TinyMLOps for real-time ultra-low power MCUs applied to frame-based event classification

EuroMLSys 2023

Minh Tri Lê*^, Julyan Arbel*

*TDK InvenSense; ^Inria Grenoble Rhône-Alpes

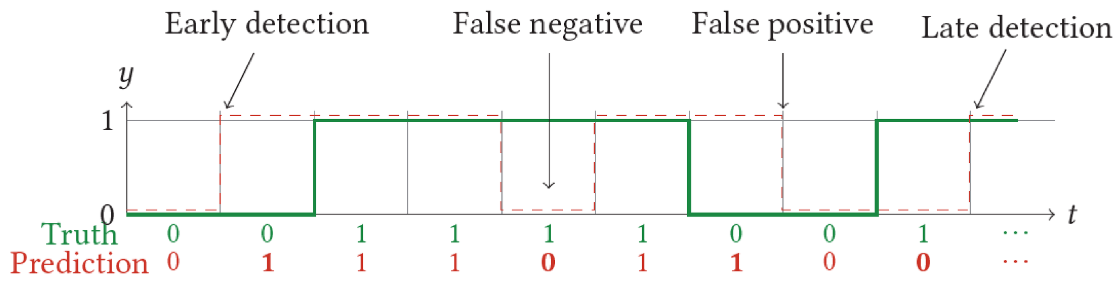
Introduction

TinyML: Emerging field at the intersection of Machine Learning (ML) and Internet of Things (IoT).

- **Why?** Enable intelligent processing of real-time data and close to the source, offers privacy, low-cost systems, new opportunities, …
- **Applications**: Gesture recognition, keyword spotting, anomaly detection, …
- **Challenges**: High power footprint algorithm on ultra-low power microcontrollers: **≤10^3 KB** memory, **10^2 MHz** clock, **1 mW** scale, early stage of the field → ***Lack of mature tools and practices***.
- **TinyMLOps**: Subset of Machine Learning Operations (*MLOps*) focusing on best practices to deploy ML models on low-power embedded systems.

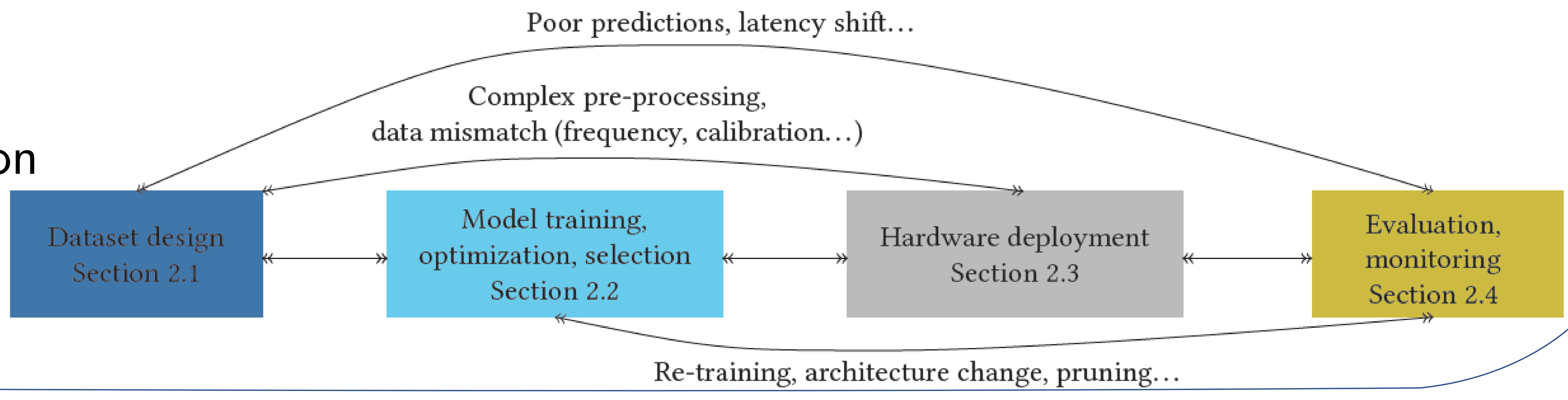
•**Our use case**:

- **Neural networks** on **ultra-low power microcontrollers** for **real-time, always-on event classification**,
- Continuous **frame-by-frame** processing: **1 input data → 1 output decision**.



