ISSN: 2181-3337 | SCIENTISTS.UZ VOLUME 5 ISSUE 9 | SEPTEMBER 2026

FEDERATED TINYML FOR DISTRIBUTED IOT SYSTEMS: A PRIVACY-PRESERVING AND ENERGY-EFFICIENT EDGE INTELLIGENCE FRAMEWORK

Mahmudov Mahmud[1], **Rixsiboyev Ravshan**[2]

Tashkent State Technical University named after Islam Karimov
Tashkent, Uzbekistan

DOI: 10.67620/sai.59a0011

ABSTRACT

The rapid expansion of the Internet of Things (IoT) has increased demand for machine learning approaches that can operate efficiently on resource-constrained edge devices while protecting locally generated data. This paper proposes a Federated Tiny Machine Learning (TinyML) framework for distributed IoT systems that combines on-device inference and lightweight local training with federated model aggregation. In the proposed architecture, ESP32-class microcontrollers collect and preprocess sensor data, execute TensorFlow Lite for Microcontrollers models, and transmit model updates rather than raw observations to a coordinating server. The server applies Federated Averaging to produce a global model that is redistributed to participating devices. The framework is organized into sensor, edge, federated aggregation, and application layers and is designed to address communication overhead, privacy exposure, heterogeneous non-independent and identically distributed data, intermittent connectivity, and limited energy and memory resources. A representative hardware and sensor configuration is specified to demonstrate implementation feasibility. The proposed design emphasizes data minimization, communication efficiency, scalability, and continuous learning at the edge. As a conceptual architecture, it provides a foundation for future empirical evaluation of accuracy, communication cost, energy consumption, convergence behavior, and robustness across heterogeneous IoT deployments.

ACCEPTED

06.10.26

Keywords: *federated learning, TinyML, edge AI, Internet of Things, privacy-preserving machine learning, edge computing.*

CC BY

Introduction

The Internet of Things (IoT) has become a foundational technology for digital transformation in industrial automation, precision agriculture, healthcare, smart transportation, environmental monitoring, and smart cities. Advances in embedded systems and wireless communication allow large numbers of IoT devices to collect, process, and exchange heterogeneous sensor data continuously. This growth creates challenges related to data processing, communication efficiency, scalability, security, and privacy.

Traditional IoT systems commonly rely on cloud-centric architectures in which sensor data are transmitted to centralized servers for storage, analysis, and machine learning. Although cloud platforms provide substantial computational capacity, continuous transmission of raw sensor data increases network traffic, communication latency, and energy consumption and may expose sensitive information. These

limitations are especially important in real-time applications and in environments with unstable or limited Internet connectivity.

Edge artificial intelligence (Edge AI) addresses some of these limitations by moving inference closer to the data source. Tiny Machine Learning (TinyML) extends this approach to low-power microcontrollers with constrained memory and processing capacity. Lightweight models can be deployed on embedded devices using frameworks such as TensorFlow Lite for Microcontrollers (Warden & Situnayake, 2019). Embedded IoT platforms have also been used in remote monitoring, water-control, parking, visualization, and industrial automation applications (Eltazarov et al., 2024; Gaibnazarov et al., 2024; Gulyamova et al., 2024; Makhmudov et al., 2023, 2024).

Despite the advantages of on-device inference, many TinyML implementations still depend on centralized model training. Aggregating data from distributed devices can increase communication overhead and privacy risk and may perform poorly when local data distributions differ substantially across locations and devices.

Federated learning (FL) provides a distributed alternative. Each participating client trains a local model on its own data and transmits model updates to a coordinating server. The server combines the updates, commonly using Federated Averaging (FedAvg), to construct a global model (McMahan et al., 2017). Keeping raw data decentralized can reduce direct data exposure, although federated learning does not by itself guarantee privacy and may require additional safeguards (Kairouz et al., 2021).

Integrating FL with TinyML is challenging because microcontrollers have limited RAM, flash memory, processing capability, battery capacity, and communication bandwidth. Practical Federated TinyML therefore requires lightweight model architectures, efficient communication, selective client participation, and energy-aware optimization. This paper presents a conceptual Federated TinyML framework for distributed IoT systems that emphasizes communication efficiency, energy awareness, scalability, and data minimization.

Literature Review

TinyML enables machine learning inference on microcontrollers and other highly resource-constrained embedded platforms. Warden and Situnayake (2019) describe workflows for training, converting, and deploying compact models on embedded devices using TensorFlow Lite for Microcontrollers. This approach supports local inference without continuous dependence on cloud computation.

