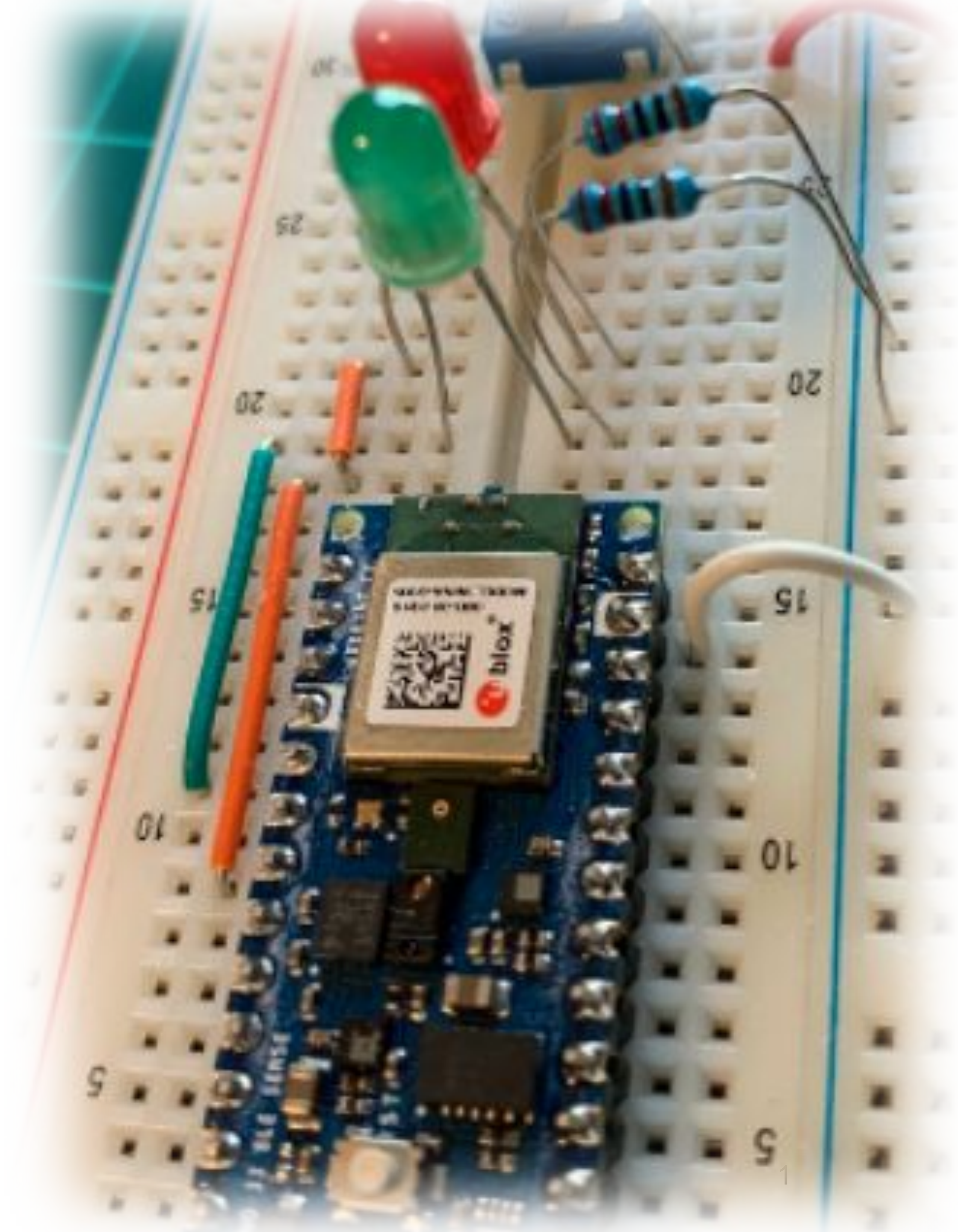
IESTI01 – TinyML

Embedded
Machine Learning

1. About the Course & Syllabus
(Introductory On-Line Class)

Prof. Marcelo Rovai
UNIFEI



Dear students,

Welcome to the first class of the IESTI01 (TinyML) EAD course. I am Professor Marcelo Rovai, a former student here at UNIFEI, and I am pleased to be with you this semester.

IESTI01 is a discipline that mixes Machine Learning (part of Artificial Intelligence) with small devices, such as microcontrollers and sensors, whose main characteristics are ultra-low power consumption, usually 32-bit CPUs, and a few kilobytes of memory.

