model training across distributed edge devices while preserving data privacy. However, its direct application to localization in IoT environments introduces two fundamental challenges: communication overhead and system heterogeneity. These challenges are particularly critical in edge-based localization systems, where devices operate under strict constraints on bandwidth, energy, and computational resources.

5.4.1 Communication Efficiency in Federated Learning

In conventional FL frameworks such as FedAvg, devices periodically transmit model updates to a central server for aggregation. While effective in homogeneous and high-bandwidth settings, this approach becomes impractical in large-scale IoT deployments due to:

- Limited uplink bandwidth
- Energy constraints of battery-powered devices
- High communication frequency in dynamic localization tasks

To address these limitations, communication-efficient FL techniques have been proposed, focusing on reducing the size and frequency of transmitted updates.

Mitigations:

Gradient Compression and Sparsification

Gradient compression techniques reduce communication cost by transmitting only a subset of model updates or by encoding them in a compact form. Methods such as FedSparQ employ sparsification strategies combined with quantization to significantly reduce communication payload while preserving convergence properties [56]. In addition, quantized compressed sensing approaches have been shown to further improve communication efficiency by enabling accurate gradient reconstruction from compressed updates [55]. These approaches are particularly beneficial in localization tasks where frequent updates are required to adapt to environmental changes.

Quantization-Aware Federated Learning

Quantization techniques reduce the precision of model parameters (e.g., from 32-bit floating point to 8-bit integers), thereby lowering transmission size and memory footprint. Advanced methods such as TEASQ-Fed integrate task-adaptive quantization strategies into the federated training loop, dynamically adjusting precision levels based on device capabilities and task requirements [57]. This is especially relevant for TinyML-based localization, where models must fit within strict memory budgets.

Over-the-Air Aggregation

Over-the-air computation (AirComp) techniques, such as AirComp-based federated learning [58], exploit the superposition property of wireless channels to aggregate model updates directly in the air. This eliminates the need for sequential communication with individual devices, significantly reducing latency and band-width usage. In dense IoT localization scenarios—such as smart buildings or industrial environments—this approach enables scalable

and low-latency model aggregation. Hybrid approaches, such as TCS-H (Time-Correlated Sparsification with Hierarchical aggregation), further enhance communication efficiency through sparsification and hierarchical coordination.

5.4.2 Implications for Localization Systems

The integration of communication-efficient and heterogeneity-aware FL techniques is essential for enabling
scalable and adaptive localization systems. Specifically:

- Dynamic environments (e.g., indoor localization with changing layouts) benefit from frequent but lightweight updates enabled by compression and quantization.
- Large-scale deployments (e.g., smart cities, industrial IoT) require hierarchical and over-the-air aggregation to maintain scalability.
- Multi-modal localization systems (e.g., combining IMU, Wi-Fi, and UWB) demand heterogeneity-aware learning to accommodate diverse data characteristics.

By systematically incorporating these techniques, federated TinyML systems can achieve a balance between communication efficiency, model accuracy, and energy consumption, which is critical for real-world edge localization applications.

5.5 Heterogeneity and Convergence in Federated Learning

One of the fundamental challenges in FL is the heterogeneity of data across participating devices. In real-world localization scenarios, each edge device may collect data from vastly different environments—urban vs. rural, indoor vs. outdoor, or static vs. mobile contexts. This leads to non-independent and identically distributed (non-IID) data, which complicates model convergence and can degrade overall performance [22, 24].

Moreover, communication overheads pose additional barriers. Frequent synchronization between devices and the central server can be costly in terms of bandwidth and energy, especially in large-scale deployments [41]. These issues are exacerbated when devices operate asynchronously or intermittently due to connectivity constraints [23].

Mitigation: To address data heterogeneity, algorithms such as FedProx have been developed to improve convergence by introducing a proximal term that stabilizes local updates [42]. Additionally, communication-efficient FL techniques such as gradient sparsification, quantized updates, and over-the-air aggregation have been proposed to mitigate bandwidth and energy constraints in edge environments. These techniques can be particularly valuable in localization systems where devices vary in capability, connectivity, and data quality.

5.5.1 Heterogeneity-Aware Federated Learning

Edge localization systems are inherently heterogeneous, with variations in:

- Device hardware (microcontrollers vs. edge GPUs)
- Sensor modalities (RSSI, CSI, IMU, UWB)
- Data distributions (non-IID environments)

Such heterogeneity can degrade model convergence and performance in standard FL frameworks.