Federated learning was formalized as a practical approach to decentralized model training by McMahan et al. (2017), who introduced FedAvg and demonstrated that iterative model averaging can reduce communication rounds under decentralized and non-IID data conditions. Subsequent work has identified communication efficiency, statistical heterogeneity, privacy, security, client availability, and systems constraints as central open problems in federated learning (Kairouz et al., 2021).

The intersection of TinyML and federated learning is particularly relevant to distributed IoT networks, where clients may have intermittent connectivity and strongly heterogeneous sensing conditions. A Federated TinyML design must therefore balance model quality with memory use, computation, communication frequency, and energy consumption. The present study focuses on a lightweight architecture for ESP32-class devices and a central aggregation server.

ISSN: 2181-3337 | SCIENTISTS.UZ

Table 1

Conceptual Comparison of Learning Paradigms

Feature	Cloud AI	TinyML	Federated Learning	Federated TinyML
Local inference	No	Yes	Possible	Yes
Centralized raw data	Typically	Training often centralized	No	No
Privacy exposure	Higher	Moderate	Lower*	Lower*
Communication demand	High	Low for inference	Model-update dependent	Optimized for model updates
Energy efficiency at edge	Low	High	Device dependent	Design objective
Scalability	Moderate	High	High	High
Real-time local processing	Limited	Good	Possible	Good
Continuous collaborative learning	No	Limited	Yes	Yes

Note. *Lower exposure refers to keeping raw training data on clients; federated learning alone does not provide a formal privacy guarantee. The table is a conceptual comparison proposed in this paper, not an empirical benchmark.

Proposed Federated TinyML Architecture

The proposed architecture is designed for distributed IoT systems in which raw sensor data remain on local devices. Each IoT node performs data acquisition, preprocessing, TinyML inference, and lightweight local model updates. Only model updates are communicated to the aggregation server.

The architecture contains four layers: sensor, edge, federated aggregation, and application. The sensor layer collects environmental or industrial measurements such as temperature, humidity, pressure, vibration, gas concentration, acceleration, and angular velocity. The edge layer contains resource-constrained microcontrollers, such as the ESP32, running a TinyML runtime. The federated aggregation layer receives local model updates and combines them using FedAvg. The application layer provides monitoring, visualization, analytics, and decision support.

Table 2

Proposed System Architecture

Layer	Components	Primary function
Application	Dashboard, mobile app, web server	Visualization and decision support
Federated aggregation	FedAvg server	Global model aggregation
Edge	ESP32 + TinyML runtime	Local training and inference
Sensor	DHT22, MPU6050, BME280, MQ135	Data acquisition

Note. The listed components represent a reference implementation; equivalent sensors, microcontrollers, and application interfaces may be substituted.

Federated Learning Workflow

The workflow proceeds iteratively. First, each ESP32 node collects local sensor data and preprocesses them using normalization and feature extraction as appropriate. A TinyML model then performs local inference and incremental training. Instead of transmitting raw sensor measurements, the client sends updated model parameters to the federated server. The server aggregates the received client models using FedAvg and redistributes the updated global model to participating devices. The process repeats across communication rounds until a predefined stopping criterion is reached.

ISSN: 2181-3337 | SCIENTISTS.UZ

Figure 1

Federated TinyML Workflow

Sensor data	Local preprocessing	TinyML training	Model update	FedAvg server	Global model	Redistribution

Note. The figure summarizes the conceptual communication cycle. Raw sensor data remain on the originating client; only model updates are transmitted to the aggregation server.

Table 3

Reference Hardware Specification

Component	Specification
Microcontroller	ESP32-WROOM-32
CPU	Dual-core Xtensa LX6
Clock frequency	240 MHz
SRAM	520 KB
Flash memory	4 MB
Wi-Fi	IEEE 802.11 b/g/n
Bluetooth	BLE 4.2
AI framework	TensorFlow Lite for Microcontrollers

Note. Specifications reproduce the reference configuration described in the source manuscript and should be verified against the exact hardware revision before implementation.

Scientific Contribution

The proposed framework contributes a unified conceptual architecture for combining TinyML and federated learning on resource-constrained ESP32-class devices. Its design goals are (a) to keep raw sensor data local, (b) to reduce communication by exchanging model updates rather than continuous sensor streams, (c) to support energy-aware collaborative learning for long-term IoT deployments, and (d) to provide a scalable structure that can be adapted to heterogeneous sensing applications. These contributions are architectural claims; empirical validation is required before quantitative improvements in energy use, communication cost, accuracy, or scalability can be asserted.