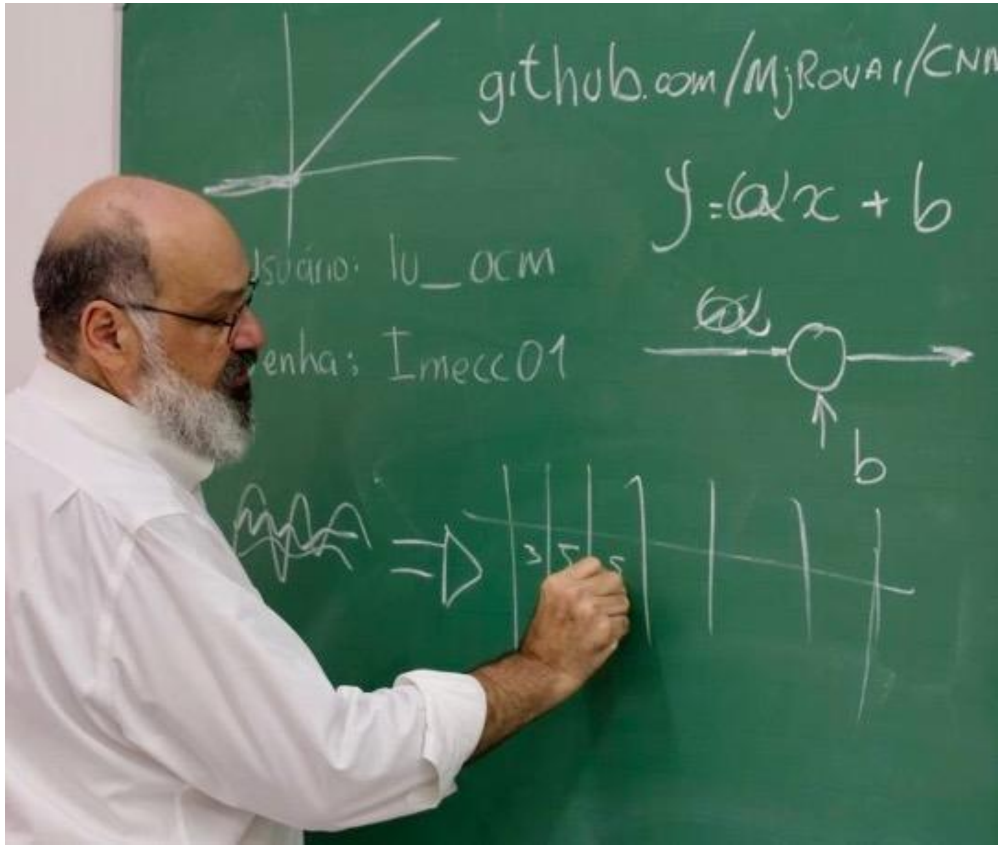
We understand that the explosive growth of Machine Learning, the ease of use of software development platforms such as TensorFlow (TF) and PyTorch, which are based on the Python language, and the current generation of powerful microcontrollers make TinyML an indispensable topic of study for Engineering students in Electronics, Computing, and Control and automation.

This mix of expertise and the pioneering nature of this discipline (We were the second university worldwide to have this type of course, Harvard School of Engineering being the first) leads us to significant challenges concerning the necessary basis for the minimum understanding of the matter. Thus, the time we have available for the course is short. So, we must commit to following the weekly classes and doing complementary activities such as readings, laboratories, and assignments. You can count on me to clarify doubts and to review the necessary concepts for a good understanding of the course (For that, use the FORUM, available in each class, or e-mail me).

Greetings from the south of the world!

Prof. Marcelo Rovai

Marcelo Rovai is an educator and professional in the field of engineering and technology, holding the title of Professor Honoris Causa from the Federal University of Itajubá, Brazil. His educational background includes an Engineering degree from UNIFEI and an advanced specialization from the Polytechnic School of São Paulo University (POLI/USP). Further enhancing his expertise, he earned an MBA from IBMEC (INSPER) and a Master's in Data Science from the Universidad del Desarrollo (UDD) in Chile.



With a career spanning several high-profile technology companies such as AVIBRAS Airspace, AT&T, NCR, and IGT, where he served as Vice President for Latin America, he brings a wealth of industry experience to his academic endeavors. He is a prolific writer on electronics-related topics and shares his knowledge through open platforms like Hackster.io.

