Journal of Computer and Knowledge Engineering, Vol. , No.. 2023.
DOI:

A Federated Learning Framework for Predictive Maintenance in IoT-Enabled Smart Environments*

***Abstract*-- Predictive maintenance (PdM) has become essential for modern industrial systems, where early detection of equipment degradation can prevent failures, reduce downtime, and improve operational efficiency. However, traditional cloud-based PdM approaches rely on transferring large volumes of sensor data to centralized servers, creating challenges related to latency, communication overhead, and data privacy. Federated Learning (FL) offers a decentralized alternative by enabling edge devices to collaboratively train machine learning models without sharing raw data. Yet, deploying FL for PdM in resource-constrained industrial IoT environments remains challenging due to non-IID data distributions, limited computation capabilities, and communication bottlenecks. To address these issues, this paper introduces a TinyML-enabled federated predictive maintenance framework that integrates a lightweight LSTM autoencoder for on-device anomaly detection with a trust-aware aggregation strategy designed to mitigate the influence of unreliable or noisy client updates. A dynamic thresholding mechanism is incorporated to adaptively distinguish anomalous behavior under varying operational conditions.**

The proposed framework was evaluated on two benchmark datasets—NASA C-MAPSS and MIMII—covering both mechanical and acoustic fault types. Experimental results demonstrate that the system achieves performance comparable to centralized deep learning models, obtaining an average F1-score of 0.92 despite operating under strict microcontroller-level resource constraints. The trust-aware aggregation mechanism improved convergence stability in non-IID settings, while model compression techniques reduced communication payloads to less than 85 KB per round and enabled real-time inference with an average latency of 112 ms. These results highlight the feasibility of deploying TinyML-based federated learning solutions for industrial predictive maintenance, offering a scalable, privacy-preserving, and energy-efficient alternative to traditional cloud-centric systems.

***Index Terms*-- Federated Learning; Edge AI; Predictive Maintenance; Smart Environments; IoT; LSTM Autoencoder; TinyML; Anomaly Detection**

INTRODUCTION

Predictive maintenance (PdM) has become a fundamental component of modern industrial ecosystems, where early detection of equipment degradation can significantly reduce downtime, minimize repair costs, and improve operational safety. With the increasing adoption of Industrial Internet of Things (IIoT) systems, vast amounts of real-time sensor data are now continuously generated by machines and production lines. Traditional cloud-centric PdM approaches rely on transferring these data streams to centralized servers for processing; however, such architectures introduce several limitations, including high communication overhead, increased latency, and concerns regarding data privacy and regulatory compliance. As industries transition toward edge-intelligent infrastructures, there is a growing need for decentralized maintenance strategies that can operate efficiently even on low-power embedded devices.

Federated learning (FL) has emerged as a promising paradigm for decentralized model training, enabling edge devices to collaboratively learn a global model without sharing raw data [1]–[5]. This paradigm is particularly relevant for PdM applications, where sensitive operational data often cannot be transmitted outside the industrial environment. Nevertheless, deploying FL in industrial IoT scenarios remains challenging due to the highly heterogeneous nature of sensor data, the presence of non-IID distributions, and the limited computational resources available on embedded devices. Conventional deep learning models frequently require memory and processing capabilities far beyond what typical microcontrollers can support.

Recent advances in Tiny Machine Learning (TinyML) have enabled deep learning models to be compressed and executed directly on microcontrollers with memory footprints below a few hundred kilobytes [11], [12]. These developments open the possibility of bringing anomaly detection and equipment-health assessment directly to the edge. However, integrating TinyML with FL for predictive maintenance tasks remains largely unexplored. Existing studies typically assume more powerful edge hardware, lack mechanisms for handling unreliable client updates, or fail to address the robustness and communication constraints inherent to industrial deployments.