Personalization and Adaptive Aggregation

To address non-IID data distributions, personalized FL approaches allow devices to maintain partially customized models. Techniques such as adaptive weighting and clustering-based aggregation improve model generalization across diverse environments, which is crucial for localization tasks spanning

different physical layouts and signal conditions [59].

Hierarchical Federated Learning

Hierarchical FL (HFL) introduces intermediate aggregation layers such as edge gateways or cluster heads between devices and the central server [52,53]. This architecture offers several advantages:

- Reduces communication frequency and bandwidth usage
- Enables localized model adaptation within clusters
- Improves scalability in large deployments

In localization systems, HFL is particularly effective when devices are naturally grouped (e.g., rooms, floors, or buildings), allowing models to capture spatially correlated patterns while still contributing to a global model.

5.6 Accuracy–Efficiency Trade-offs and Hierarchical Federated Learning Architectures

Ultra-low-power edge devices employed in TinyML deployments impose strict constraints on memory, computational capacity, and energy consumption. These constraints necessitate the use of model compression and quantization techniques to enable on-device inference while maintaining acceptable performance levels [44,45].

Post-training quantization, particularly the conversion from floating-point to 8-bit integer representations, typically results in a modest reduction in model accuracy, often in the range of 1–3% for classification tasks, while yielding substantial improvements in memory footprint, inference latency, and energy efficiency [44,46]. However, more aggressive optimization strategies—such as structured pruning, mixed-precision quantization, or extreme model compression—can introduce larger accuracy degradation. This effect is especially pronounced in regression-based localization models, where small numerical perturbations may propagate nonlinearly and lead to meter-level positioning errors [47–49]. Quantization-aware training (QAT) partially mitigates these effects by incorporating quantization noise during training, thereby improving robustness compared to post-training quantization alone [46].

Federated learning (FL) has emerged as a promising paradigm for collaborative model training across distributed edge devices while preserving data privacy by keeping raw data localized [50, 51]. Despite its advantages, conventional FL approaches such as FedAvg incur significant communication overhead due to frequent global model aggregation, which becomes a scalability bottleneck in dense IoT localization deployments [50].

Hierarchical federated learning (HFL) architectures address this limitation by introducing intermediate aggregation layers, such as edge gateways or cluster heads, between local devices and the central server [52,53]. By aggregating local updates at intermediate nodes before forwarding them upstream, HFL significantly reduces communication frequency and bandwidth requirements while preserving convergence stability. This hierarchical paradigm is particularly well suited for large-scale IoT localization systems, where devices exhibit heterogeneous computational resources, non-identically distributed data, and intermittent connectivity [51, 53].

From a regulatory perspective, the combination of on-device TinyML and hierarchical FL aligns well with the principles of the EU AI Act, including data minimization, robustness, and accountability. By reducing centralized data exposure and enabling localized risk management, these architectures support the development of trustworthy and compliant AI-enabled localization systems [54]. Overall, balancing accuracy degradation introduced by TinyML optimizations with communication-efficient hierarchical learning architectures remains a critical research direction for

scalable, privacy-preserving, and energy-efficient localization in future IoT systems.

5.7 Security and Privacy Vulnerabilities

Edge-deployed machine learning models are inherently exposed to security threats due to their decentralized nature and limited computational safeguards [22, 23]. Common vulnerabilities include adversarial attacks, model inversion, and data extraction. Federated Learning, while designed to preserve privacy by avoiding raw data transmission, introduces new risks such as model poisoning and gradient leakage [24].

In the context of the EU AI Act which enforces strict requirements for high-risk AI systems, including transparency, robustness, and cybersecurity [37,38], addressing these vulnerabilities is both a technical and legal imperative [36]. The Act emphasizes privacy-by-design, accountability, and risk mitigation, particularly for systems involved in sensitive tasks like localization [39,40].

Mitigation: Integrate advanced privacy-preserving techniques such as homomorphic encryption to enable secure computation on encrypted data, and federated differential privacy to protect individual data contributions during collaborative training [23]. These methods align with the EU AI Act's provisions on data governance and system resilience. Table 5 summarizes key mitigation techniques and maps them to relevant articles of the EU AI Act [37,40].

Figure 3 presents a comprehensive ethical framework that incorporates transparency, accountability, and human-centric principles into AI system development [43]. The figure illustrates how existing risks (dashed lines on the left) should be addressed and replaced by essential requirements (solid lines on the right).