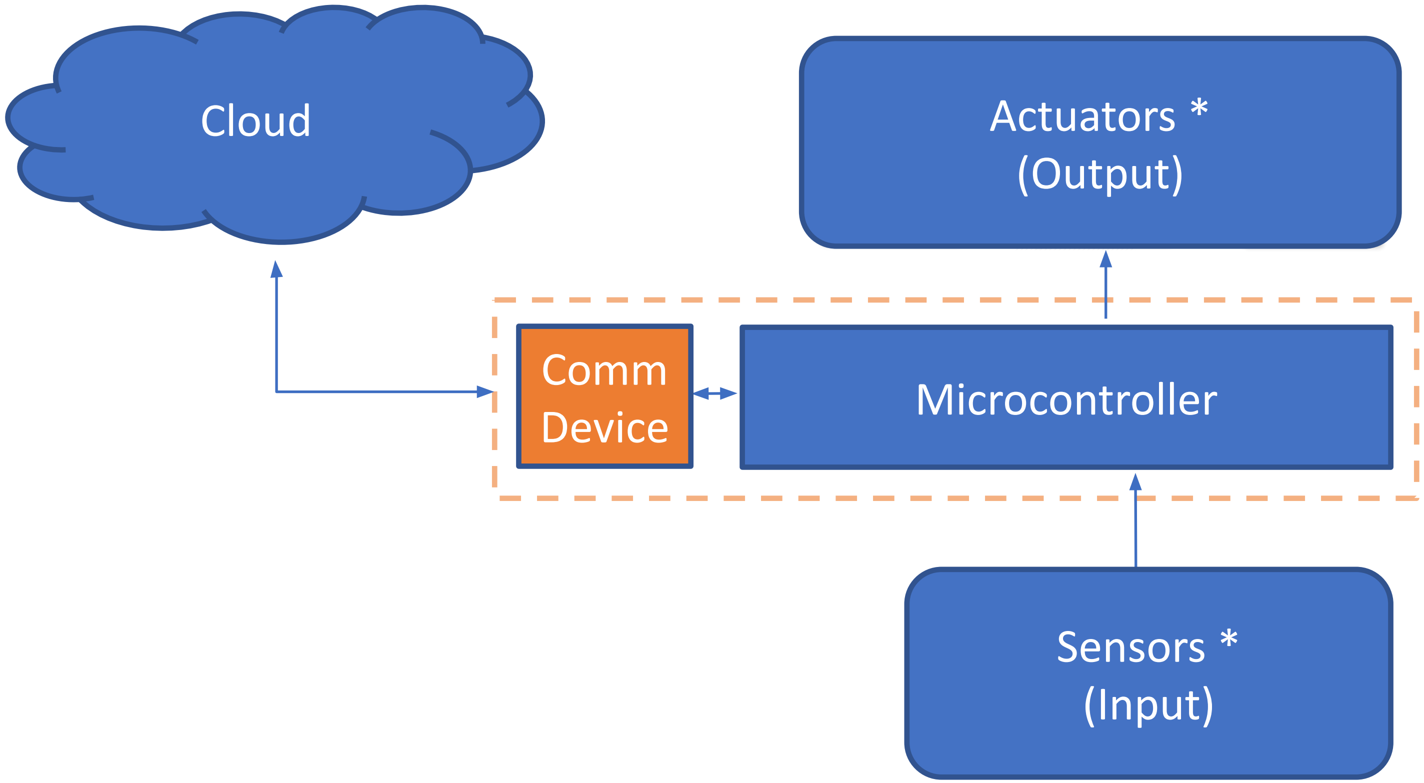
In addition to his professional pursuits, he is dedicated to educational outreach, serving as a volunteer professor at the IESTI (UNIFEI) and engaging with the TinyML4D group and the EDGE AIP – the Academia-Industry Partnership of EDGEAI Foundation as a Co-Chair, promoting EdgeAI education in developing countries. His work underscores a commitment to leveraging technology for societal advancement.

“**Edge AI** is a truly complete technology. As a topic, it makes use of knowledge from everything from the physical properties of semiconductor electronics all the way up to the engineering of high-level architectures that span devices and the cloud. It demands expertise in the most cutting-edge approaches to artificial intelligence and machine learning along with the most venerable skills of bare-metal embedded software engineering. It makes use of the entire history of computer science and electrical engineering, laid out end to end.”