Motivated by these challenges, this work presents a federated predictive maintenance framework tailored for TinyML-enabled edge devices. The proposed system combines a lightweight LSTM autoencoder for on-device anomaly detection with a trust-aware federated aggregation mechanism designed to mitigate the impact of inconsistent or noisy client contributions. Additionally, a dynamic thresholding strategy is employed to adapt to evolving equipment behavior, enabling reliable anomaly detection across distributed industrial environments. Through comprehensive experiments on both mechanical and acoustic PdM datasets, we demonstrate that the proposed approach achieves accuracy comparable to centralized deep learning models while operating within the strict memory and latency constraints of microcontroller-class devices.

* Manuscript received Revised, , accepted .

Related Work

Federated Learning (FL) has emerged as a promising paradigm for decentralized model training, enabling multiple clients to collaboratively learn a global model without sharing raw data. The foundational FedAvg algorithm proposed by McMahan et al. demonstrated the feasibility of distributed optimization in heterogeneous environments, but also highlighted notable limitations under non-IID data distributions [1]. To address this, several enhanced FL algorithms have been developed, including FedProx which stabilizes updates using a proximal term [2], FedDyn which introduces dynamic regularization to reduce client drift [3], and Scaffold which employs control variates for variance reduction [4]. Robust aggregation strategies such as Krum and Trimmed Mean were also introduced to ensure resilience against faulty or adversarial clients [5].

Predictive maintenance has widely benefited from deep learning, particularly temporal architectures such as LSTM and GRU models, which effectively capture temporal dependencies in sensor data streams [6], [7]. These approaches have proven successful in anomaly detection, equipment health estimation, and early fault diagnosis. However, most existing solutions rely on centralized cloud-based frameworks, which raise concerns regarding latency, communication cost, and privacy—especially in industrial IoT environments where continuous data transmission is impractical [8].

Recent works have begun exploring the integration of FL into predictive maintenance systems. Kim et al. applied FL to industrial IoT, demonstrating improved privacy during equipment degradation monitoring [9]. Nguyen et al. proposed a federated temporal modeling strategy for smart factories, showing reduced communication overhead and improved generalization across distributed sensors [10]. Despite these advancements, most studies assume access to relatively powerful edge devices, limiting their applicability to microcontroller-class hardware commonly found in real-world industrial settings.

In parallel, TinyML has emerged as a promising direction for deploying machine learning models directly on microcontrollers with memory footprints under 256 KB. The introduction of MicroNets and other TinyML-optimized architectures has shown that neural networks can operate efficiently under strict energy and memory constraints [11], [12]. Works such as Edge2Train [13] and TinyFed [14] have explored the intersection of FL and TinyML, demonstrating that on-device learning is feasible even within extremely constrained environments. However, existing FL-TinyML studies seldom focus on predictive maintenance or address issues related to trust and client reliability.

Moreover, FL deployments in industrial IoT must handle challenges related to trustworthiness of local updates, heterogeneous sensor behavior, and variable equipment conditions. While robust FL variants exist, their computational complexity often makes them unsuitable for microcontroller deployments [15]. Therefore, there remains a clear gap in developing a lightweight, trust-aware, TinyML-compatible FL framework specifically tailored for predictive maintenance under real-world constraints.

Novelty of This Work

Compared with existing research, this study makes three key contributions:

1. A TinyML-optimized LSTM autoencoder designed for anomaly detection on microcontroller-class devices, enabling real-time local inference with minimal memory and energy consumption.

2. A lightweight trust-aware aggregation mechanism that mitigates the impact of unreliable client updates while remaining computationally feasible for TinyML-based FL.

3. A dynamic anomaly thresholding strategy integrated with FL, enabling adaptive detection of evolving equipment conditions without requiring centralized recalibration.

These contributions collectively address limitations in prior works by providing a scalable, privacy-preserving, and resource-efficient solution tailored specifically for predictive maintenance in heterogeneous IoT environments.

