# Demo: A Practical Testbed for Decentralized Federated Learning on Physical Edge Devices

Chao Feng*, Nicolas Huber*, Alberto Huertas Celdrán*, Gérôme Bovet†, Burkhard Stiller*
*Communication Systems Group, Department of Informatics, University of Zurich UZH, CH–8050 Zürich, Switzerland
[cfeng, huertas, stiller]@ifi.uzh.ch, nicolas.huber3@uzh.ch
†Cyber-Defence Campus, armasuisse Science & Technology, CH–3602 Thun, Switzerland gerome.bovet@armasuisse.ch

***Abstract*—Federated Learning (FL) enables collaborative model training without sharing raw data, preserving participant privacy. Decentralized FL (DFL) eliminates reliance on a central server, mitigating the single point of failure inherent in the traditional FL paradigm, while introducing deployment challenges on resource-constrained devices. To evaluate real-world applicability, this work designs and deploys a physical testbed using edge devices such as Raspberry Pi and Jetson Nano. The testbed is built upon a DFL training platform, NEBULA, and extends it with a power monitoring module to measure energy consumption during training. Experiments across multiple datasets show that model performance is influenced by the communication topology, with denser topologies leading to better outcomes in DFL settings.**

***Index Terms*—Testbed, Federated Learning, Resource Consumption**

## I. Introduction

With the proliferation of advanced sensors, enhanced processing capabilities, and widespread connectivity, edge devices now generate vast amounts of data, offering significant opportunities for Machine Learning (ML) [1]. However, the distributed nature of data from IoT devices makes traditional centralized ML approaches impractical due to privacy concerns, communication overhead, and regulatory constraints [2]. Federated Learning (FL), first introduced by [3], addresses these issues by enabling devices to collaboratively train a shared model without sharing raw data, exchanging only model updates instead. FL can be categorized into Centralized Federated Learning (CFL), where a central server coordinates training and aggregation, and Decentralized Federated Learning (DFL), where nodes interact directly without a central authority. While CFL is widely used, DFL offers better scalability, fault tolerance, and eliminates single points of failure and trust bottlenecks [2].

Training FL models on resource-constrained devices, such as smartphones, IoT devices, and edge computing nodes, introduces significant challenges due to their limited computational power, memory, and energy capacities, complicating the implementation of privacy-preserving methods and efficient model training [4]. Current DFL platforms, such as NEBULA [5], primarily rely on simulation-based or virtualization techniques, like Docker containers, for development and testing. However, these approaches fail to adequately reflect the constraints and complexities encountered when deploying FL on heterogeneous, resource-constrained hardware, using diverse datasets and varying overlay networks.

This work addresses this research gap by designing and implementing a realistic, physical DFL testbed utilizing heterogeneous resource-constrained devices, including multiple models of Raspberry Pi and NVIDIA developer kits such as the Jetson series, to replicate real-world deployment scenarios closely. The testbed supports experimentation across multiple topological configurations, enabling comprehensive evaluations with diverse datasets. Compared with simulation-based or virtualization-based platforms, the proposed physical testbed achieves comparable model training performance while accurately reflecting device heterogeneity, hardware constraints, and network topologies, thus providing a robust and realistic environment for evaluating decentralized federated learning frameworks. Additionally, the designed testbed incorporates mechanisms to record energy consumption throughout the model training process, laying a foundation for subsequent research on DFL systems' sustainability and energy efficiency.

## II. Testbed Design

This section details the construction of a physical testbed designed to evaluate DFL under realistic hardware and network conditions. Rather than relying on simulated or virtualized setups, the testbed is built upon actual, heterogeneous edge devices with constrained resources. It incorporates Raspberry Pi 4 Model B units (two with 4GB RAM and one with 2GB RAM, running Debian GNU/Linux 12) and an NVIDIA Jetson Nano Developer Kit (running Ubuntu 18.04.6 LTS), interconnected via a local Ethernet network.