Table 5. Security and Privacy Mitigation Techniques in Compliance with the EU AI Act.

Mitigation Technique	Description	Relevant EU AI Act Provision
Homomorphic Encryption	Enables computation on encrypted data without decryption, preserving confidentiality during model training and inference.	Article 15: Robustness and Accuracy
Federated Differential Privacy	Adds noise to local model updates to prevent leakage of individual data points during aggregation.	Article 10: Data Governance
Secure Aggregation Protocols	Ensures that model updates from multiple clients are aggregated securely without exposing individual contributions.	Article 9: Risk Management System
Adversarial Robustness Testing	Evaluates model resilience against adversarial inputs and poisoning.	Article 15: Robustness and Accuracy
Model Explainability Tools	Enhances transparency by making model decisions interpretable to users and auditors.	Article 13: Transparency
Access Control and Audit Logging	Restricts unauthorized access and tracks system interactions to ensure accountability.	Article 14: Human Oversight

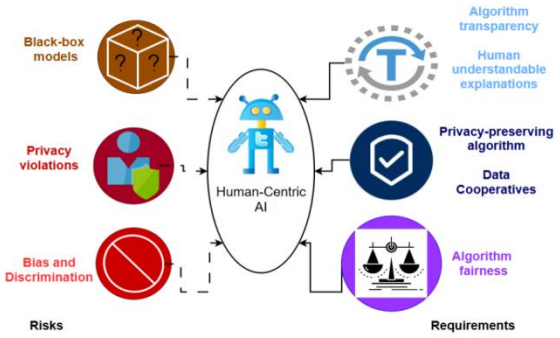

Fig 3. Human-centric AI, adapted from [43].

To ensure that AI systems contribute positively to society, it is important to embed ethical guidelines throughout the AI lifecycle—from problem formulation and data collection to deployment and monitoring. This includes promoting transparency, implementing fairness-aware algorithms, safeguarding user privacy, and establishing clear accountability for outcomes generated by AI systems.

6. CONCLUSION

The convergence of TinyML and Federated Learning (FL) represents a significant advancement in the design and deployment of localization systems, enabling intelligent, privacy-aware, and energy-efficient processing directly on edge devices. This paradigm supports scalable and adaptive localization solutions across a wide range of application domains, including smart cities, industrial IoT, and context-aware environments.

This survey has examined key localization applications, such as RF fingerprinting, asset tracking, and seamless indoor–outdoor positioning, demonstrating how the integration of TinyML and FL enhances both performance and deployment feasibility. In addition, the paper has analyzed critical challenges, including resource limitations, system and data heterogeneity, and regulatory considerations, particularly in the context of privacy-preserving and human-centric AI frameworks.

Overall, the findings underscore that while TinyML and FL offer substantial benefits for decentralized localization, their practical realization requires careful co-design of models, communication strategies, and system architectures to balance accuracy, efficiency, and scalability.

7. FUTURE RESEARCH DIRECTIONS

Future research should further investigate the integration of TinyML and Federated Learning (FL) for RF fingerprinting-based localization, particularly in the context of ultra-low-cost IoT devices. In such settings, devices can perform lightweight feature extraction from RSSI or CSI measurements and collaboratively contribute to global model updates through federated optimization [4,25,28]. This approach offers significant potential in terms of scalability, energy efficiency, robustness to environmental dynamics, and inherent privacy preservation.

Despite these advantages, several open challenges remain. Communication overhead continues to be a critical bottleneck, especially in large-scale deployments with frequent model updates. In addition, the presence of non-IID data across devices and the need for robust security mechanisms introduce further complexity in real-world scenarios [24,41]. Addressing these issues requires the development of more adaptive and communication-efficient FL strategies tailored to resource-constrained environments.

Emerging directions include the exploration of hierarchical and hybrid FL architectures, advanced aggregation techniques, and tighter integration with next-generation edge hardware platforms [3]. Furthermore, enhancing system resilience through quantum-resistant security mechanisms [38] and leveraging automated, AI-driven optimization for TinyML models [33] represent promising avenues for improving both performance and reliability.

Looking ahead, the co-design of federated learning algorithms, TinyML model architectures, and edge system constraints will be essential to enable truly adaptive and scalable localization systems. Such advances will play a central role in supporting the next

generation of context-aware, privacy-preserving, and intelligent IoT applications.