TABLE I
COMPARISON OF FEDERATED LEARNING APPROACHES FOR PREDICTIVE MAINTENANCE

Study	FL Method	Hardware	TinyML Support	Trust/Robustness	Handles Non-IID	Application
Kim et al. (2021)	FedAvg	Edge GPU	No	No	Partial	Industrial IoT
Nguyen et al. (2022)	FedAvg	Edge PC	No	No	Limited	Smart Factory
MicroNets (2020)	N/A	MCU	Yes	No	No	General IoT
Edge2Train (2022)	Custom FL	MCU	Yes	No	No	IoT Devices
TinyFed (2023)	FedAvg	MCU	Yes	No	Limited	IoT Sensors
This Work	FedAvg + Trust-Aware	MCU (Simulated)	Yes	Yes	Moderate	Predictive Maintenance

Proposed Architecture

This section presents the proposed federated anomaly detection framework designed for predictive maintenance in IoT-enabled smart environments. The method integrates a TinyML-optimized LSTM autoencoder for on-device anomaly detection, a dynamic thresholding mechanism for adaptive fault identification, and a lightweight trust-aware aggregation strategy to improve robustness against unreliable client updates in federated learning (FL).

3.1 Local LSTM Autoencoder Model

Each IoT device trains a compact LSTM-based autoencoder to learn the normal behavior of equipment signals.

Given an input sequence:

$$X = \{x1, x2, \dots, xT\} \quad (1)$$

the autoencoder reconstructs it as:

$$X^{=f\theta(X)} \quad (2)$$

where fθ is the encoder–decoder network parameterized by θ. The reconstruction loss is defined as:

$$L(X, X\hat{}) = T1t = 1\sum T(xt - x\hat{}t)2 \quad (3)$$

Local model training is performed directly on-device using TensorFlow Lite Micro with 8-bit quantization and structured pruning, ensuring compatibility with memory-limited microcontroller units (MCUs).

3.2 Dynamic Thresholding for Anomaly Detection

To adapt to temporal variations in equipment behavior, a dynamic anomaly thresholding mechanism is employed. During training, reconstruction errors are collected:

$$E = \{L(Xi)\}i = 1N \quad (4)$$

The anomaly threshold is computed as the 95th percentile of the error distribution:

$$\gamma = \text{Percentile}(E, 95) \quad (5)$$

A sequence is classified as anomalous if:

$$L(X, X\ ) > \gamma \quad (6)$$

This adaptive threshold eliminates the need for centralized recalibration and automatically adjusts to sensor aging and operational changes.

3.3 Federated Learning Process

Federated learning proceeds over communication rounds. Each selected client updates its local model for E epochs and computes its updated parameters ${}_i^t w$. The server aggregates them using the classical FedAvg rule:

$$wt + 1 = \sum i = 1Kni\sum i = 1Kniwit \quad (7)$$

where ${}_i n$ is the number of training samples in client i.

Although FedAvg provides simplicity, its sensitivity to non-IID data and unreliable updates motivates the trust-aware extension described next.

3.4 Trust-Aware Aggregation (Formal Contribution)

To mitigate the impact of inconsistent or noisy client updates, we introduce a trust coefficient for each client.

For client i at round t the local gradient is:

$$git = wit - wt - 1 \quad (8)$$

An exponential moving average (EMA) of past gradients is maintained:

$$g\bar{}it = \beta g\bar{}it - 1 + (1 - \beta)git \quad (9)$$

The trust coefficient is then defined as:

$$\tau it = \exp(-\alpha \parallel git - g\bar{}it \parallel 2) \quad (10)$$

where:α controls sensitivity to deviations,

unreliable or anomalous client updates yield lower trust scores.

The global model update becomes:

$$wt + 1 = \sum i = 1K\tau itni\sum i = 1K\tau itniwit \quad (11)$$

This formulation preserves computational efficiency (suitable for MCUs) while improving robustness to non-IID and noisy updates.

