Freie Universität Berlin

RIOT Summit
September 18 – 19, 2023

U-TOE - Universal TinyML On-board Evaluation Toolkit for Low-Power IoT

RIOT Summit, 2023

Zhaolan Huang, MSc
Freie Universität Berlin

collaborative work with K. Zandberg, K. Schleiser, and E. Baccelli (Inria)

19.09.2023

AI is invading everything

- Automation, healthcare, financial, cyber-security...
- Become significant components and even the core of systems.

A Push to Offload AI to the Edge

AI is invading everything

- Automation, healthcare, financial, cyber-security...
- Become significant components and even the core of systems.

AI at edge is a trend

For privacy and efficacy reasons, operating AI at the edge of the network (closest to data origin) is more desirable.

- On-site processing of sensor data.
- Reduce latency and communication bandwidth.

Agenda

Freie Universität Berlin

- Crash course
  - (Tiny) Machine Learning
  - Deep Learning: Neural Network
- Challenges and Related Works
  - Challenges in TinyML
  - Related Works
- U-TOE Design and Workflow
  - Architectural Design
  - Workflow using U-TOE
- Preliminary Experimental Results
- Perspectives and Conclusion
  - Perspectives
  - Conclusion

Freie Universität Berlin

- Crash course
  - (Tiny) Machine Learning
  - Deep Learning: Neural Network
- Challenges and Related Works
  - Challenges in TinyML
  - Related Works
- U-TOE Design and Workflow
  - Architectural Design
  - Workflow using U-TOE
- Preliminary Experimental Results
- Perspectives and Conclusion
  - Perspectives
  - Conclusion

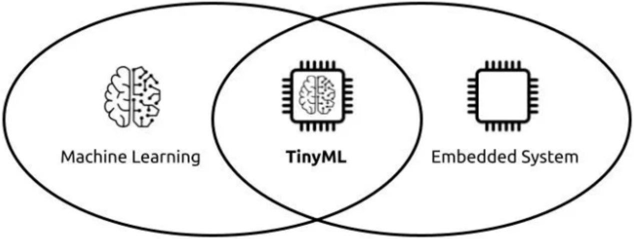
Freie Universität Berlin

Machine Learning (ML)

- Complex, compute-intensive algorithms.
- Data-driven decision making.
- Most popular model: (Deep) Neural Network.

Tiny Machine Learning (TinyML)

- Complex, compute-intensive algorithms.
- Data-driven decision making.
- Most popular model: (Deep) Neural Network.
- Deploy on resource-constrained devices.



TinyML: Machine Learning + Embedded System

ML Model

Computational representation of a real-world process or system

- (Mathematically) A Function with tunable parameters that maps input data to predictions
- Learns from data (<u>Model Training</u>)
- A trained neural network is a ML model

ML Model

Computational representation of a real-world process or system

- (Mathematically) A Function with tunable parameters that maps input data to predictions
- Learns from data (Model Training)
- A trained neural network is a ML model

Training and Inference

- Training: Modifying model's parameters based on numerous data to approximate real-world process
- Inference: Using a trained model to make predictions or decisions on new, unseen data

ML Model

Computational representation of a real-world process or system

- (Mathematically) A Function with tunable parameters that maps input data to predictions
- Learns from data (Model Training)
- A trained neural network is a ML model

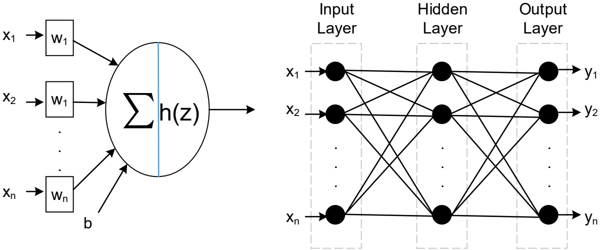
Training and Inference

- Training: Modifying model's parameters based on numerous data to approximate real-world process
- Inference: Using a trained model to make predictions or decisions on new, unseen data
- U-TOE focuses model inference on low-power devices.