REFERENCES

[1] Abreha, H. G., Hayajneh, M., & Serhani, M. A. (2022). Federated Learning in Edge Computing: A Systematic Survey. Sensors, 22(2), 450.

[2] Mughal, F. R., He, J., Das, B., et al. (2024). Adaptive Federated Learning for Resource-Constrained IoT Devices through Edge Intelligence and Multi-Edge Clustering. Scientific Reports, Nature Publishing Group.

[3] Ren, H., Anicic, D., & Runkler, T. A. (2023). TinyReptile: TinyML with Federated Meta-Learning. In International Joint Conference on Neural Networks (IJCNN). arXiv:2304.05201. https://doi.org/10.48550/arXiv.2304.05201

[4] Ficco, M., Guerriero, A., Milite, E., Palmieri, F., Pietrantuono, R., & Russo, S. (2024). Federated Learning for IoT Devices: Enhancing TinyML with On-Board Training. Information Fusion, 104, 102189. https://doi.org/10.1016/j.inffus.2023.102189

[5] Etiabi, Y., Njima, W., & Amhoud, E. M. (2024). FeMLoc: Federated Meta-learning for Adaptive Wireless Indoor Localization Tasks in IoT Networks. arXiv preprint arXiv:2405.11079. https://doi.org/10.48550/arXiv.2405.11079

[6] Ye, F., Wang, R., Tang, S., Duan, S., & Xu, C. (2023). Federated Learning-Enabled Cooperative Localization in Multi-agent System. International Journal of Wireless Information Networks, 31, 61–72. https://doi.org/10.1007/s10776-023-00610-0

[7] Dritsas, E., & Trigka, M. (2025). Federated Learning for IoT: A Survey of Techniques, Challenges, and Applications. Journal of Sensor and Actuator Networks, 14(1),9. https://doi.org/10.3390/jsan14010009

[8] R. Shahbazian, G. Macrina, E. Scalzo, and F. Guerriero," Machine Learning Assists IoT Localization: A Review of Current Challenges and Future Trends," Sensors 2023, 23, 3551. https://doi.org/10.3390/s23073551.

[9] Filho, C. P., Marques Jr., E., Chang, V., dos Santos, L., Bernardini, F., Pires, P. F., Ochi, L., & Delicato, F. C. (2022). A Systematic Literature Review on Distributed Machine Learning in Edge Computing. Sensors, 22(7), 2665. https://doi.org/10.3390/s22072665

[10] Truong, H.-L., Truong-Huu, T., & Cao, T.-D. (2022). Making Distributed Edge Machine Learning for Resource-Constrained Communities and Environments Smarter: Contexts and Challenges. Journal of Reliable Intelligent Environments, 9, 119–134. https://doi.org/10.1007/s40860-022-00176-3

[11] Ray, P. P. (2022). A review on TinyML: State-of-the-art and prospects. Journal of King Saud University- Computer and Information Sciences, 34(4), 1595–1623. https://doi.org/10.1016/j.jksuci.2021.11.019

[12] Khouas, A. R., Bouadjenek, M. R., Hacid, H., & Aryal, S. (2024). Training Machine Learning Models at the Edge: A Survey. arXiv preprint arXiv:2403.02619. https://arxiv.org/pdf/2403.02619

[13] Patil, S. G. (2024). Systems for Machine Learning on Edge Devices. Master's thesis, University of California, Berkeley. EECS Department Technical Report No. UCB/EECS-2024-1. https://www2.eecs.berkeley.edu/Pubs/TechRpts/2024/EECS-2024-1.pdf

[14] Amaral, R., Murthy, A., Stronach, M., Jain, K., Khurmi, K., & Murthy, P. (2024). Train, Optimize, and Deploy Models on Edge Devices Using Amazon SageMaker and Qualcomm AI Hub. AWS Machine Learning Blog. https://aws.amazon.com/blogs/machine-learning/train-optimize-and-deploy-models-on-edge-devices-using-amazon-sagemaker-and-qualcomm-ai-hub/

[15] Qualcomm Developer Blog, Optimizing Your AI Model for the Edge, June 2025. Available: https://www.qualcomm.com/developer/blog/2025/06/optimizing-your-ai-model-for-the-edge

[16] Google AI for Developers, Model Optimization — Google AI Edge, 2025. Available: https://ai.google.dev/edge/litert/models/model_optimization