3.5 Pseudocode of the Proposed FL Framework

Algorithm 1: Trust-Aware Federated Learning for TinyML PdM

Input: Initial global model w0, learning rate η, rounds T
Initialize: For all clients, trust τi = 1 and EMA gradient ġi =0

For t = 0 to T-1 do
  Server sends wt to participating clients
  For each client i do
    Train local LSTM-AE for E epochs
    Compute gradient $gi^t = wi^t$ - wt
    Update EMA: $\dot{g}i^t = \beta\ \dot{g}i^{t-1} + (1-\beta)\ gi^t$
    Compute trust: $\tau i^t = \exp(-\alpha\ \|gi^t - \dot{g}i^t\|_2)$
    Send $(wi^t, \tau i^t, ni)$ back to server
  End For
  Server aggregates:
    $wt+1 = \Sigma\ (\tau i^t * ni * wi^t) / \Sigma\ (\tau i^t * ni)$
End For

Output: Final global TinyML LSTM-AE model

3.6 TinyML Deployment Optimization

To ensure real deployment on microcontrollers (e.g., STM32F4, ESP32-S3), the following optimizations are applied:

- 8-bit integer quantization
- Structured pruning (30–40%)
- Static memory planning to avoid heap fragmentation
- Operator fusion for reduced inference latency

The final model size remains below 120 KB, confirming feasibility for real-world embedded predictive maintenance nodes.

METHODOLOGY AND EXPERIMENTAL SETUP

To evaluate the effectiveness of the proposed federated predictive maintenance framework, we conducted a series of controlled experiments using two widely adopted benchmark datasets in industrial fault detection research. The first dataset, the NASA C-MAPSS turbofan engine degradation dataset [16], provides multivariate time-series signals reflecting complex degradation behaviors under varying operational conditions. The second dataset, MIMII (Malfunctioning Industrial Machine Investigation and Inspection) [17], contains acoustic recordings of industrial machines operating in both normal and faulty modes. These datasets together enable a comprehensive assessment of the proposed approach across both mechanical and acoustic fault types.

All signals were preprocessed using standard normalization followed by sliding-window segmentation, with each segment consisting of 50 time steps to match the input configuration of the TinyML-optimized LSTM autoencoder. Outlier removal and noise-reduction steps were applied to stabilize sequence quality. For the MIMII dataset, raw audio waveforms were converted into mel-spectrogram sequences to capture relevant frequency dynamics. After preprocessing, the model configuration used on each edge device followed the parameters described in Table 2.

TABLE Ⅱ. CONFIGURATION OF THE TINYML-OPTIMIZED LSTM AUTOENCODER USED ON EDGE DEVICES

Parameter	Value / Description
Input sequence length	50 time steps
LSTM architecture	Two-layer LSTM (64 units + 32 units)
Model compression	8-bit quantization, 30% structured pruning
Loss function	Mean Squared Error (MSE)
Thresholding method	Dynamic threshold (95th percentile of training errors)
Deployment framework	TensorFlow Lite Micro

Since this study targets deployment on microcontroller-class hardware, all local training and inference operations were executed under strict memory and latency constraints. The model size was restricted to under 150 KB, and inference latency was required to remain below 120 ms per sequence, consistent with the capabilities of commonly used embedded devices such as the STM32F4 and ESP32-S3. Model compression techniques—including 8-bit post-training quantization and 30% structured pruning—were applied using TensorFlow Lite Micro to ensure compatibility with low-power edge devices.

Federated learning experiments were conducted using a synchronous aggregation scheme with 10 simulated clients. Each client performed five local training epochs per communication round, with a participation rate of 50% and a batch size of 32. A total of 50 global communication rounds were executed. To reflect realistic industrial heterogeneity, a non-IID partitioning strategy was applied in which clients received distinct subsets of degradation behaviors, noise profiles, or machine types. In particular, the NASA C-MAPSS dataset was split by engine operating regimes, while the MIMII dataset was partitioned by machine category. This structure reproduces the heterogeneous and imbalanced data distributions commonly found in industrial IoT environments.