Freie Universität Berlin

- Crash course
  - (Tiny) Machine Learning
  - Deep Learning: Neural Network
- Challenges and Related Works
  - Challenges in TinyML
  - Related Works
- U-TOE Design and Workflow
  - Architectural Design
  - Workflow using U-TOE
- Preliminary Experimental Results
- Perspectives and Conclusion
  - Perspectives
  - Conclusion

Layer-wise (non-linear) function composition



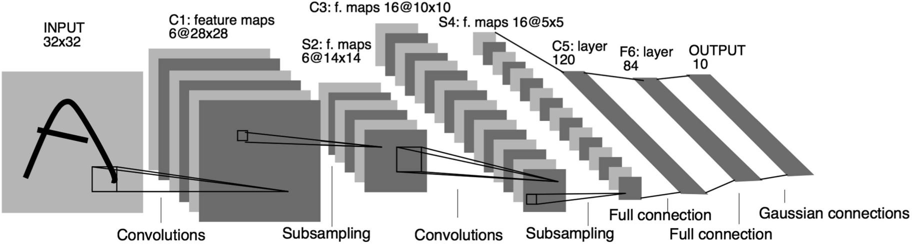
(a) Neuron (b) Neural Network

Neuron	Layer	Network
$z = w^T x + b$	$Z_L = Wx + B$	$Y = H^N(H^{N-1}(...H^1(Z_1)...))$
$y = h(z)$	$Y_L = H(Z)$	$N = 3$, Multilayer perceptron $N > 10$, Deep Learning

Non-Linear: $h(z), H(Z)$

Neuron	Layer	Network
$z = w^T * x + b$	$Z_L = W * x + B$	$Y = H^N(H^{N-1}(...H^1(Z_1)...))$
$y = h(z)$	$Y_L = H(Z)$	$N = 3$, Multilayer perceptron $N > 10$, Deep Learning

Convolutional Neural Network



Freie Universität Berlin

Model Building Blocks: Operators

- Affine Transformations (z): Convolution, matrix multiplication, addition...
  Multiplication: $Z_L = Wx + B$, $\mathcal{O}(MN)$, with $W : MxN$
  2D-Convolution: $Z_L = W * X + B$, $\mathcal{O}(N^2K^2)$, with $W : KxK, X : NxN$
  In practice: $K = 1, 3, 5, 7$
  $\rightarrow$ Compute-intensive in order of input dimension N

Model Building Blocks: Operators

- Affine Transformations (z): Convolution, matrix multiplication, addition...
  Multiplication: $Z_L = Wx + B$, $\mathcal{O}(MN)$, with $W : MxN$
  2D-Convolution: $Z_L = W * X + B$, $\mathcal{O}(N^2 K^2)$, with $W : KxK, X : NxN$
  In practice: $K = 1, 3, 5, 7$
  $\rightarrow$ Compute-intensive in order of input dimension N
- Non-linear Operators (h(z)): Pooling, activation functions, (batch) normalization, dropout, quantization...

Major ML Frameworks

Tensorflow (Google), PyTorch (Meta AI & Linux Foundation), Keras, MXNet...Used for building neural network models in few lines

Agenda

Freie Universität Berlin

- Crash course
  - (Tiny) Machine Learning
  - Deep Learning: Neural Network
- Challenges and Related Works
  - Challenges in TinyML
  - Related Works
- U-TOE Design and Workflow
  - Architectural Design
  - Workflow using U-TOE
- Preliminary Experimental Results
- Perspectives and Conclusion
  - Perspectives
  - Conclusion

So, that elephant will be stuffed into tiny devices...

Yes,

ChatGPT

~175 Billion Parameters
Training on ~10,000 Nvidia GPUs

But

264KB Memory

So, that elephant will be stuffed into tiny devices...

- Resource Constraints: Processor(s), storage, memory.
- Real-time Processing: Real-time inference in critical applications.
- Power Efficiency: Do FAST, sleep more.
- Model Size: Prototype and optimize neural networks under resource budget within multiple iterations.

So, that elephant will be stuffed into tiny devices...

- Resource Constraints: Processor(s), storage, memory.
- Real-time Processing: Real-time inference in critical applications.
- Power Efficiency: Do <u>FAST</u>, sleep more.
- Model Size: Prototype and optimize neural networks under resource budget within multiple iterations.