Situnayake, Daniel; Plunkett, Jenny
AI at the Edge (pp. 215-216)
O'Reilly Media

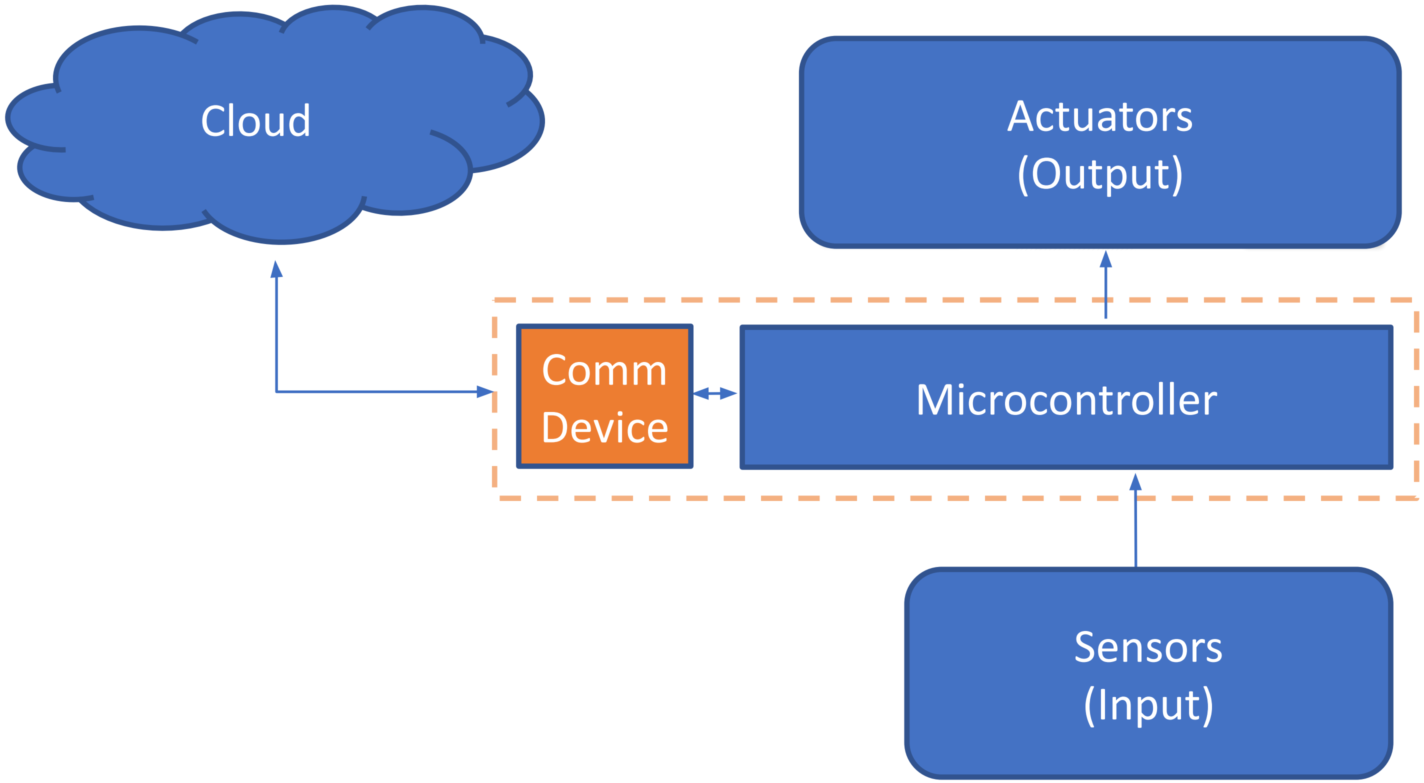
Internet of Things (IoT)

Typical IoT Project



* “Things”