A DFL platform, *i.e.*, NEBULA, serves as the underlying infrastructure for model training, aggregation, and inter-device communication within the testbed. Originally, NEBULA employs Docker containers to emulate DFL nodes, necessitating several modifications for physical hardware deployment:

- Deploying lightweight HTTP servers on each device to securely receive FL configurations from a controller.
- Implementing periodic HTTP-based transmission of real-time metrics (e.g., CPU, memory usage) from nodes to the controller, facilitating continuous frontend visualization.
- Extending monitoring capabilities to track energy consumption using JT-TC66C USB multimeters.

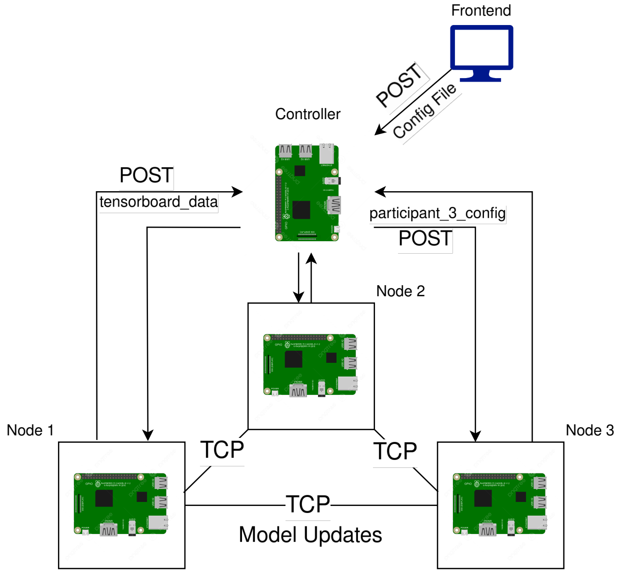

Fig. 1. System architecture showing FL configuration and metric transmission.

Figure 1 depicts the testbed architecture. The controller receives configuration files from the frontend and distributes them to participant nodes. Each node hosts a local HTTP server to accept the configuration, followed by the server's termination to release the port. Nodes then establish TCP connections with their neighbors and begin decentralized training.

Figure 2 presents the scenario execution sequence: the frontend initiates the experiment by sending configuration parameters to the controller, which bootstraps node instances and distributes configuration files. During the training process, nodes periodically transmit training metrics to the controller, which are then visualized on the frontend. Additionally, if energy monitoring is enabled, nodes log power consumption throughout the training cycle and transmit a summary at the end of the scenario. The JT-TC66C multimeters measure power consumption in real time and provide both periodic metric updates and post-training batch reports. This dual-reporting mechanism enables accurate sustainability assessments without interrupting model training.

## III. Setup and Demonstration

To evaluate the performance and feasibility of the proposed DFL testbed, a series of experiments were conducted under two deployment scenarios: Physical Scenario and Virtualized Scenario.

TABLE I
Comparison of Validation Scenarios.

| Characteristic | Physical Scenario | Virtualized Scenario |
|---|---|---|
| Participant | 3 Raspberry Pi 4,<br>1 NVIDIA Jetson Nano | 4 Docker containers |
| Dataset | MNIST<br>FashionMNIST | MNIST<br>FashionMNIST |
| Topology | Fully connected,<br>Star, Ring, Random | Fully Connected |

