PhD school: Network-Enhanced On-Device AI: a Recipe for Interoperable and Cooperative AI Applications

Paulius Daubaris
University of Helsinki
Helsinki, Finland
paulius.daubaris@helsinki.fi

Sasu Tarkoma
University of Helsinki
Helsinki, Finland
sasu.tarkoma@helsinki.fi

Roberto Morabito
EURECOM
Biot, France
roberto.morabito@eurecom.fr

Abstract

TinyML enables the execution of machine learning models on resource-constrained devices, offering benefits in privacy and energy consumption. However, current research often limits TinyML to single-device, single-task scenarios, hindering the potential for collaborative edge computing. In this respect, we believe that systems where TinyML-enabled devices collaborate directly with one another remain largely unexplored, presenting significant opportunities for innovation in cooperative edge computing. This doctoral research investigates collaborative TinyML systems comprised of highly constrained devices (e.g., < 1000 KB RAM, < 2000 KB flash storage), with objectives such as: developing methods for intelligent computation offloading, creating a lightweight networking system for node communication, and investigating protocols for enhanced interoperability.

CCS Concepts

• **Networks** → **Network protocols**; • **Computer systems organization** → Embedded hardware; *Embedded software*; • **Computing methodologies** → *Neural networks.*

Keywords

TinyML, edge computing, constrained devices, networking

1 Research Overview

Machine Learning (ML) at the edge has progressed to the point where it is now possible to shrink ML models and execute them on resource-constrained devices, invoking a paradigm known as TinyML [6, 9]. This advancement has opened up new possibilities for data processing and sensing at the edge. For example, in terms of privacy and energy consumption, TinyML is favourable, as data processing is performed on-device and, generally, the algorithms executed on such devices are less resource-intensive [9]. Nonetheless, TinyML is usually perceived as a single-device, single task paradigm [13]. Such a setup prevents TinyML-enabled devices from collaborating with heterogeneous edge nodes to tackle complex tasks, such as optimizing system performance, enhancing data accuracy, or improving energy efficiency. Lack of collaboration restricts the potential of edge computing, as the isolated operation of individual nodes fails to exploit the synergies that could be achieved through coordinated efforts (e.g., personalization [7]).

Related work typically follows three main patterns: 1) edge devices used for this type of research often fall under general purpose device group (J) depicted in Table 1, which are powerful enough to execute ML models running with the support of software unavailable to more constrained devices [8]; 2) considered edge devices fall under the microcontroller device group (M), performing inference solely on-device, without considering the benefits of leveraging the knowledge or capabilities of proximate edge nodes [9]; 3) different device group nodes are used for the task of federated learning (see, for example [14]). These common approaches do not bridge the gap between two device groups. The existing gap restricts efficient utilization of the edge environment especially for use cases where having contextual information matters.

Group	Name	Data Size	Code Size
M	C0	« 10 KB	« 100 KB
M	C1	10 KB	100 KB
M	C2	50 KiB	250 KB
M	C3	100 KB	500..1000 KB
M	C4	300..1000 KB	1000..2000 KB
J	C10	(16..)32..64..128 MB	4..8..16 MB
J	C15	0.5..1 GB	(substantial)
J	C16	1..4 GB	(substantial)
J	C17	4..32 GB	(substantial)
J	C19	(substantial)	(substantial)

Table 1: Classes of constrained devices (KB = 1024 bytes). Adapted from [3]

Considering the limitations of these devices (e.g., lacking capacity to run sophisticated software and network stacks, limited resources), we see the need to investigate edge systems comprised of actually constrained nodes (i.e., (M) group devices) that interoperate, share local knowledge, and leverage proximate nodes within the bounds of the edge environment. Such approach could invoke a new generation of edge use cases with additional opportunities to optimize different aspects of such systems (e.g., intelligent warehouse environment management, autonomous vehicles). Nonetheless, to complete the puzzle, we identify pieces related to software and networking missing. Ultimately, we seek to answer the following research questions:

- **RQ1:** *How can resource-constrained edge systems perform inference in a more sustainable manner without performance degradation from the application point of view?*
  We seek to understand how to minimize the energy footprint of resource-constrained devices as inference remains a resource-intensive task.
- **RQ2:** *What are the networking protocol requirements for collaborative, resource-constrained edge systems?*
  We will analyze state-of-the-art application layer protocols, identify gaps, and extend them in order to facilitate practical communication between nodes, enabling them to discover each other and integrate seamlessly.
- **RQ3:** *How can device capabilities be represented among heterogeneous resource-constrained devices to maintain interoperability?*

We will explore the potential for interoperable data representation solutions to represent both device and application capabilities. This aims to ensure uniform knowledge among heterogeneous devices, requiring minimal adaptation and maintenance.