Typical IoT Project



5 Quintillion

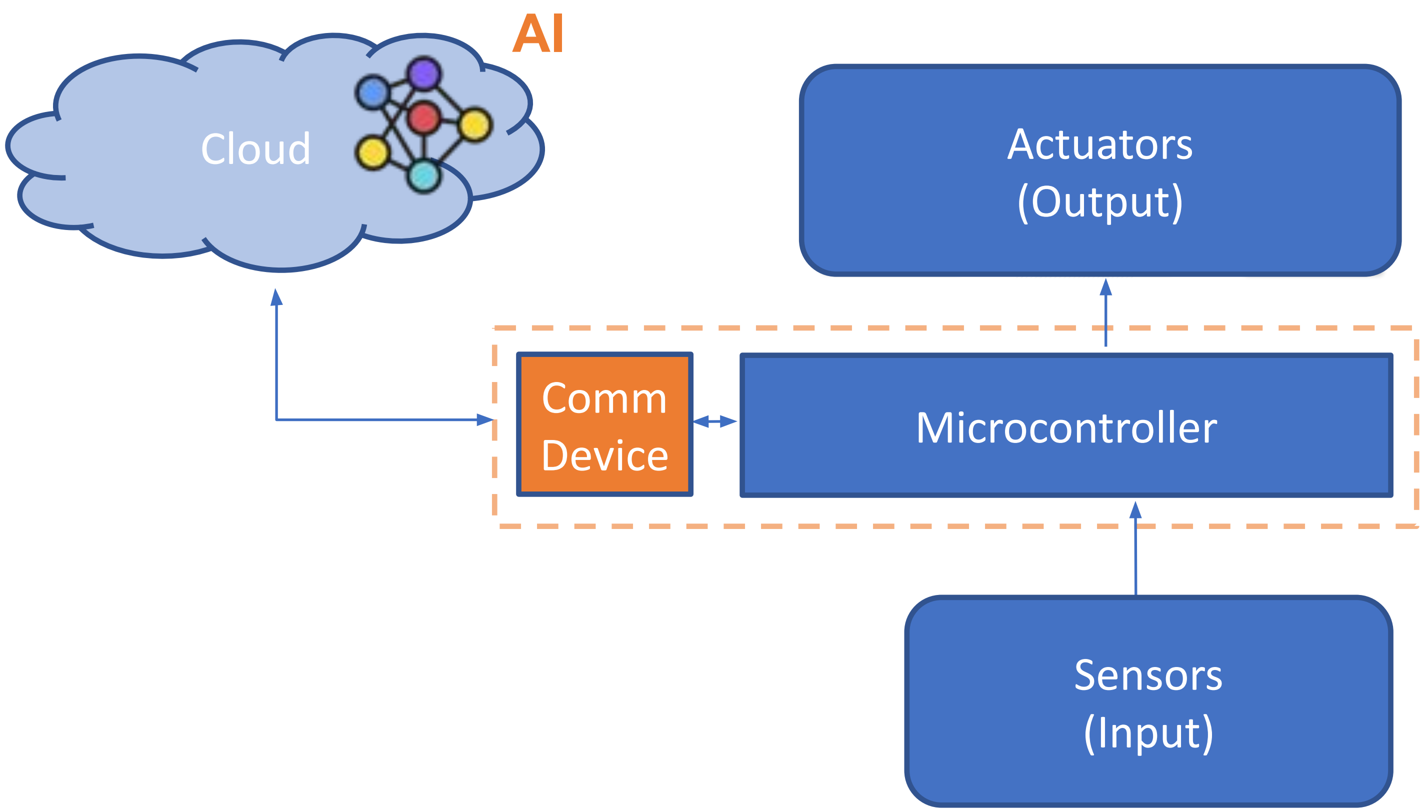
bytes of data produced every day by IoT

<1%

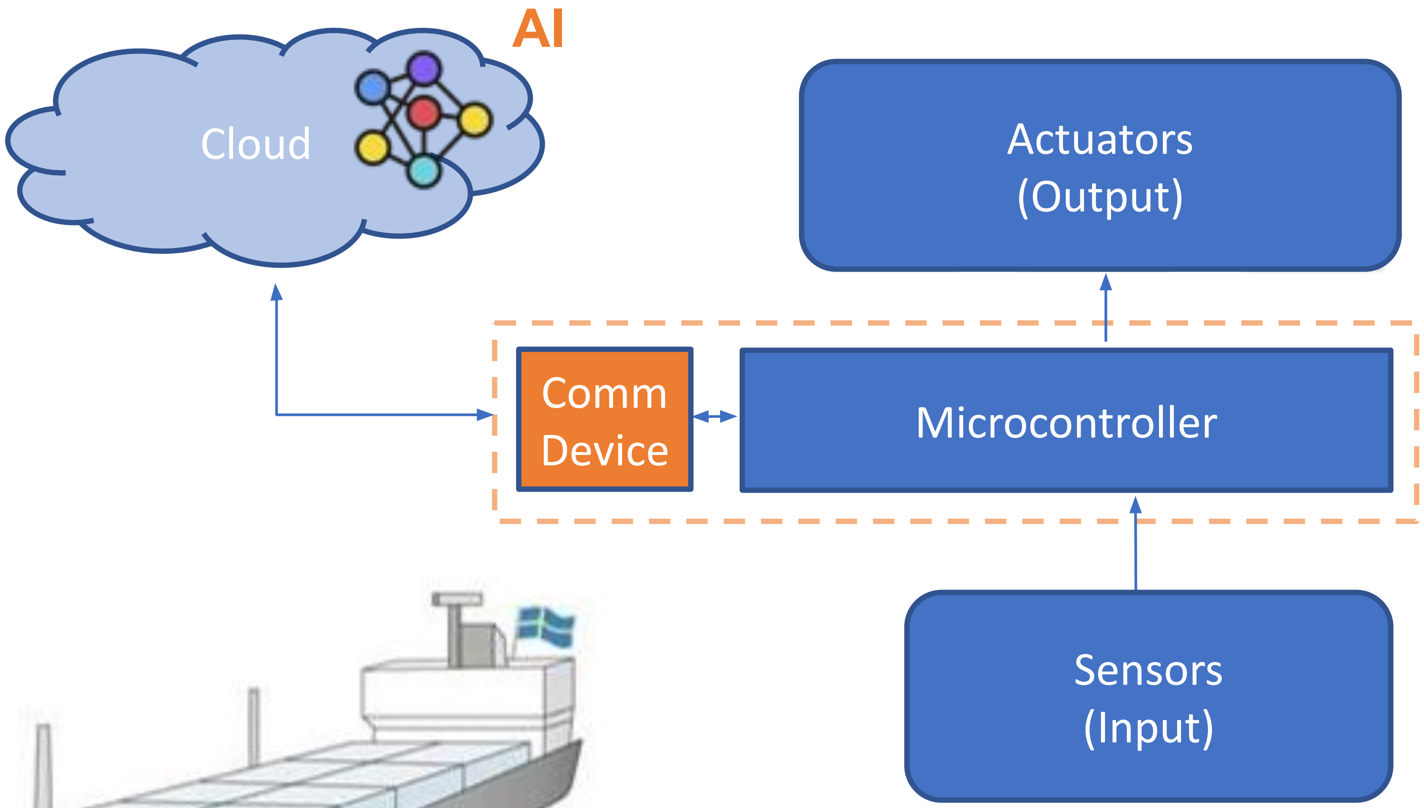
of unstructured data is analyzed or used at all