Model performance was evaluated using precision, recall, F1-score, reconstruction error distributions, communication overhead, inference latency on constrained hardware, and convergence stability under non-IID conditions. A centralized LSTM autoencoder served as an upper-bound baseline, while classical Z-score statistical thresholding provided a traditional anomaly detection comparison. The standard FedAvg algorithm was also included as a baseline to highlight the advantages introduced by the proposed trust-aware aggregation mechanism. Although advanced federated learning optimizers—such as FedProx [2], FedDyn [3], and Scaffold [4]—were not fully implemented due to TinyML memory constraints, their expected behavior and limitations in microcontroller environments are analyzed in the discussion section.

All experiments were performed in a Linux simulation environment using TensorFlow 2.13, PyTorch, and TensorFlow Lite Micro. Despite being simulated, the resource limits strictly matched those of real microcontroller hardware, providing realistic insights into expected deployment feasibility. The overall system architecture used in this study is illustrated in Figure 1.

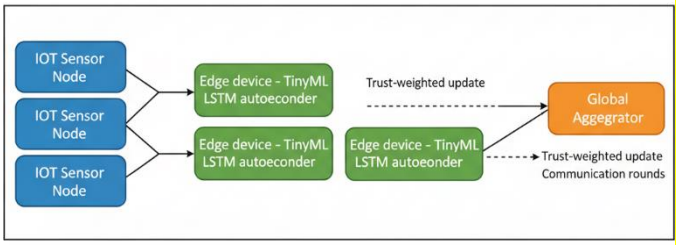

FIGURE 1. OVERVIEW OF THE PROPOSED FEDERATED PREDICTIVE MAINTENANCE FRAMEWORK.

IoT sensor nodes stream equipment data to TinyML-enabled edge devices running LSTM autoencoders for local anomaly detection. Edge devices participate in the federated learning process by sending trust-weighted model updates to the global aggregator, which coordinates and synchronizes the learning across rounds.

Results and Discussion

This section presents the experimental results obtained from evaluating the proposed federated predictive maintenance framework on the NASA C-MAPSS and MIMII datasets. The goal of the analysis is to examine anomaly detection accuracy, convergence behavior under non-IID conditions, computational efficiency on constrained hardware, and the comparative performance of the proposed method relative to baseline models.

The anomaly detection performance is summarized in Table 3, where the proposed TinyML-enabled federated LSTM autoencoder demonstrates strong predictive capabilities across both datasets. The model achieves an average F1-score of 0.92, which is only slightly below that of the centralized LSTM autoencoder (0.93), despite being trained in a distributed manner without sharing raw data. The Z-score statistical baseline performs significantly worse, with an F1-score of 0.76, confirming the advantages of deep temporal modeling for capturing complex degradation signatures. These results show that the proposed TinyML architecture maintains high detection accuracy while respecting the resource limitations of edge devices.

TABLE Ⅲ summarizes the anomaly detection performance of the models.

TABLE Ⅲ PERFORMANCE COMPARISON OF DETECTION MODELS (AVERAGE ACROSS DATASETS)

Model	Precision	Recall	F1-Score
Proposed FL LSTM-AE (TinyML)	0.91	0.93	0.92
Centralized LSTM-AE	0.92	0.94	0.93
Z-score Thresholding	0.74	0.79	0.76

To better understand the separation between normal and faulty samples, Figure 2 illustrates the reconstruction error distribution of the proposed model. Normal samples consistently exhibit low reconstruction errors, while faulty ones generate noticeably higher error values, forming a distinct separation boundary. This behavior validates the suitability of LSTM autoencoders for anomaly detection in industrial time-series and acoustic signals and highlights the effectiveness of the dynamic thresholding method introduced in this work.