System Model

The Federated TinyML framework consists of a distributed set of K resource-constrained IoT clients connected to a central aggregation server through wireless communication. Each client is equipped with one or more sensors and a microcontroller capable of TinyML inference and local model optimization. Client k owns a local dataset D_k. Because measurements originate from different devices and environments, the local datasets may be non-independent and identically distributed (non-IID).

During communication round t, participating clients receive the current global model, perform local optimization on their private datasets, and return model updates to the server. The server aggregates these updates to obtain the next global model. This design separates raw-data storage from collaborative model optimization: sensor observations remain local, whereas model parameters or parameter updates cross the network.

Table 4

Reference Sensor Configuration

Sensor	Measured parameter	Sampling rate
DHT22	Temperature, humidity	1 Hz
MPU6050	Acceleration, angular velocity	50 Hz
BME280	Pressure, temperature	5 Hz
MQ135	Air quality	1 Hz

Note. Sampling rates are the proposed configuration from the source manuscript and may require adjustment based on application requirements and sensor operating constraints.

ISSN: 2181-3337 | SCIENTISTS.UZ

Limitations and Future Evaluation

The present paper describes a conceptual framework and reference configuration rather than a completed experimental evaluation. Consequently, claims about accuracy, convergence speed, energy savings, communication reduction, and privacy improvement should be tested empirically. Future work should compare centralized training, standalone TinyML, and Federated TinyML under controlled non-IID data partitions; measure bytes transmitted per round and device-level energy consumption; evaluate client dropout and intermittent connectivity; and assess privacy and security risks associated with shared model updates.

Conclusion

Federated TinyML offers a promising architecture for collaborative edge intelligence in distributed IoT systems. By combining on-device TinyML with federated aggregation, the proposed framework keeps raw sensor data on local devices while enabling multiple clients to contribute to a shared model. The four-layer architecture provides a practical structure for sensor acquisition, local inference and training, model aggregation, and application-level decision support. The principal engineering challenge is to achieve useful learning performance within strict memory, computation, communication, and energy constraints. The framework therefore provides a basis for implementation and empirical testing rather than evidence of measured performance. Future experimental work should quantify model accuracy, convergence, communication overhead, energy consumption, robustness, and privacy-related risks across heterogeneous IoT deployments.

REFERENCES

1. Eltazarov, B. T., Rikhsiboyev, R. R., Makhmudov, M. A., Alimova, Z. A., & Kurbonov, S. M. (2024). Remote patient monitoring: DS18B20 sensors in medical systems. SCIENCE Journal, 3(11-1), 123–128.
2. Gaibnazarov, B. B., Makhmudov, M. A., & Tuychiev, F. F. (2024). Smart automatic parking system using NodeMCU microcontroller. Science and Innovation, 3(Special Issue 36), 32–33.
3. Gulyamova, S. T., Tulyaganov, A. A., Abdukarimov, S. N., Mahmudov, M. A., & Ergashev, A. E. (2024). Texnologik jarayonlarni boshqarishda PLC (logik kontrollerlar) ning o‘rni. Science and Innovation, 3(Special Issue 50), 434–436.
4. Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., D’Oliveira, R. G. L., Eichner, H., El Rouayheb, S., Evans, D., Gardner, J., Garrett, Z., Gascón, A., Ghazi, B., Gibbons, P. B., … Zhao, S. (2021). Advances and open problems in federated learning. Foundations and Trends in Machine Learning, 14(1–2), 1–210. https://doi.org/10.1561/2200000083
5. Makhmudov, M. A., Alimova, Z., & Kimizbayeva, A. (2023). Display data in histogram form using the Arduino platform. Universum, 16–19.
6. Makhmudov, M. A., Abdikhakimov, B. I., Ulugberdiyev, A. S., Kimizbayeva, A. E., & Gulyamova, S. T. (2024). Automatic water pump control system using GSM module and Arduino: Automation of water supply system (GSMUZ08). Science and Innovation, 3(A10), 221–226.
7. McMahan, H. B., Moore, E., Ramage, D., Hampson, S., & Agüera y Arcas, B. (2017). Communication-efficient learning of deep networks from decentralized data. In A. Singh & J. Zhu (Eds.), Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (Vol. 54, pp. 1273–1282). PMLR. https://proceedings.mlr.press/v54/mcmahan17a.html
8. Warden, P., & Situnayake, D. (2019). TinyML: Machine learning with TensorFlow Lite on Arduino and ultra-low-power microcontrollers. O’Reilly Media.

ISSN: 2181-3337 | SCIENTISTS.UZ