2 Research Approach and Methodology

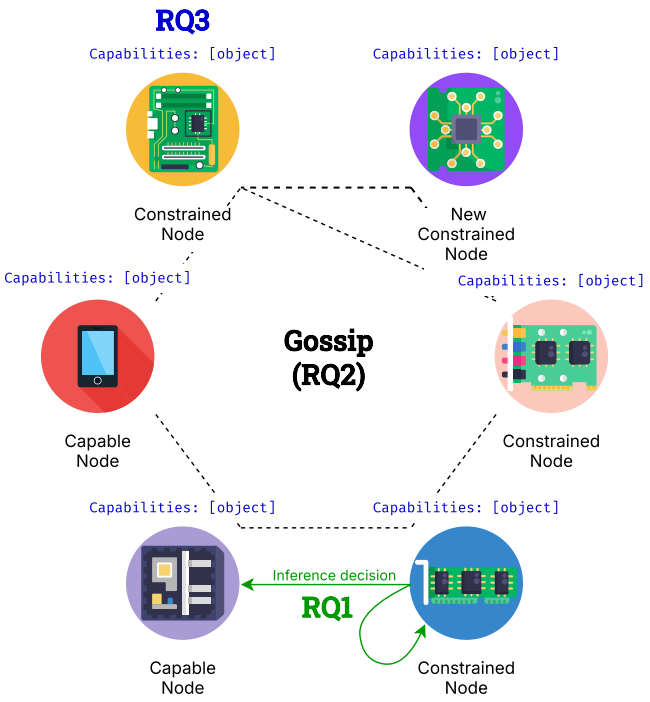

Figure 1: A high-level representation of the envisioned system encompassing the research questions

Figure 1 situates the research questions within the context of an edge environment that includes both constrained (M) and general-purpose (J) devices. In the remainder of this section, we elaborate on our research approach and methodology for addressing each of these questions.

During the initial stage of the project, we investigate the potential for constrained devices to intelligently offload their computations using a combination of Hierarchical Inference (HI) [1] and Early-Exit (EE) [11] models. This approach, illustrated in Figure 2, enables TinyML-enabled devices to transfer tasks to more capable nodes when Quality of Service (QoS) metrics are not met by on-device inference, thus aiming to improve both the accuracy, efficiency and sustainability of the system. To gain insight into how well the approach works, we benchmark and compare on-device, HI, EE, EE-HI models in terms of accuracy, latency, and energy consumption, thus addressing **RQ1**.

Nonetheless, the initial work is limited due to the fact that it considers cloud as the offloading target without being aware of other proximate devices that could be a more sustainable choice for offloading computations. As a result, we explore the possibilities of

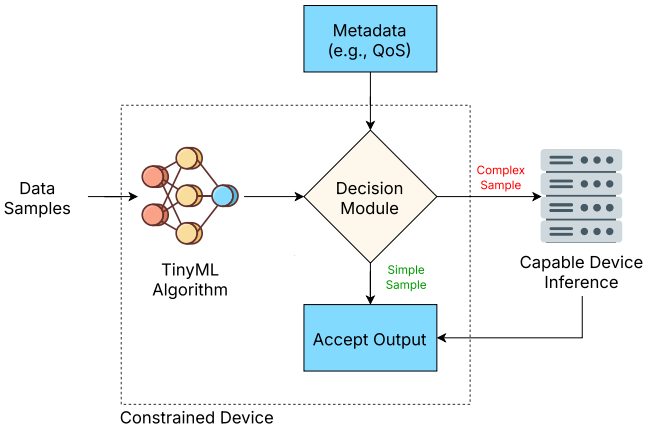

Figure 2: HI appraoch for inference at the network edge. Adapted from [1]

building a network backbone – a lightweight, gossip-based networking system. This system, illustrated in Figure 3, will enable nodes to communicate, spread their local knowledge, thus discovering peers, verify membership, and transfer computations using well-known and established algorithms [4, 5]. Moreover, with such software solution we seek to provide the nodes the capability to seamlessly integrate into the environment. This task involves investigating networking solutions for TinyML-enabled resource-constrained devices and determining the most appropriate protocols, data types and formats, and implementation aspects, thereby addressing **RQ2**. We will conduct extensive benchmarks and compare our solution to the state-of-the-art.

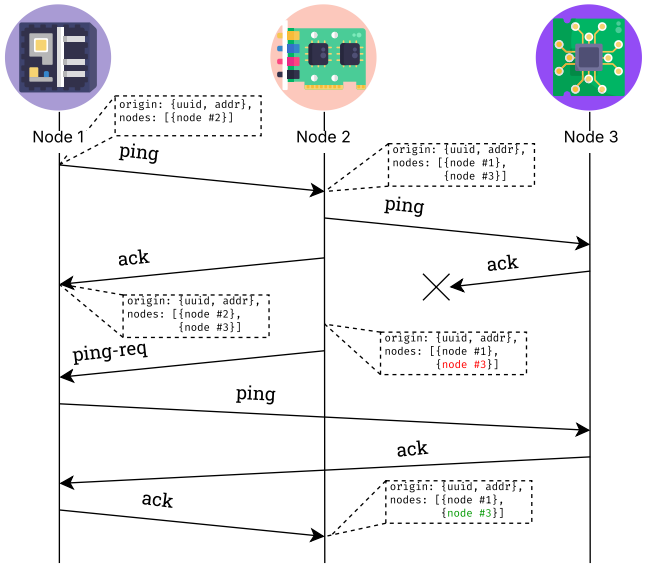

Figure 3: The workflow of a lightweight, gossip-based networking system. This enables peer edge nodes to be discovered, etc.

To tackle **RQ3**, we will build on top of the solution of RQ2 and investigate lightweight application layer protocols and means to

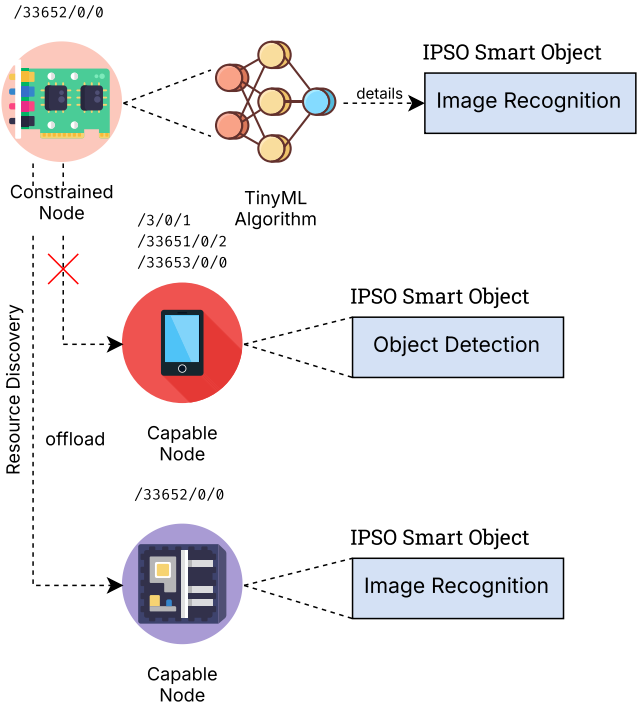

Figure 4: Contextualization of IPSO objects within the collaborative edge environment. Device resources and AI capabilities can be represented in standardized way within the cluster of devices

extend them to represent device capabilities in a standardized manner for enhanced interoperability. One example of such an enabler is IPSO Smart Objects that leverage standardized objects among various devices [12]. Such objects can be created to encompass information about device ML capabilities as discussed in [10]. Leveraging such an approach can provide the means to represent constrained TinyML-enabled device inherent capabilities that can be further used to either offload computations to more capable devices or collaborate with other devices on joint tasks as shown in Figure 4.

3 Preliminary Results

Preliminary results for the initial phase during which we set out to build a system employing HI with the addition of EE models showing that exiting the neural network early and offloading computations is, in certain cases, advantageous [2]. The strategy allows for energy savings and reduced latency compared to solely performing inference on the device or offloading the data to a more capable device for remote inference. We conducted experiments to analyze the latency, energy and accuracy metrics with 5 different constrained devices for image classification tasks leveraging models for two datasets: CIFAR-10 and ImageNet. The experiment results demonstrated that the EE-HI approach can achieve even better performance than HI, with the reduction of latency and energy consumption to up to 60%. Although this work used image classification as the main task, HI and EE-HI are can be applied to other use cases, such as audio classification, object detection.

4 Challenges and Feedback Needs

Based on the current experiences in carrying out research at a Ph.D. faculty level, some technical and non-technical challenges desired to be discussed are:

- Implementation work can be quite challenging and time-consuming. It is certainly difficult to do it alone and also to find peers to collaborate with, especially considering that the topic of interest is quite different than of other students in the faculty. It would be useful to discuss of how to approach and find potential collaboration opportunities within such an environment.
- Moreover, the stream of new ideas can be quite distracting, especially when reading new research. It begs a question asking how not to fall into the trap of the "new trend" and instead focus on the main line of work or balance between the two.
- Beyond the aspects above, it would be useful to discuss time management and how to approach tasks that are similar in priority and also, how more experienced members of the faculty approach this issue.

References

[1] Ghina Al-Atat, Andrea Fresa, Adarsh Prasad Behera, Vishnu Narayanan Moothedath, James Gross, and Jaya Prakash Champati. 2023. The Case for Hierarchical Deep Learning Inference at the Network Edge *(NetAISys '23)*. Association for Computing Machinery, New York, NY, USA, Article 3, 6 pages.

[2] Adarsh Prasad Behera, Paulius Daubaris, Iñaki Bravo, José Gallego, Roberto Morabito, Joerg Widmer, and Jaya Prakash Varma Champati. 2024. Exploring the Boundaries of On-Device Inference: When Tiny Falls Short, Go Hierarchical. arXiv:2407.11061

[3] C. Bormann, M. Ersue, A. Keranen, and C. Gomez. 2022. *Terminology for Constrained-Node Networks.* Internet-Draft. Internet Engineering Task Force. https://www.ietf.org/archive/id/draft-bormann-lwig-7228bis-08.html Work in Progress.

[4] Armon Dadgar, James Phillips, and Jon Currey. 2018. Lifeguard: Local Health Awareness for More Accurate Failure Detection. In *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*. 22–25.

[5] A. Das, I. Gupta, and A. Motivala. 2002. SWIM: scalable weakly-consistent infection-style process group membership protocol. In *Proceedings International Conference on Dependable Systems and Networks*. 303–312.

[6] Dr. Lachit Dutta and Swapna Bharali. 2021. TinyML Meets IoT: A Comprehensive Survey. *Internet of Things* 16 (2021), 100461.

[7] Stefanos Laskaridis, Stylianos I. Venieris, Alexandros Kouris, Rui Li, and Nicholas D. Lane. 2024. The Future of Consumer Edge-AI Computing. *IEEE Pervasive Computing* (2024), 1–10.

[8] M. G. Sarwar Murshed, Christopher Murphy, Daqing Hou, Nazar Khan, Ganesh Ananthanarayanan, and Faraz Hussain. 2021. Machine Learning at the Network Edge: A Survey. *ACM Comput. Surv.* 54, 8, Article 170 (Oct. 2021), 37 pages.

[9] Ramon Sanchez-Iborra and Antonio F. Skarmeta. 2020. TinyML-Enabled Frugal Smart Objects: Challenges and Opportunities. *IEEE Circuits and Systems Magazine* 20, 3 (2020), 4–18.

[10] Tomasz Szydlo and Marcin Nagy. 2023. Device management and network connectivity as missing elements in TinyML landscape. arXiv:2304.11669 [cs.NI]

[11] Surat Teerapittayanon, Bradley McDanel, and H.T. Kung. 2016. BranchyNet: Fast inference via early exiting from deep neural networks. In *2016 23rd International Conference on Pattern Recognition (ICPR)*. 2464–2469.

[12] David Tracey and Cormac Sreenan. 2017. OMA LWM2M in a holistic architecture for the Internet of Things. In *2017 IEEE 14th International Conference on Networking, Sensing and Control (ICNSC)*. 198–203.

[13] Vasileios Tsoukas, Anargyros Gkogkidis, Eleni Boumpa, and Athanasios Kakarountas. 2024. A Review on the emerging technology of TinyML. *ACM Comput. Surv.* 56, 10, Article 259 (June 2024), 37 pages.

[14] Yunfan Ye, Shen Li, Fang Liu, Yonghao Tang, and Wanting Hu. 2020. EdgeFed: Optimized Federated Learning Based on Edge Computing. *IEEE Access* 8 (2020), 209191–209198.