Problem Statement

Thus, we need a toolkit for

- Model Evaluation: Consumption of resources
- Bottleneck Location: Know where to shape
- Hardware Selection: Provide MCU candidates

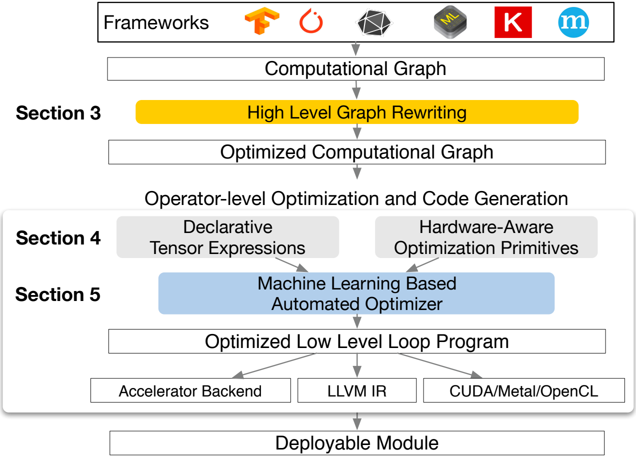
Agenda

- Crash course
  - (Tiny) Machine Learning
  - Deep Learning: Neural Network
- Challenges and Related Works
  - Challenges in TinyML
  - Related Works
- U-TOE Design and Workflow
  - Architectural Design
  - Workflow using U-TOE
- Preliminary Experimental Results
- Perspectives and Conclusion
  - Perspectives
  - Conclusion

Freie Universität Berlin

- Model Compilation
- Model Profilers
- Benchmarking Suites and TinyML Benchmarks
- Low-power IoT Platform and Testbeds

- Model Compilation: (micro) TVM



[Chen et.al arXiv:1802.04799 [cs.LG]]

Related Works

- Model Profilers
  - Internal tools of major ML frameworks (Tensorflow, Pytorch, MXNet...): merely support on various IoT boards.
  - ML-EXray: Easy to use, but not support IoT boards.

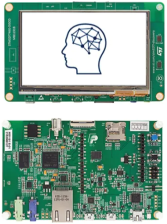
Related Works

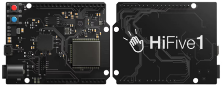
- Benchmarking Suites and TinyML Benchmarks
  - MLPerf Tiny: Standard benchmark suite with representative ML models.
  - Prior TinyML benchmarks focuses on comparison of specific frameworks on specific boards for specific tasks.

Related Works

Freie Universität Berlin

- Low-power IoT Platform and Testbed





STM32F746g-disco	Raspberry Pi Pico	HiFive1 Rev B
216 MHz	125 MHz	320 MHz
340 KB RAM	264 KB RAM	16 KB RAM

After reviewing prior work, we still can’t conveniently evaluate customized models from arbitrary ML frameworks on arbitrary low-power IoT boards, there is a gap from ML models to boards.

Agenda

Freie Universität Berlin

- Crash course
  - (Tiny) Machine Learning
  - Deep Learning: Neural Network
- Challenges and Related Works
  - Challenges in TinyML
  - Related Works
- U-TOE Design and Workflow
  - Architectural Design
  - Workflow using U-TOE
- Preliminary Experimental Results
- Perspectives and Conclusion
  - Perspectives
  - Conclusion

The goals of U-TOE are automatically compressing, flashing and evaluating arbitrary models on arbitrary commercial off-the-shelf low-power boards.

Performance Metrics

- Memory (RAM) Consumption
- Storage (Flash) Consumption
- Computational Latency

The goals of U-TOE are automatically compressing, flashing and evaluating arbitrary models on arbitrary commercial off-the-shelf low-power boards.

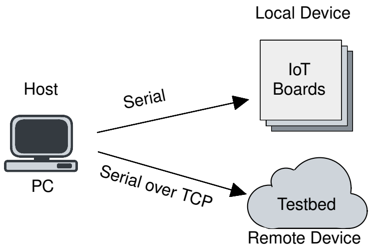
Performance Metrics

- Memory (RAM) Consumption
- Storage (Flash) Consumption
- Computational Latency

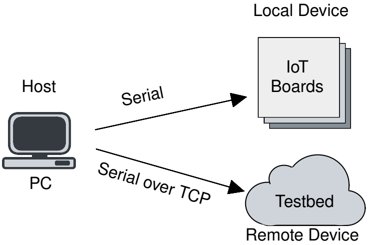
Granularity

- Per-Model Evaluation
- Per-Operator Evaluation

Architectural Design



(a) Hardware Configuration

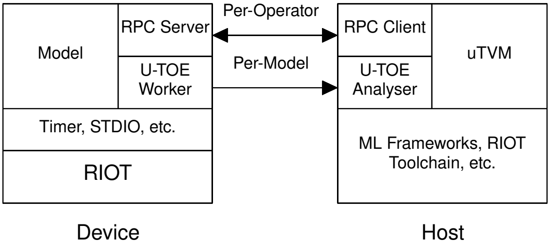


(a) Hardware Configuration

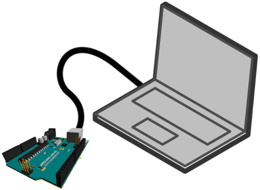
You don’t have boards in hand?