[17] Xubin Wang and Weijia Jia, Optimizing Edge AI: A Comprehensive Survey on Data, Model, and System Strategies, arXiv preprint arXiv:2501.03265, Jan. 2025. Available: https://arxiv.org/abs/2501.03265

[18] Infoservices Team, Edge AI + Multi-Cloud: The Future of Smart Infrastructure, May 2025. Available: https://blogs.infoservices.com/embedded-system/edge-ai-multi-cloud-smart-infrastructure/

[19] Mahesh Vaijainthymala Krishnamoorthy, Edge AI: TensorFlow Lite vs. ONNX Runtime vs. PyTorch Mobile, DZone, June 2025. Available: https://dzone.com/articles/edge-ai-tensorflow-lite-vs-onnx-runtime-vs-pytorch

[20] Dinesh Subramani and Samir Araujo, Demystifying Machine Learning at the Edge Through Real Use Cases, June 2022. Available: https://aws.amazon.com/blogs/machine-learning/demystifying-machine-learning-at-

the-edge-through-real-use-cases/. Accessed: 2025-09-24.

[21] Vivienne Sze, Yu-Hsin Chen, Tien-Ju Yang, and Joel S. Emer, "Efficient Processing of Deep Neural Networks: A Tutorial and Survey," Proceedings of the IEEE, vol. 105, no. 12, pp. 2295–2329, 2017. doi: 10.1109/JPROC.2017.2761740.

[22] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in Proc. of AISTATS, 2017.

[23] Mikko A. Heikkil¨a, "On Using Secure Aggregation in Differentially Private Federated Learning with Multiple Local Steps," arXiv preprint arXiv:2407.19286, Mar. 2025. Available: https://arxiv.org/html/2407.19286v2

[24] Bifta Sama Bari and Kumar Yelamarthi, "Advancing Federated Learning: A Comprehensive Solution for Model Aggregation, Heterogeneity, Privacy, and Security," SN Computer Science, vol. 6, article no. 411, Apr. 2025. Available: https://link.springer.com/article/10.1007/s42979-025-03934-1

[25] Leandro Antonio Pazmi˜no Ortiz, Ivonne Fernanda Maldonado Soliz, and Vanessa Katherine Guevara Balarezo, "Advancing TinyML in IoT: A Holistic System-Level Perspective for Resource-Constrained AI," Future Internet, vol. 17, no. 6, article 257, June 2025. doi: 10.3390/fi17060257.

[26] Thanaphon Suwannaphong, Ferdian Jovan, Ian Craddock, and Ryan McConville, "Optimising TinyML with Quantization and Distillation of Transformer and Mamba Models for Indoor Localisation on Edge Devices," arXiv preprint arXiv:2412.09289, Dec. 2024. Available: https://arxiv.org/pdf/2412.09289

[27] B. Yuen, Y. Bie, D. Cairns, G. Harper, J. Xu, C. Chang, X. Dong, and T. Lu, "Wi-Fi and Bluetooth Contact Tracing Without User Intervention," IEEE Access, vol. 10, pp. 91027–91044, 2022. doi: 10.1109/ACCESS.2022.3201645.

[28] Marco Zennaro, Diego M´endez, Moez Altayeb, Daniel Crovo, and Gianpaolo Manzoni, "Indoor Positioning Systems: Edge-Based RF Fingerprinting," SciTinyML Workshop, Harvard University, May 2024. Available: https://tinyml.seas.harvard.edu/SciTinyML-24/assets/slides/5-Diego-Location-Fingerprinting.pdf

[29] Jared Maks, TinyML on the Edge: IMU Sensors on Arduino Nano 33 BLE Sense, GitHub repository, 2025. Available: https://github.com/jaredmaks/tinyml-on-the-edge

[30] F. Zafari, A. Gkelias, and K. K. Leung, "A Survey of Indoor Localization Systems and Technologies," IEEE Communications Surveys & Tutorials, vol. 21, no. 3, pp. 2568–2599, 2019. doi: 10.1109/COMST.2019.2911558

[31] Su Liu, Seamless Indoor–Outdoor Positioning Using TinyML and Sensor Fusion, MASc Thesis, McMaster University, April 2023. Available: https://macsphere.mcmaster.ca/bitstream/11375/28513/2/Liu_Su_202304_MASc.pdf