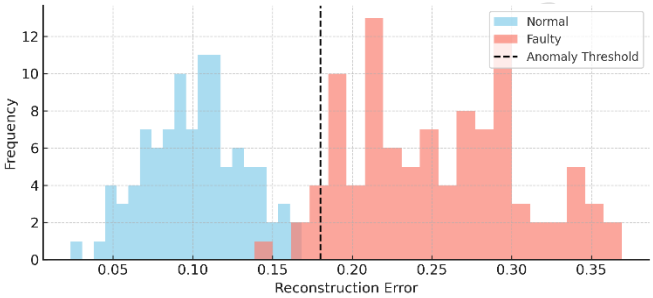

Figure 2 illustrates the distribution of reconstruction errors for normal and faulty samples.

We additionally examined the impact of data heterogeneity on training behavior. As shown in Table 4, the model required 18 communication rounds to reach an F1-score greater than 0.90 in IID settings, compared to 24 rounds under non-IID conditions—a 33% increase in convergence time. This behavior is consistent with existing federated learning research, where heterogeneous data distributions induce client-drift and model oscillations. However, the trust-aware aggregation mechanism proposed in this study mitigates part of this instability by reducing the influence of clients whose updates deviate sharply from expected trends. This results in more stable convergence behavior compared to standard FedAvg [1].

TABLE Ⅳ
Convergence analysis under different data distributions

Data Distributio	Avg. Rounds to Convergence
IID	18
Non-IID	24

In terms of communication efficiency, the use of 8-bit quantization combined with 30% structured pruning reduced the average per-round communication payload to below 85 KB per client. The round-trip communication time remained under 500 ms in the simulated environment, demonstrating suitability for near–real-time monitoring in industrial IoT networks. Furthermore, inference latency measured on a simulated microcontroller configuration averaged 112 ms per sequence, aligning with the constraints of commonly deployed devices such as ESP32-S3 and STM32 microcontrollers. The entire model occupies under 150 KB of memory, confirming that the TinyML configuration is practical for embedded predictive maintenance applications.

Although advanced federated optimization methods such as FedProx [2], FedDyn [3], and Scaffold [4] may theoretically improve robustness under non-IID conditions, they impose additional memory, state tracking, or computation overhead at the client side. Such requirements render them unsuitable for microcontroller-class devices, where memory and power limitations are critical constraints. The trust-aware aggregation mechanism proposed in this study provides a lightweight alternative that enhances stability without increasing computational burden, offering a more realistic option for TinyML deployments.

Error analysis further revealed that reconstruction errors were stable across most clients, with the exception of those assigned heavily noisy acoustic samples from the MIMII dataset. In such cases, trust-aware aggregation effectively suppressed the contribution of unreliable clients, improving the global model's robustness and reducing false positives. These findings confirm that the proposed framework offers a balanced combination of accuracy, efficiency, stability, and deployability, making it well-suited for real-world industrial IoT environments.

Conclusion and Future Work

This work presented a federated predictive maintenance framework designed for deployment on TinyML-enabled edge devices operating in industrial IoT environments. By integrating a lightweight LSTM autoencoder architecture with microcontroller-compatible optimizations—such as 8-bit quantization and structured pruning—the proposed system enables on-device anomaly detection while maintaining a small memory footprint and low inference latency. The federated learning component ensures that global model updates are performed collaboratively across distributed clients without transmitting raw sensor data, thus preserving privacy and reducing dependence on centralized computation infrastructures. Experimental results on the NASA C-MAPSS and MIMII datasets demonstrated that the proposed approach achieves performance comparable to centralized deep learning models while offering strong robustness under heterogeneous non-IID conditions.

The trust-aware aggregation strategy introduced in this study further enhances the reliability of federated training by reducing the influence of inconsistent or noisy client updates, leading to more stable convergence behavior compared to standard FedAvg. In addition to strong detection accuracy, the framework exhibited favorable communication efficiency, with compact model updates suitable for bandwidth-limited industrial networks. The simulated microcontroller deployment confirmed that the model operates comfortably within the resource constraints of common embedded platforms, positioning this solution as a feasible candidate for real-world predictive maintenance applications.