Problem:

- What are the specific challenges and solutions to design and deploy tinyML solutions ?
- How to apply it our frame-based event classification?



1. Dataset design

Challenges:

- Precise labeling, start/end of an event?
- Build realistic dataset from noisy sensor data.

Solutions:

- Define early/late accept of frames (softer labels)
- Data augmentation (e.g., background noise, shifts, scaling)

2. Model training, optimization

Challenges:

- What model architectures for ultra-low power microcontrollers?
- High power footprint in **computation and size.**
- Limited operations: Integer-only inference, no explicit division ($\neq 2^n$)
- Model has poor performance → Tune hyper-parameters or go back to dataset design.

Solutions:

- Convolutional 1D GRU are polyvalent with good size-performance tradeoff for sequence classification [1]
- Model compression [2]: Pruning, knowledge distillation, low-rank matrix decomposition, weight sharing,
- Quantization [2]: Reduce parameter precision from 32-bits floating point to lower-bit integer (e.g., 8-bits), **mandatory step.**

3. Hardware deployment

Challenges:

- Scarcity of suitable library for model conversion/inference on low-power embedded hardware.
- Heterogeneous hardware landscape

Solutions:

• Frameworks to quantize and deploy deep learning models on MCUs:
  •TensorFlow Lite Micro (TFLM) [3]: Interpreter-based → wide hardware support, good performance, missing some operations (e.g., GRU, Conv1D, …), difficult to customize and debug.
  •NNoM [4]: C code generation, lightweight, wide hardware support, smaller community and adoption, unstable performance results.

Our solution [5]:

•Create our own tinyMLOps framework: bugfix, added missing supported operations or options (GRU, Conv1D), …
•C code generation for wide support, CMSIS-NN support, lightweight.

4. Evaluation

Challenges:

- Finding and measuring meaningful metrics that reflects on-device user experience before release
- Ambiguous errors in frame-based events: early/late detection

Solutions:

- Create custom metrics and tune the optimal model output threshold (softmax/sigmoid input) and plot the results.
- User-define early/late acceptance margin of frame, based on the application: Responsive but lower quality inference vs slower with higher quality inference?