[32] D. Avellaneda, D. M´endez, and G. Fortino, "A TinyML Deep Learning Approach for Indoor Tracking of Assets," Sensors, vol. 23, no. 3, article 1542, Jan. 2023. doi: 10.3390/s23031542

[33] Rakhee Kallimani, Krishna Pai, Prasoon Raghuwanshi, Sridhar Iyer, and Onel L. A. L´opez, "TinyML: Tools, Applications, Challenges, and Future Research Directions," Multimedia Tools and Applications, 2023. doi: 10.1007/s11042-023-16740-9

[34] TensorFlow Team, TensorFlow Model Optimization Toolkit, 2025. Available: https://www.tensorflow.org/model_optimization

[35] ONNX Community, Open Neural Network Exchange (ONNX), 2025. Available: https://onnx.ai

[36] European Commission, Regulation of the European Parliament and of the Council Laying Down Harmonised Rules on Artificial Intelligence (Artificial Intelligence Act), 2024. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A52021PC0206

[37] European Parliament, "EU AI Act: First regulation on artificial intelligence," 2024. Available: https://www.europarl.europa.eu/topics/en/article/20230601STO93804/eu-ai-act-first-regulation-on-artificial-intelligence. Accessed: 2025-05-28.

[38] Robin Staab, Mark Vero, Mislav Balunovi´c, and Martin Vechev, "Beyond Memorization: Violating Privacy Via Inference with Large Language Models," arXiv preprint arXiv:2310.07298, 2024. Available: https://arxiv.org/abs/2310.07298

[39] Donghyeok Lee, Christina Todorova, and Alireza Dehghani, "Ethical Risks and Future Direction in Building Trust for Large Language Models Application under the EU AI Act," in Proc. of the 2024 Conference on AI Regulation, Dec. 2024, pp. 41–46. doi: 10.1145/3701268.3701272

[40] European Commission, "General-Purpose AI Code of Practice now available," 2025. Available: https://ec.europa.eu/commission/presscorner/detail/en/ip_25_1787. Accessed: 2025-07-26.

[41] Peter Kairouz et al., "Advances and Open Problems in Federated Learning," Foundations and Trends in Machine Learning, vol. 14, no. 1–2, pp. 1–210, 2021. doi: 10.1561/2200000083

[42] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith, "Federated Optimization in Heterogeneous Networks," in Proc. of MLSys, 2020. Available: https://arxiv.org/abs/1812.06127

[43] Bruno Lepri, Nuria Oliver, and Alex Pentland, "Ethical machines: The human-centric use of artificial intelligence," iScience, vol. 24, no. 3, article 102249, 2021. doi: 10.1016/j.isci.2021.102249. Available: https://www.sciencedirect.com/science/article/pii/S2589004221002170

[44] C. R. Banbury, V. J. Reddi, P. T. Chaudhary, et al., "Benchmarking TinyML Systems: Challenges and Direction," arXiv preprint arXiv:2102.07638, pp. 1–13, 2021.

[45] Kunran Xu, Huawei Zhang, Yishi Li, Yuhao Zhang, Rui Lai, Yi Liu, "An Ultra-low Power TinyML System for Real-time Visual Processing at Edge," arXiv:2207.04663, 2023.

[46] B. Jacob, S. Kligys, B. Chen, et al., "Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference," arXiv:1712.05877, 2017.

[47] S. Han, H. Mao, and W. J. Dally, "Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding," arXiv:1510.00149, 2016.

[48] K. Bregar and M. Mohorˇciˇc," Improving Indoor Localization Using Convolutional Neural Networks on Computationally Restricted Devices," in IEEE Access, vol. 6, pp. 17429-17441, 2018, doi: 10.1109/ACCESS.2018.2817800, url: https://ieeexplore.ieee.org/document/8320781

[49] F. Zafari, A. Gkelias, and K. K. Leung, "A Survey of Indoor Localization Systems and Technologies," IEEE Communications Surveys & Tutorials, vol. 21, no. 3, pp. 2568–2599, 2019.

[50] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS), pp. 1273–1282, 2017.

[51] P. Kairouz, H. B. McMahan, B. Avent, et al., "Advances and Open Problems in Federated Learning," arXiv:1912.04977 10.1561/9781680837896, 2021.

[52] C. Briggs, Z. Fan, and P. Andras, "Federated Learning with Hierarchical Clustering of Local Updates to Improve Training on Non-IID Data," Proceedings of the International Joint Conference on Neural Networks (IJCNN), pp. 1–9, 2020.