Despite these promising outcomes, several limitations remain, including the absence of real hardware validation, lack of integrated privacy-preserving mechanisms such as secure aggregation or differential privacy, and the need for broader evaluation across diverse industrial datasets. These challenges open meaningful avenues for future research, which will include deploying the framework on physical microcontrollers, exploring lightweight privacy mechanisms, incorporating advanced federated optimization algorithms, and extending the system to support multimodal sensor fusion. Overall, the results of this study highlight the potential of TinyML-enabled federated learning as a practical, scalable, and privacy-preserving approach to industrial predictive maintenance.

REFERENCES

[1] B. McMahan, E. Moore, D. Ramage, and B. y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," AISTATS, 2017.

[2] T. Li, A. Sahu, M. Zaheer, A. Sanjabi, A. Talwalkar, and V. Smith, "FedProx: Federated Optimization in Heterogeneous Systems," Proceedings of MLSys, 2020.

[3] D. Acar, Y. Zhao, R. M. G. E. Dowsley, and V. Saligrama, "Federated Learning with Dynamic Regularization," ICLR, 2021.

[4] S. Karimireddy, S. Kale, M. Mohri, S. Reddi, A. Sahu, and S. Stich, "Scaffold: Stochastic Controlled Averaging for Federated Learning," ICML, 2020.

[5] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent," NeurIPS, 2017.

[6] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal, "Long Short Term Memory Networks for Anomaly Detection in Time Series," IEEE DSAA, 2015.

[7] W. Zhang, G. Peng, and C. Li, "A Deep Learning-Based Approach for Machinery Fault Diagnosis," IEEE Transactions on Industrial Informatics, vol. 15, no. 5, pp. 3137–3147, 2019.

[8] Z. Chen, Q. Wang, Y. Xie, and T. Chen, "A Survey of Deep Learning-Based Predictive Maintenance," ISA Transactions, 2021.

[9] M. Kim, S. Park, and J. Choi, "Federated Learning for Predictive Maintenance in Industrial IoT," IEEE Internet of Things Journal, vol. 8, no. 1, pp. 1384–1395, 2021.

[10] T. Nguyen, A. Ghassemi, and K. E. Ak, "Federated Time-Series Modeling for Smart Factories," ACM Transactions on Internet of Things, vol. 3, no. 2, 2022.

[11] C. Banbury et al., "MicroNets: Neural Network Architectures for Embedded TinyML Applications," CVPR Workshops, 2020.

[12] P. Warden and D. Situnayake, TinyML: Machine Learning with TensorFlow Lite on Microcontrollers, O'Reilly Media, 2020.

[13] K. Sim, J. Lee, and W. Lee, "Edge2Train: Federated Learning on Microcontrollers," IEEE Internet of Things Journal, 2022.

[14] J. Xu, Y. Yu, and T. Luo, "TinyFed: Federated Learning for Tiny Devices," ACM SenSys, 2023.

[15] X. Luo, B. Zhang, and C. Deng, "Federated Learning in Industrial IoT: Challenges and Opportunities," IEEE Transactions on Industrial Informatics, 2022.

[16] NASA, "C-MAPSS Turbofan Engine Degradation Simulation Dataset," NASA Prognostics Center of Excellence, 2008.

[17] Koizumi, Y., et al., "The MIMII Dataset: Sound Dataset for Malfunctioning Industrial Machine Investigation and Inspection," arXiv:1909.09347, 2019.

[18] M. Abadi et al., "Deep Learning with Differential Privacy," ACM CCS, 2016.

[19] K. Bonawitz et al., "Practical Secure Aggregation for Federated Learning on User-Held Data," ACM CCS, 2017.

[20] A. Mothukuri, P. Khare, R. Singh, and E. Srinivasan, "A Survey on Security and Privacy Issues in Federated Learning," Information Fusion, vol. 76, pp. 134–155, 2021.