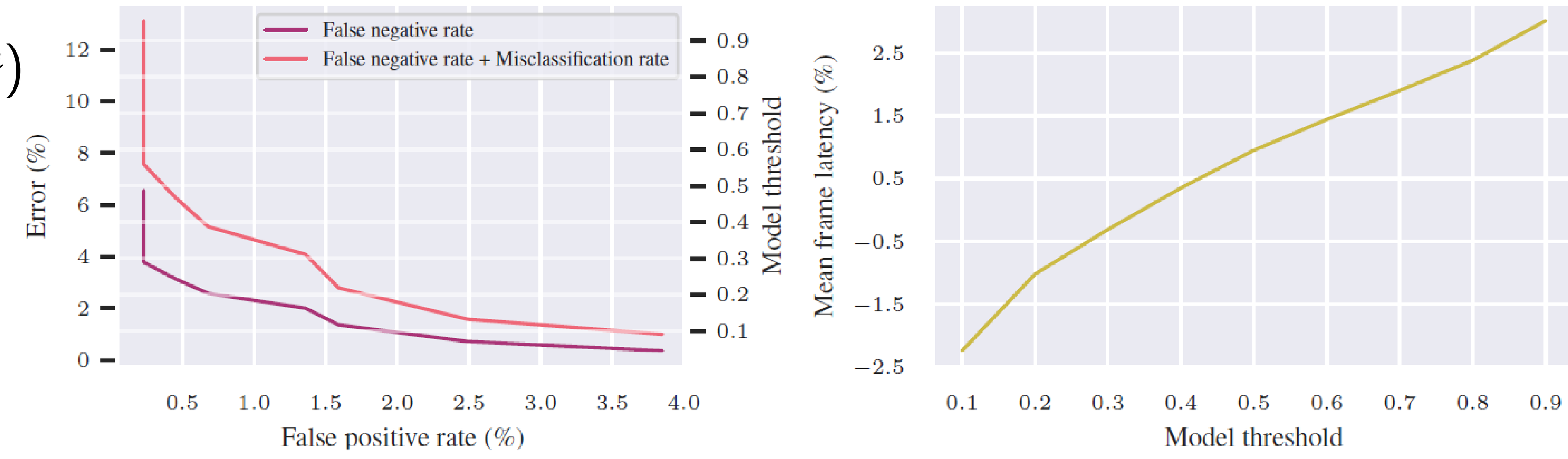
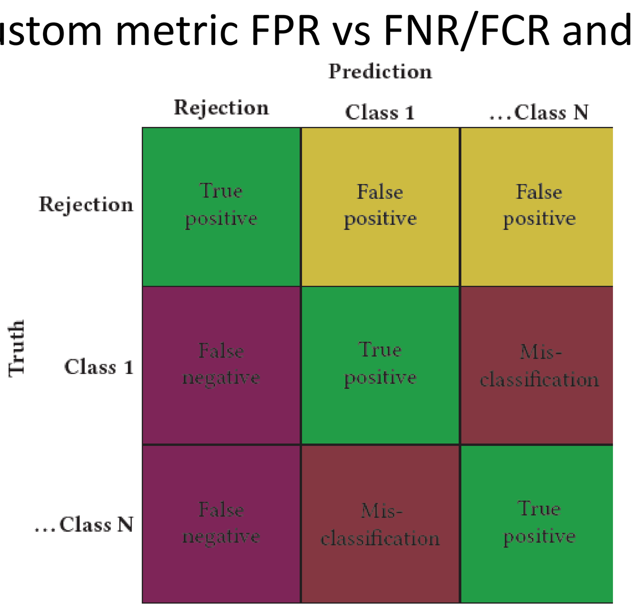


Figure 3: Tuning model output threshold for custom metric FPR vs FNR/FCR and latency.

Table 1: TinyMLOps solutions on an HAR dataset deployed on an Arm Cortex M-4 MCU.

Metric	tf.lite	NNoM	Our
Accuracy (%)	85.5	68.24	86.95
Model size (KB)	6.72	0.29	1.41
Stack size (KB)	12	6.1	5.5
Code memory (KB)	303	16.12	5.5



Conclusion

TinyMLOps: unique set of challenges and solutions, non-linear process and nascent field.
Our tinyMLOps solution: Competitive results with existing solutions, but is more stable and lightweight, while keeping performance
Frame-based event classification: careful consideration on datasets and metrics for real-time inference on ultra-low power microcontrollers.

[1] Roberto Cahuantzi, Xinye Chen, and Stefan Güttel. 2021. A Comparison of LSTM and GRU Networks for Learning Symbolic Sequences.
[2] Yu Cheng, Duo Wang, Pan Zhou, and Tao Zhang. 2020. A Survey of Model Compression and Acceleration for Deep Neural Networks.
[3] Robert David, Jared Duke, Advait Jain, Vijay Janapa Reddi, Nat Jeffries, Jian Li, Nick Kreeger, Ian Nappier, Meghna Natraj, Shlomi Regev, Rocky Rhodes, Tiezhen Wang, and Pete Warden. 2021. TensorFlow Lite Micro: Embedded Machine Learning on TinyML Systems.
[4] Jianjia Ma. 2020. A higher-level Neural Network library on Microcontrollers (NNoM).
[5] Abbas Ataya. 2022. Tiny ML for Tiny Sensors: Waking smarter for less.