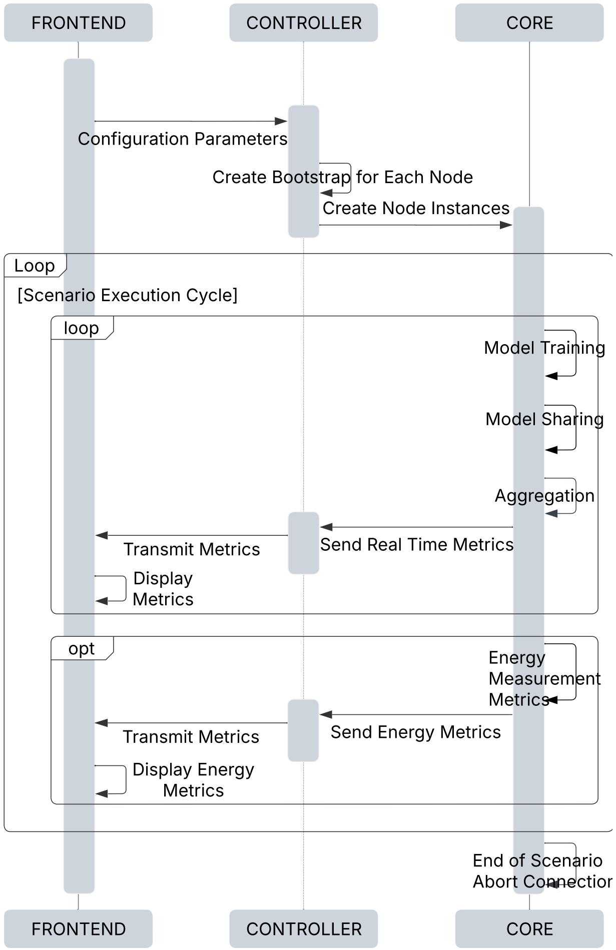

Fig. 2. Sequence diagram of configuration distribution, model training, and metric reporting.

### A. Experiments Setup

Table I provides an overview of the experimental configuration in both settings. The physical testbed includes four edge devices: three Raspberry Pi 4 Model B boards and one NVIDIA Jetson Nano. This configuration introduces heterogeneity in terms of CPU and GPU capabilities. The Jetson Nano features a 128-core Maxwell GPU and a quad-core Cortex-A57 CPU (1.43GHz), whereas the Raspberry Pi 4 uses a Cortex-A72 CPU (1.5GHz). In contrast, the virtualized setup uses Docker containers deployed on a machine with an AMD Ryzen 7 5825U processor, 16GB RAM, and 512GB SSD, ensuring consistent and stable performance.

Both scenarios use MNIST [6] and FashionMNIST [7], two widely recognized benchmarks for image classification. A simple multi-layer perceptron (MLP) model was used across all experiments, and the FedAvg [3] algorithm was trained over 10 federation rounds with one local epoch per round. To simulate realistic data heterogeneity, all datasets were partitioned independently and identically distributed (IID).

TABLE II
SUMMARY OF EXPERIMENTAL RESULTS FOR PHYSICAL AND VIRTUALIZED SCENARIOS

| Scenario | Avg. $F_1$-Score | CPU Usage (%) | RAM Usage (%) | Net Traffic (MB) | Power (W) | Energy (J) |
|---|---|---|---|---|---|---|
| Fully Conn. MNIST (Phys.) | 82.0 | 25.7 | 33.3 | 30 | 3.3 | 1404 |
| Fully Conn. Fashion (Phys.) | 81.0 | 28.7 | 33.3 | 30 | 3.5 | 1350 |
| Star MNIST (Phys.) | 76.0 | 25.8 | 32.8 | P0: 30, Oth: 10 | 3.0 | 1251 |
| Star Fashion (Phys.) | 75.0 | 29.1 | 33.9 | P0: 30, Oth: 10 | 3.5 | 1324 |
| Ring MNIST (Phys.) | 78.5 | 26.0 | 32.9 | 20 | 3.05 | 1249 |
| Ring Fashion (Phys.) | 68.8 | 28.3 | 33.1 | 20 | 3.45 | 1404 |
| Random MNIST (Phys.) | 74.7 | 26.1 | 33.4 | Varied | 3.1 | 1305 |
| Random Fashion (Phys.) | 64.9 | 28.7 | 33.0 | Varied | 3.5 | 1368 |
| Fully Conn. MNIST (Virt.) | 82.0 | 26.8 | 7.4 | 30 | N/A | N/A |
| Fully Conn. Fashion (Virt.) | 81.0 | 34.4 | 7.4 | 30 | N/A | N/A |

To assess the testbed's flexibility and DFL's sensitivity to communication structures, the physical setup tested four topologies: fully connected, star, ring, and random. Each topology was evaluated using both datasets. The virtualized baseline used only the fully connected topology due to simplicity and consistency constraints.