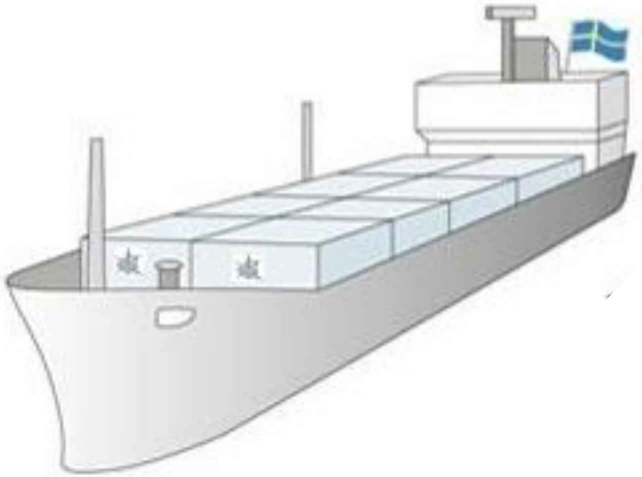
Source: Harvard Business Review, [What's Your Data Strategy?](), April 18, 2017
Cisco, [Internet of Things (IoT) Data Continues to Explode Exponentially. Who Is Using That Data and How?](), Feb 5, 2018

Typical AIoT Project



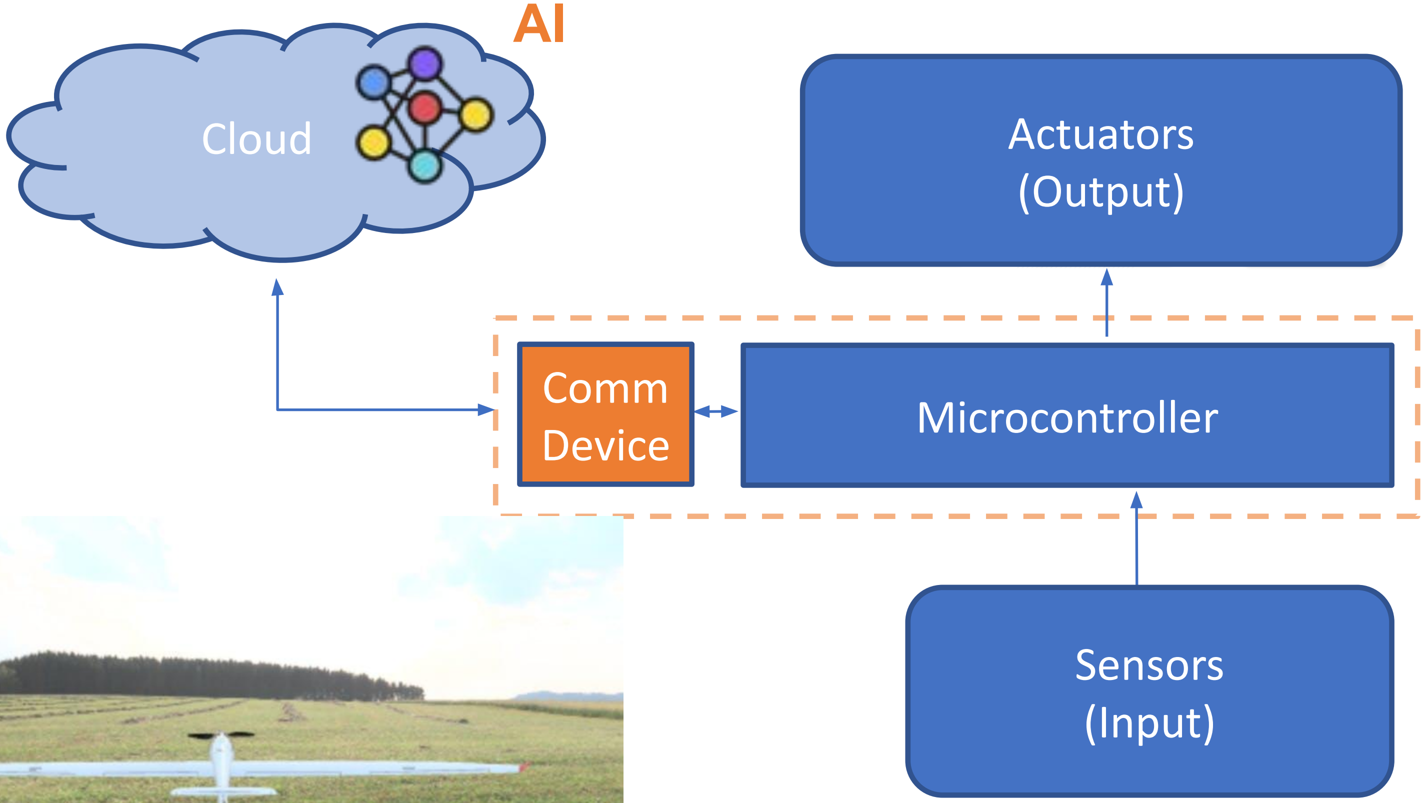
Typical AIoT Project … … Issues





Bandwidth

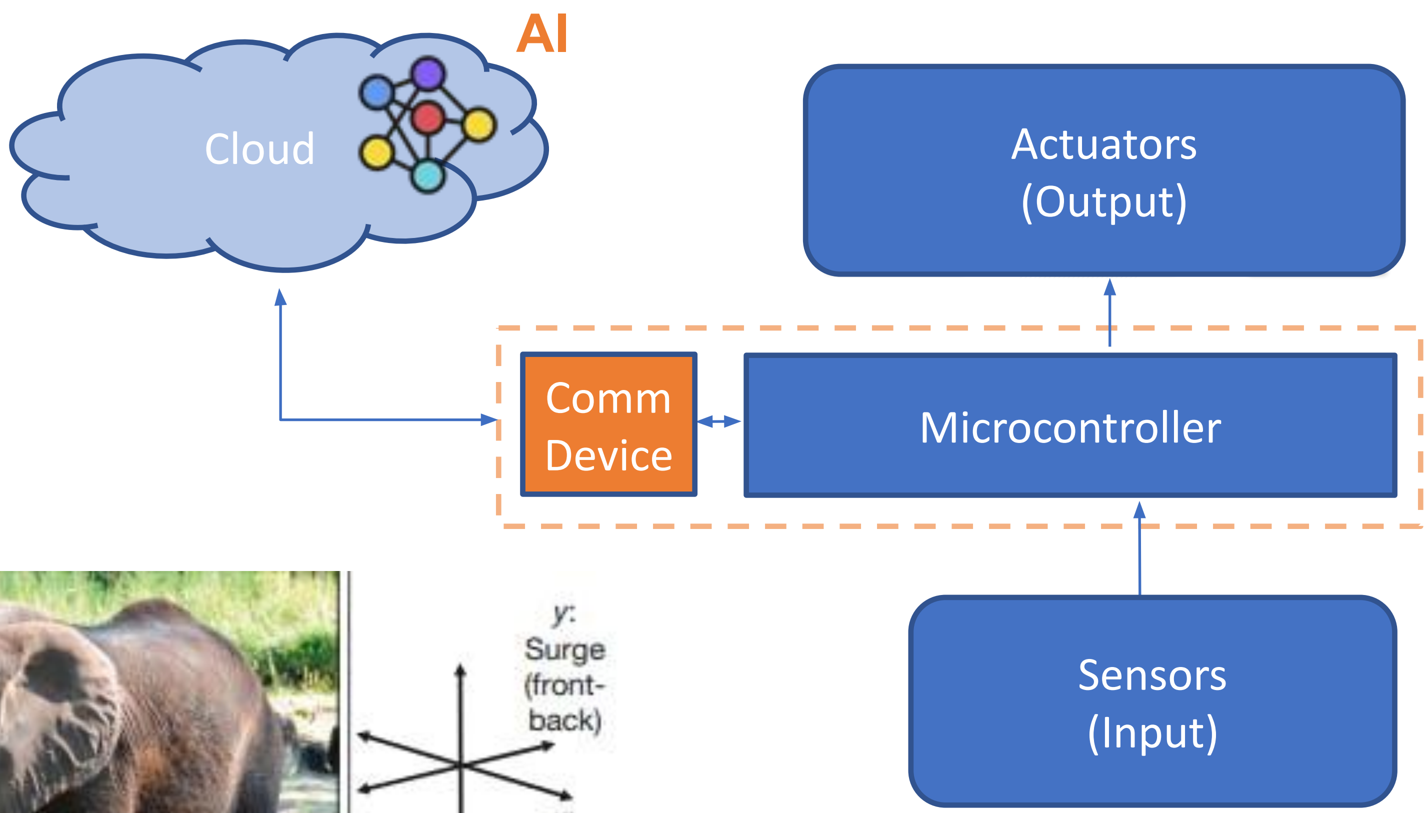
Typical AIoT Project … … Issues