No Problem! Try out remote boards on FIT IoT-LAB Testbed!

Architectural Design



(b) Software Architecture



Agenda

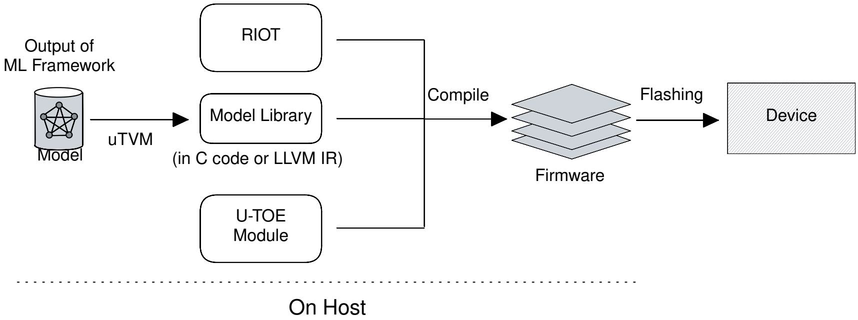
Freie Universität Berlin

- Crash course
  - (Tiny) Machine Learning
  - Deep Learning: Neural Network
- Challenges and Related Works
  - Challenges in TinyML
  - Related Works
- U-TOE Design and Workflow
  - Architectural Design
  - Workflow using U-TOE
- Preliminary Experimental Results
- Perspectives and Conclusion
  - Perspectives
  - Conclusion

From NN models to boards...

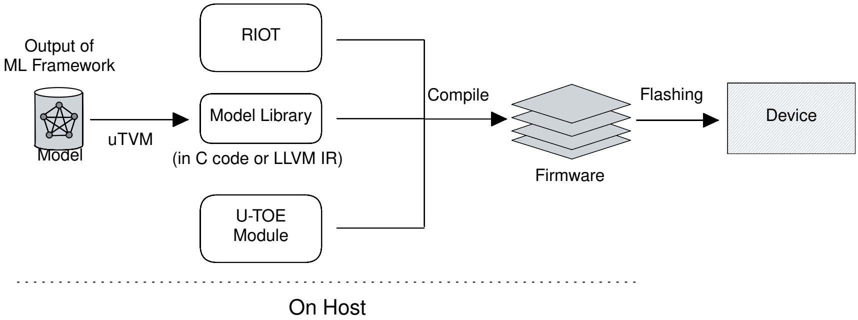
Freie Universität Berlin

From NN models to boards...



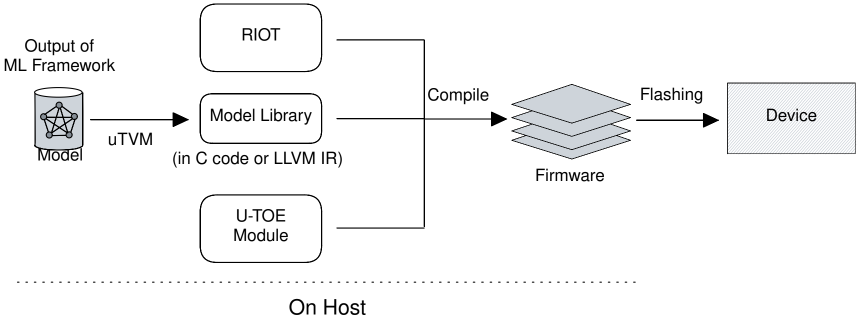
1. TVM translates NN model into C / LLVM IR.

From NN models to boards...



1. TVM translates NN model into C / LLVM IR.
2. Co-compile with RIOT and U-TOE module.

From NN models to boards...



1. TVM translates NN model into C / LLVM IR.
2. Co-compile with RIOT and U-TOE module.
3. Flash to board and log back performance metrics.

Agenda

Freie Universität Berlin

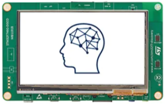
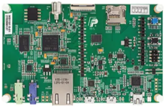
- Crash course
  - (Tiny) Machine Learning
  - Deep Learning: Neural Network
- Challenges and Related Works
  - Challenges in TinyML
  - Related Works
- U-TOE Design and Workflow
  - Architectural Design
  - Workflow using U-TOE
- Preliminary Experimental Results
- Perspectives and Conclusion
  - Perspectives
  - Conclusion

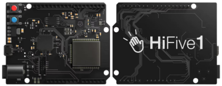
Preliminary Experimental Results

- Model Zoo: Quantized to INT8.

Model (# Parameters)	Task	Remarks
LeNet-5 (~40K)	Image Classification	-
MobileNetV1 (~500K)	Visual Wake Words	With width multiplier 0.25
DS-CNN Small (~22K)	Keyword Spotting	Depthwise separable CNN
Deep AutoEncoder (~264K)	Anomaly Detection	-
RNNoise (~87K)	Noise Suppression	GRU-based network
Sinus (~0.30K)	Regression	TFLite sine value example

- MCU Zoo: ARM Cortex M0+, M3, M4, M7 and RISC-V

Evaluation results of LeNet5 on various IoT boards.

Board	Core	Memory	Storage	Latency
arduino-zero	M0+ @ 48 MHz	11.292	64.940	182.068
rpi-pico	M0+ @ 125 MHz	28.704	109.504	70.117
openmote-b	M3 @ 32 MHz	11.100	66.080	200.367
IoT-LAB M3	M3 @ 72 MHz	11.296	62.260	97.751
nucleo-wl55jc	M4 @ 48 MHz	11.288	63.180	98.661
nrf52840dk	M4 @ 64 MHz	11.348	61.332	66.088
b-l475e-iot01a	M4 @ 80 MHz	11.288	61.604	52.901
stm32f746g-disco	M7 @ 216 MHz	11.076	64.712	39.601
esp32c3-devkit	RISC-V @ 80 MHz	258.874	222.272	54.953
sipeed-longan-nano	RISC-V @ 108 MHz	103.108	106.422	37.789
hifive1b	RISC-V @ 320 MHz	60.884	66.492	153.747

Memory and storage consumption in KB, computational latency in ms.

Freie Universität Berlin

Evaluation of various models on stm32f746-disco board.

Model	Task	Memory	Storage	Latency
DS-CNN Small	Keyword Spotting	68.992	71.796	461.396
MobileNetV1-0.25x	Visual Wake Words	185.352	491.668	1435.938
LeNet-5	Image Classification	12.068	65.851	39.601
Deep AutoEncoder	Anomaly Detection	6.532	292.696	35.638
RNNoise	Noise Suppression	4.688	119.652	12.154

Memory and storage consumption in KB, computational latency in ms.

Evaluation of various models on stm32f746-disco board.

Model	Task	Memory	Storage	Latency
DS-CNN Small	Keyword Spotting	68.992	71.796	461.396
MobileNetV1-0.25x	Visual Wake Words	185.352	491.668	1435.938
LeNet-5	Image Classification	12.068	65.851	39.601
Deep AutoEncoder	Anomaly Detection	6.532	292.696	35.638
RNNoise	Noise Suppression	4.688	119.652	12.154

Memory and storage consumption in KB, computational latency in ms.

Per-Operator Evaluation Output of TFlite sinus model.

Ops	Latency	Latency (%)	Asso. Params	Memory	Storage
add_nn_relu	8.856	15.22%	p0, p1	0.128	0.128
add_nn_relu_1	46.682	80.23%	p2, p3	0.128	1.088
add	2.646	4.54%	p4, p5	0.068	0.068

Memory and storage consumption in KB, computational latency in us.

Now, we successfully built a generic solution for performance evaluation of neural network models on various IoT boards, but it still lack of...

Freie Universität Berlin

- Crash course
  - (Tiny) Machine Learning
  - Deep Learning: Neural Network
- Challenges and Related Works
  - Challenges in TinyML
  - Related Works
- U-TOE Design and Workflow
  - Architectural Design
  - Workflow using U-TOE
- Preliminary Experimental Results
- Perspectives and Conclusion
  - Perspectives
  - Conclusion

Perspectives: What's Next?

- Further Development & Community Support

Perspectives: What's Next?

- Further Development & Community Support
  - (Ongoing) GUI for user-friendly interaction

Perspectives: What's Next?

- Further Development & Community Support
  - (Ongoing) GUI for user-friendly interaction
  - Co-evolve with RIOT hardware support and OS functionalities, potentially as RIOT pkg

Perspectives: What's Next?

- Further Development & Community Support
  - (Ongoing) GUI for user-friendly interaction
  - Co-evolve with RIOT hardware support and OS functionalities, potentially as RIOT pkg
  - Numerical issue in TVM: https://github.com/apache/tvm/issues/15285 (Keyword: *inconsistent results*), need interaction with TVM community.

Perspectives: What's Next?

Freie Universität Berlin

- Further Development & Community Support
  - (Ongoing) GUI for user-friendly interaction
  - Co-evolve with RIOT hardware support and OS functionalities, potentially as RIOT pkg
  - Numerical issue in TVM: https://github.com/apache/tvm/issues/15285 (Keyword: *inconsistent results*), need interaction with TVM community.
- Extended ML Support

Perspectives: What's Next?

- Further Development & Community Support
  - (Ongoing) GUI for user-friendly interaction
  - Co-evolve with RIOT hardware support and OS functionalities, potentially as RIOT pkg
  - Numerical issue in TVM: https://github.com/apache/tvm/issues/15285 (Keyword: *inconsistent results*), need interaction with TVM community.
- Extended ML Support
  - Support on-device learning scenario

Perspectives: What's Next?

Freie Universität Berlin

- Further Development & Community Support
  - (Ongoing) GUI for user-friendly interaction
  - Co-evolve with RIOT hardware support and OS functionalities, potentially as RIOT pkg
  - Numerical issue in TVM: https://github.com/apache/tvm/issues/15285 (Keyword: *inconsistent results*), need interaction with TVM community.
- Extended ML Support
  - Support on-device learning scenario
  - Support ML models other than neural network

Perspectives: What's Next?

- Further Development & Community Support
  - (Ongoing) GUI for user-friendly interaction
  - Co-evolve with RIOT hardware support and OS functionalities, potentially as RIOT pkg
  - Numerical issue in TVM: https://github.com/apache/tvm/issues/15285 (Keyword: *inconsistent results*), need interaction with TVM community.
- Extended ML Support
  - Support on-device learning scenario
  - Support ML models other than neural network
  - Generalize to compute-intensive tasks

Compile, link, flash and execute U-TOE for model Sinus on FIT IoT-lab testbed.

```
2023-06-07 14:13:15,449 # main(): This is RIOT! (Version: 9515d-wip/utvm)
2023-06-07 14:13:15,452 # U-TOE Per-Model Evaluation
2023-06-07 14:13:15,454 # Press any key to start >
2023-06-07 14:13:17,149 # trial: 0, usec: 154938, ret: 0
2023-06-07 14:13:17,305 # trial: 1, usec: 153900, ret: 0
2023-06-07 14:13:17,461 # trial: 2, usec: 153748, ret: 0
2023-06-07 14:13:17,617 # trial: 3, usec: 153717, ret: 0
2023-06-07 14:13:17,773 # trial: 4, usec: 153717, ret: 0
2023-06-07 14:13:17,929 # trial: 5, usec: 153717, ret: 0
2023-06-07 14:13:18,085 # trial: 6, usec: 153717, ret: 0
2023-06-07 14:13:18,241 # trial: 7, usec: 153748, ret: 0
2023-06-07 14:13:18,397 # trial: 8, usec: 153717, ret: 0
2023-06-07 14:13:18,553 # trial: 9, usec: 153717, ret: 0
2023-06-07 14:13:18,555 # Evaluation finished >
```

Agenda

Freie Universität Berlin

- Crash course
  - (Tiny) Machine Learning
  - Deep Learning: Neural Network
- Challenges and Related Works
  - Challenges in TinyML
  - Related Works
- U-TOE Design and Workflow
  - Architectural Design
  - Workflow using U-TOE
- Preliminary Experimental Results
- Perspectives and Conclusion
  - Perspectives
  - Conclusion

Conclusion

- Provided open-source, generic model-to-board evaluation solution.
- Provided comparative experimental benchmarks using U-TOE, reproducible both on an openaccess IoT testbed and on PC.

Thanks! And Questions?

Code:
https://github.com/zhaolanhuang/U-TOE

arXiv:
https://arxiv.org/abs/2306.14574

E-Mail: zhaolan.huang@fu-berlin.de

If you want to cite this work, please use:
Z. Huang, K. Zandberg, K. Schleiser, E. Baccelli. U-TOE: Universal TinyML On-board Evaluation Toolkit for Low-Power IoT. In Proc. of 12th IFIP/IEEE PEMWN, Sept. 2023.

Acknowledgment: We would like to thank Cedric Adjih and Nadjib Achir for useful discussions and suggestions. The research partly received funding from the MESRI-BMBF German/French cybersecurity program under grant agreements No. ANR-20-CYAL-0005 and 16KIS1395K. This work reflects only the authors' views.