[53] Qiu, C., Wu, Z., Wang, H., Yang, Q., Wang, Y., Su, C. (2025). Hierarchical Aggregation for Federated Learning in Heterogeneous IoT Scenarios: Enhancing Privacy and Communication Efficiency. Future Internet, 17(1), 18. https://doi.org/10.3390/fi17010018

[54] European Parliament and Council of the European Union, "Regulation (EU) 2024/1689 Laying Down Harmonised Rules on Artificial Intelligence (Artificial Intelligence Act)," Official Journal of the European Union, 2024. https://eur-lex.europa.eu/EN/legal-content/summary/rules-for-trustworthy-artificial-intelligence-in-the-eu.html

[55] Y. Oh, N. Lee, Y.-S. Jeon, and H. V. Poor, "Communication-Efficient Federated Learning via Quantized Compressed Sensing," IEEE Transactions on Wireless Communications, vol. 22, no. 2, pp. 1087–1100, 2023, doi: 10.1109/TWC.2022.3201207.

[56] C Medjadji, S Alawadi, F M. Awaysheh, G Leduc, S Kubler, Y Le Traon, "FedSparQ: Adaptive Sparse Quantization with Error Feedback for Robust & Efficient Federated Learning," arXiv:2511.05591, 2023, doi: https://doi.org/10.48550/arXiv.2511.0559.

[57] J. Jia, J. Liu, C. Zhou, H. Tian, M. Dong, and D. Dou, "Efficient Asynchronous Federated Learning with Sparsification and Quantization," arXiv preprint arXiv:2312.15186, 2023. [Online]. Available: https://arxiv.org/abs/2312.15186

[58] K. Yang, T. Jiang, Y. Shi, and Z. Ding, "Federated Learning via Over-the-Air Computation," arXivpreprint arXiv:2202.08420, 2022. [Online]. Available: https://arxiv.org/abs/2202.08420

[59] L. Colombo, A. Falcetta, and M. Roveri, "TIFeD: A Tiny Integer-Based Federated Learning Algorithm with Direct Feedback Alignment," arXiv preprint arXiv:2411.06042, 2024. [Online]. Available: https://arxiv.org/abs/2411.06042

[60] M. I. AlHajri, R. M. Shubair, and M. Chafii, "Indoor Localization Under Limited Measurements: A Cross-Environment Joint Semi-Supervised and Transfer Learning Approach," in Proc. IEEE 22nd Int. Workshop Signal Process. Adv. Wireless Commun.

(SPAWC), 2021, pp. 266–270, doi: 10.1109/SPAWC51858.2021.9593245.

[61] Z. Li, Y. Chen, X. Wang, and H. Zhang, “Hierarchical Federated Learning for Robust and Communication-Efficient Indoor Localization,” arXiv preprint arXiv:2512.18207, 2025. [Online]. Available: https://arxiv.org/abs/2512.18207

[62] S. Heydari and Q. H. Mahmoud, “Tiny Machine Learning and On-Device Inference: A Survey of Applications, Challenges, and Future Directions,” Sensors, vol. 25, no. 10, p. 3191, 2025. https://doi.org/10.3390/s25103191

[63] M. Zamzam, Y. Ashraf, T. Elshabrawy and M. Ashour, "Accuracy and Latency Tradeoffs for WiFi and BLE in an Indoor Localization System," 2022 IEEE 2nd International Conference on Electronic Technology, Communication and Information (ICETCI), Changchun, China, 2022, pp. 81-85, doi: 10.1109/ICETCI55101.2022.9832290.

[64] Mendez, D.; Zennaro, M.; Altayeb, M.; Manzoni, P. On TinyML WiFi fingerprinting-based indoor localization: Comparing RSSI vs. CSI utilization. In Proceedings of the 2024 IEEE 21st Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 6–9 January 2024; pp. 1–6.

[65] A. Nessa, B. Adhikari, F. Hussain, and X. N. Fernando, “A Survey of Machine Learning for Indoor Positioning,” IEEE Access, vol. 8, pp. 214945–214965, 2020.

[66] Leitch, S. G., Ahmed, Q. Z., Abbas, W. B., Hafeez, M., Lazaridis, P. I., Sureephong, P., & Alade, T. (2023). On Indoor Localization Using WiFi, BLE, UWB, and IMU Technologies. Sensors, 23(20), 8598. https://doi.org/10.3390/s23208598