### B. Experimental Results

The evaluation metrics include model $F_1$-score, CPU and RAM usage, network traffic, power draw, and energy consumption. Results from all physical experiments were compared against a virtualized baseline to assess both performance and resource efficiency.

As shown in Table II, the fully connected topology consistently achieved the highest average $F_1$-scores (81–82%) on both MNIST and Fashion-MNIST datasets. In contrast, sparser topologies like ring and random exhibited more variability and generally lower average performance. The star topology showed that central nodes benefited from higher communication loads, often yielding higher accuracy than edge nodes.

Resource consumption reflected the underlying training cycles. Average CPU usage ranged from 25% to 29%, with peak usage exceeding 40%. Power consumption followed similar patterns, ranging from 3 to 3.5 watts, and total energy use varied between 1251 and 1404 joules. RAM utilization remained consistent across experiments, with one Raspberry Pi (2 GB) showing higher usage due to memory constraints. Bandwidth usage aligned with topological structure: fully connected and star topologies generated higher traffic per node, while ring and random topologies maintained lower and more localized communication overhead.

The virtualized scenario matched physical $F_1$-scores but completed training significantly faster, highlighting the computational limits of edge devices. Despite lower training speeds, the physical testbed effectively replicated learning behavior, validating its feasibility for DFL research under realistic conditions.

## IV. CONCLUSION

This work presents a realistic, energy-aware testbed for evaluating DFL on heterogeneous, resource-constrained edge devices. Built on top of the NEBULA platform, the testbed integrates REST-based communication and real-time monitoring, including power and energy tracking using JT-TC66C multimeters. Experiments conducted across multiple topologies and two benchmark datasets (MNIST and Fashion-MNIST) demonstrate that accurate models can be trained on physical devices with limited computational resources, achieving performance comparable to virtualized environments. Resource usage, CPU, memory, bandwidth, and energy, remained within practical limits, confirming the testbed's suitability for realistic DFL research. Future extensions include scaling to larger federations, exploring more advanced topologies, optimizing training efficiency, and testing deployment in real-world IoT and mobile scenarios.

## ACKNOWLEDGMENT

This work has been partially supported by *(a)* the Swiss Federal Office for Defense Procurement (armasuisse) with the CyberDFL project (CYD-C-2020003) and *(b)* the University of Zürich, UZH.

## REFERENCES

[1] A. Imteaj, U. Thakker, S. Wang, J. Li, and M. H. Amini, "Federated learning for resource-constrained iot devices: Panoramas and state-of-the-art," *CoRR*, vol. abs/2002.10610, 2020.

[2] E. T. Martínez Beltrán, M. Q. Pérez, P. M. S. Sánchez, S. L. Bernal, G. Bovet, M. G. Pérez, G. M. Pérez, and A. H. Celdrán, "Decentralized federated learning: Fundamentals, state of the art, frameworks, trends, and challenges," *IEEE Communications Surveys &amp; Tutorials*, vol. 25, no. 4, p. 2983–3013, 2023.

[3] J. Konečnỳ, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," *arXiv preprint arXiv:1610.02527*, 2016.

[4] A. Huertas Celdrán, P. M. Sánchez Sánchez, C. Feng, G. Bovet, G. M. Pérez, and B. Stiller, "Privacy-preserving and syscall-based intrusion detection system for iot spectrum sensors affected by data falsification attacks," *IEEE Internet of Things Journal*, vol. 10, no. 10, pp. 8408–8415, 2023.

[5] E. T. M. Beltrán, Á. L. P. Gómez, C. Feng, P. M. S. Sánchez, S. L. Bernal, G. Bovet, M. G. Pérez, G. M. Pérez, and A. H. Celdrán, "Fedstellar: A platform for decentralized federated learning," *Expert Systems with Applications*, vol. 242, p. 122861, 2024.

[6] L. Deng, "The mnist database of handwritten digit images for machine learning research [best of the web]," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[7] H. Xiao, K. Rasul, and R. Vollgraf. (2017) Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms.