PhD School: A Privacy-Preserving and Resilient Framework for Distributed Modular Neural Networks on the Tiny Edge

Gregory De Ruyter
Hans Hallez*
gregory.deruyter@kuleuven.be
hans.hallez@kuleuven.be
M-Group, DistriNet
KU Leuven
Leuven, Belgium

Mathias Verbeke
mathias.verbeke@kuleuven.be
M-Group, DTAI, STADIUS
KU Leuven
Leuven, Belgium

Bart Vanrumste
bart.vanrumste@kuleuven.be
e-Media Research Lab, STADIUS
KU Leuven
Leuven, Belgium

Abstract

Due to the large amount and diversity of data generated by Internet of Things (IoT) devices, Artificial Intelligence (AI) has seen a breakthrough in IoT applications in recent years. Traditionally this implies that data must be sent to the cloud, which increases communication costs, causes delays in system response and makes data vulnerable to privacy breaches. A solution is to migrate AI models to constrained devices. However, these devices often have limited computational resources and have a limited ability to implement AI models. Besides that, the system also needs to be flexible when devices fail or reconnect. Therefore, this research project aims to develop a novel framework for distributing neural networks and enhancing their resilience on constrained edge devices. During the realization of the framework, research will be done on how a model can be efficiently distributed over a network of edge devices, considering energy consumption, memory footprint and bandwidth usage. Secondly, the project will explore ways to make neural networks adaptable to various combinations of input devices, making the inference more resilient in case of failing devices. Lastly, the integration of distributed learning will be investigated to enhance model performance and personalize predictions without sacrificing privacy. To demonstrate the valorization potential of this research, all results will be implemented and validated on two real-world use cases, from which a software framework will be developed and made publicly available for use in other applications.

CCS Concepts

• **Computing methodologies** → **Neural networks**; • **Computer systems organization** → **Embedded systems**; **Reliability**.

Keywords

embedded machine learning, neural networks, distributed computing, dependable machine learning

1 Introduction

Artificial Intelligence (AI) has emerged as a major game-changer, bringing notable transformations within Internet of Things (IoT) applications. The increasing number of IoT devices and the enormous amount and diversity of data they generate have made AI an essential tool that enhances IoT applications, from process optimization to patient monitoring solutions.

This research was supported by the Research Foundation - Flanders (FWO) under the SBO grant program 1SH9Y24N.

In recent years, machine learning at the edge has emerged as a viable solution to deploy machine learning in IoT applications and has several advantages over the traditional cloud-centric approach, where data is offloaded to remote computational infrastructure housing the machine learning models. As data is processed directly at the edge devices, edge computing reduces bandwidth usage, decreases communication latency and takes away possible privacy issues [3]. However, the edge environment, especially in IoT applications, consists of low-cost devices that lack the necessary processing power and available memory to run complex machine learning models. In addition, edge devices are often battery-powered and wirelessly connected, creating additional challenges to keep energy consumption low and limit bandwidth usage. Finally, sudden disconnections are not unusual and disrupt the prediction or training process of the models. This research project addresses the current challenges and limitations and has, as a result of this, the potential to advance the current state of the art significantly:

- IoT applications primarily consist of multiple, distributed low-cost sensing devices. The input data is preferably kept on these edge devices, requiring new techniques to distribute neural network models over these devices while considering their limitations.
- Edge devices can fail or suddenly disconnect, disrupting the prediction or training process of the deployed models. Therefore, this research will investigate a novel approach to make neural networks more adaptive to changes in combinations of input, creating a more resilient solution.
- Data drift (temporal changes in data distribution) decreases the prediction performance of deployed models. After deployment, this will be solved by exploring techniques to collaboratively perform the training process over the distributed edge devices within the proposed framework.

2 Limitations of current state-of-the-art techniques

Several techniques exist to address the earlier mentioned challenges on the edge. TinyML has recently gained much popularity in research [3]. This paradigm proposes mechanisms to enable the integration of machine learning on low-cost, low-energy microcontrollers. Most state-of-the-art techniques propose compression to reduce the model size [4]. The authors in [9, 13] use data quantization to reduce the size of the weights and activation. Other

works propose pruning techniques to remove redundancies in over-parameterized networks [7, 10].

However, limited resources on edge devices still restrict the use of complex models and IoT applications often involve multiple devices providing input data requiring a more distributed, collaborative solution [4], [12]. This can be achieved by dividing the machine learning model into smaller segments and distributing the segments amongst the devices. The authors in [5] propose Neurosurgeon, a framework for layer-wise splitting of neural networks between an edge device and an edge server to offload computational tasks from the constrained device. The framework dynamically defines an optimal partition point based on the edge device's inference delay or energy consumption.

Model partitioning allows input data to be kept on the edge devices [12], [8], [1]. However, the training process still requires data to be collected. First proposed by Google Research [6], federated learning enables devices to train a shared model collaboratively without exchanging their private data. The general idea is that each device trains its model with its local data and pushes model updates to a central coordinating server, aggregating the updates into a common model and sending it back to all devices. Nevertheless, implementing federated learning and other distributed learning techniques using constrained edge devices still imposes significant challenges. Indeed, training models is very resource intensive [9] and the exchange of model updates and parameters introduces a communication overhead that is not always feasible in a bandwidth-limited environment.

Despite recent advancements to bring distributed machine learning to the edge, current state-of-the-art still shows shortcomings when applied on the IoT devices at the edge:

- **Partitioning techniques assume input data is coming from one source**
  Prediction models often rely on input features coming from multiple data sources. For example, a production line may have numerous sensing devices distributed over several machines. While model partitioning can keep the input data on edge devices, current partitioning techniques in the literature require that all input data is gathered at a single point before being fed to the partitioned model. This approach results in the need to exchange input data, causing increased communication overhead.
- **Current techniques do not integrate any resiliency against failing edge devices**
  While some techniques rely on redundant devices to keep the inference or learning process operational [6, 11], most do not account for the possibility of device failure which leads to disruptions. This presents a new challenge of maintaining the inference or training process operative while ensuring satisfactory accuracy levels, even when predictions must be made using data from the remaining connected devices (graceful degradation).
- **Most partitioning and distributed learning techniques are not suitable for general-purpose microcontrollers**
  Many studies on distributed neural networks on the edge focus on devices such as single-board computers, mobile phones and FPGAs. However, low-cost devices, such as microcontrollers, are often used in IoT applications to keep the costs as low as possible. Recent advancements in TinyML have demonstrated the possibility of implementing basic machine learning models on these constrained devices, resulting in a research gap in combining TinyML with distributed machine learning and learning at the edge.

3 Methodology

Limitations in current approaches, coupled with the ongoing trend of IoT and machine learning becoming ubiquitous, underscores the necessity for a new approach to distributed AI on the edge to leverage current state-of-the-art. This research project, ***PREDISTINE: A Privacy-Preserving Resilient Framework for Distributed Modular Neural Networks on the Tiny Edge***, aims to develop a novel end-to-end framework for distributing neural networks on the edge, consisting of general-purpose microcontrollers (mainly focusing on class 2 constrained devices [2]). Given sudden losses in connectivity of edge devices are not uncommon, the project will explore modular neural architectures and integrate distributed learning techniques to address data drift, enhancing model performance while ensuring privacy. The subsequent sections elaborate more on the different aspects of the framework.

3.1 Distributed Neural Networks

As mentioned earlier, microcontrollers still limit the implementation of complex neural networks despite current advancements and offloading input data to more powerful devices is not always feasible due to the previously discussed concerns. However, by distributing the neural network across multiple edge devices, their collective computational resources can be utilized, allowing bigger models to be deployed. The distributed architecture ideally needs to be optimized for the prediction task itself, each of the edge devices' hardware constraints and the network topology. This research aims to automate the process of finding such an architecture (Neural Architecture Search), utilizing evolutionary search algorithms to solve this multi-objective optimization problem. This will, consequently, result in a set of Pareto-optimal solutions, each being optimized for a subset of objectives.

3.2 Modular Exchangeable Blocks

The model should be able to continue functioning even during edge device failures or disconnections, which result in only a subset of input features being available. One way to tackle this challenge is to maintain different neural network architectures for each subset of input combinations. However, this approach introduces an enormous storage overhead, making it unfeasible to implement on the edge. Therefore, this project investigates the possibility of reusing certain parts of the model while defining exchangeable blocks to process different combinations of input features. This would achieve a modular architecture that can handle device failures elegantly, albeit with a slight drop in accuracy due to information loss in the input data.

3.3 Distributed Learning

Once a model is deployed on edge devices, it may need to learn from the data in its operational environment. Traditionally, this involves collecting data and labels from the edge, retraining the model on a central server and then redeploying it to the edge devices. However, this approach raises significant privacy concerns as user data must be transmitted to the server. Training on the edge can mitigate this issue and some studies have demonstrated its feasibility [9]. Nonetheless, further research is required to explore how these techniques can be integrated into this framework.

References

[1] Emna Baccour, Naram Mhaisen, Alaa Awad Abdellatif, Aiman Erbad, Amr Mohamed, Mounir Hamdi, and Mohsen Guizani. 2022. Pervasive AI for IoT Applications: A Survey on Resource-Efficient Distributed Artificial Intelligence. *IEEE Communications Surveys & Tutorials* 24, 4 (2022), 2366–2418. https://doi.org/10.1109/COMST.2022.3200740

[2] C. Bormann, M. Ersue, and A. Keranen. 2014. Terminology for Constrained-Node Networks. https://doi.org/10.17487/rfc7228

[3] Jiasi Chen and Xukan Ran. 2019. Deep learning with edge computing: A review. *Proc. IEEE* 107, 8 (2019), 1655–1674.

[4] Carlos Poncinelli Filho, Elias Marques, Victor Chang, Leonardo dos Santos, Flavia Bernardini, Paulo F. Pires, Luiz Ochi, and Flavia C. Delicato. 2022. A Systematic Literature Review on Distributed Machine Learning in Edge Computing. *Sensors* 22, 7 (Jan. 2022), 2665. https://doi.org/10.3390/s22072665 Number: 7 Publisher: Multidisciplinary Digital Publishing Institute.

[5] Yiping Kang, Johann Hauswald, Cao Gao, Austin Rovinski, Trevor Mudge, Jason Mars, and Lingjia Tang. 2017. Neurosurgeon: Collaborative Intelligence Between the Cloud and Mobile Edge. In *Proceedings of the Twenty-Second International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS '17)*. Association for Computing Machinery, New York, NY, USA, 615–629. https://doi.org/10.1145/3037697.3037698

[6] Jakub Konečný, Brendan McMahan, and Daniel Ramage. 2015. *Federated Optimization: Distributed Optimization Beyond the Datacenter.* https://doi.org/10.48550/arXiv.1511.03575 arXiv:1511.03575 [cs, math]

[7] Namhoon Lee, Thalaiyasingam Ajanthan, and Philip HS Torr. 2018. Snip: Single-shot network pruning based on connection sensitivity. *arXiv preprint arXiv:1810.02340* (2018).

[8] He Li, Kaoru Ota, and Mianxiong Dong. 2018. Learning IoT in Edge: Deep Learning for the Internet of Things with Edge Computing. *IEEE Network* 32, 1 (Jan. 2018), 96–101. https://doi.org/10.1109/MNET.2018.1700202

[9] Ji Lin, Ligeng Zhu, Wei-Ming , Wei-Chen Wang, Chuang Gan, and Song Han. 2022. On-device training under 256kb memory. *Advances in Neural Information Processing Systems* 35 (2022), 22941–22954.

[10] Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. 2017. Learning efficient convolutional networks through network slimming. In *Proceedings of the IEEE international conference on computer vision*. 2736–2744.

[11] Jiachen Mao, Xiang Chen, Kent W. Nixon, Christopher Krieger, and Yiran Chen. [n. d.]. MoDNN: Local Distributed Mobile Computing System for Deep Neural Network. In *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017* (2017-03). 1396–1401. https://doi.org/10.23919/DATE.2017.7927211

[12] M. G. Sarwar Murshed, Christopher Murphy, Daqing Hou, Nazar Khan, Ganesh Ananthanarayanan, and Faraz Hussain. 2022. Machine Learning at the Network Edge: A Survey. *Comput. Surveys* 54, 8 (Nov. 2022), 1–37. https://doi.org/10.1145/3469029

[13] Daniele Palossi, Antonio Loquercio, Francesco Conti, Eric Flamand, Davide Scaramuzza, and Luca Benini. 2019. A 64-mW DNN-Based Visual Navigation Engine for Autonomous Nano-Drones. *IEEE Internet of Things Journal* 6, 5 (2019), 8357–8371. https://doi.org/10.1109/JIOT.2019.2917066