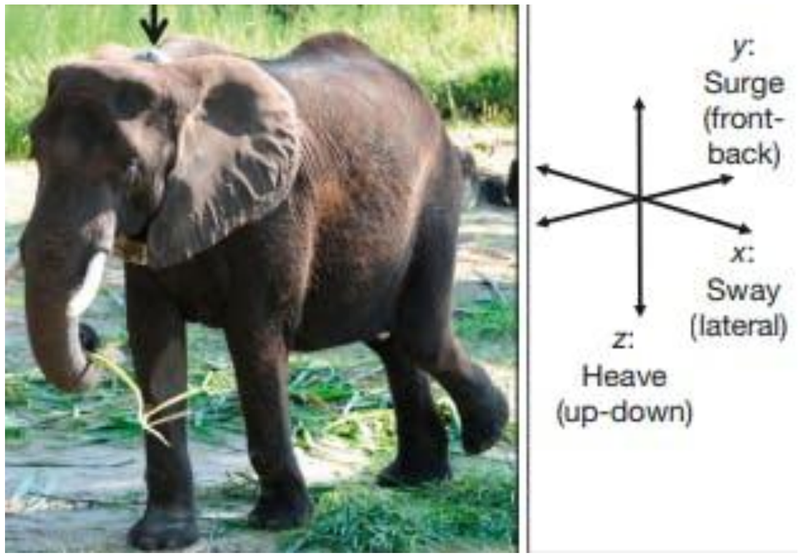


Bandwidth

Latency

Typical AIoT Project … … Issues



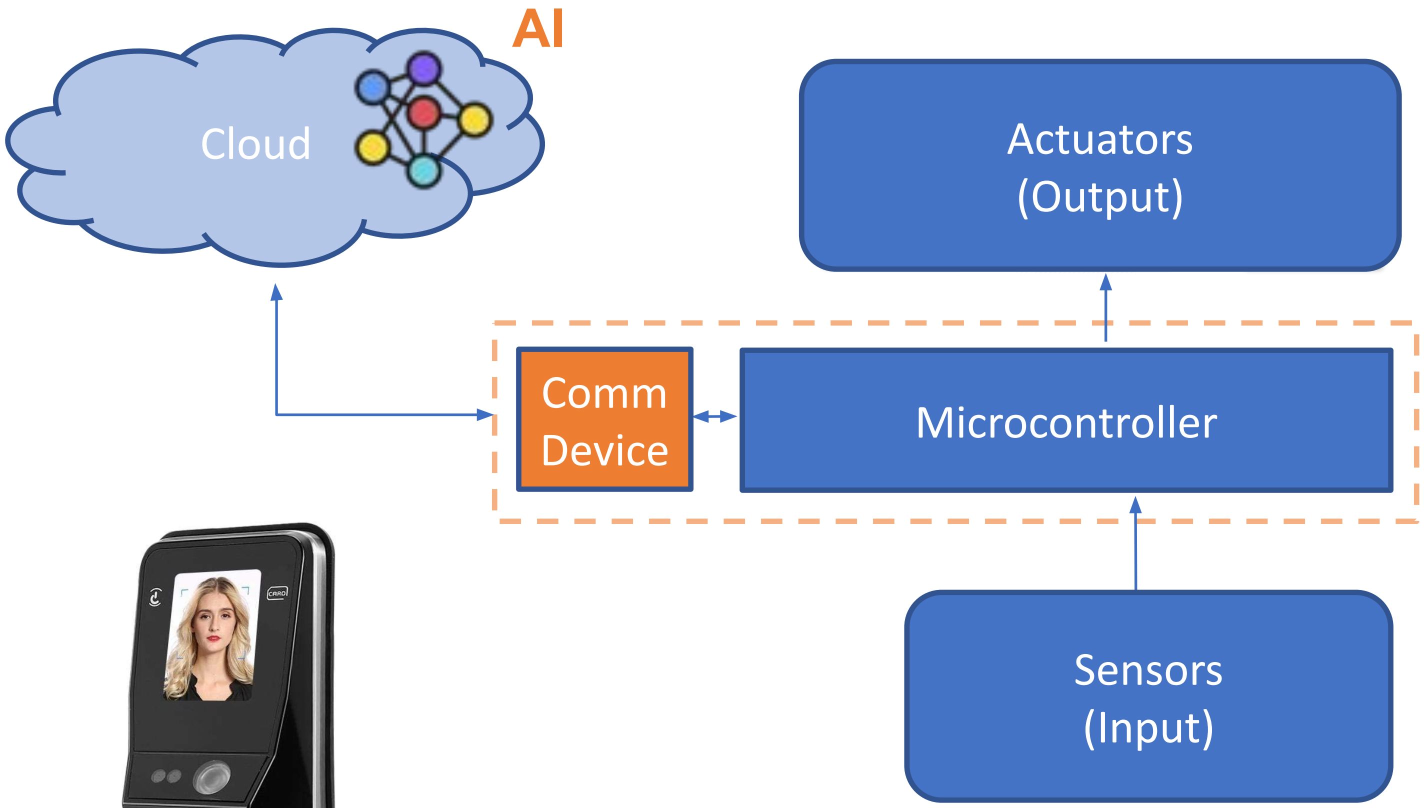


Bandwidth

Latency

Energy

Typical AIoT Project …



… Issues

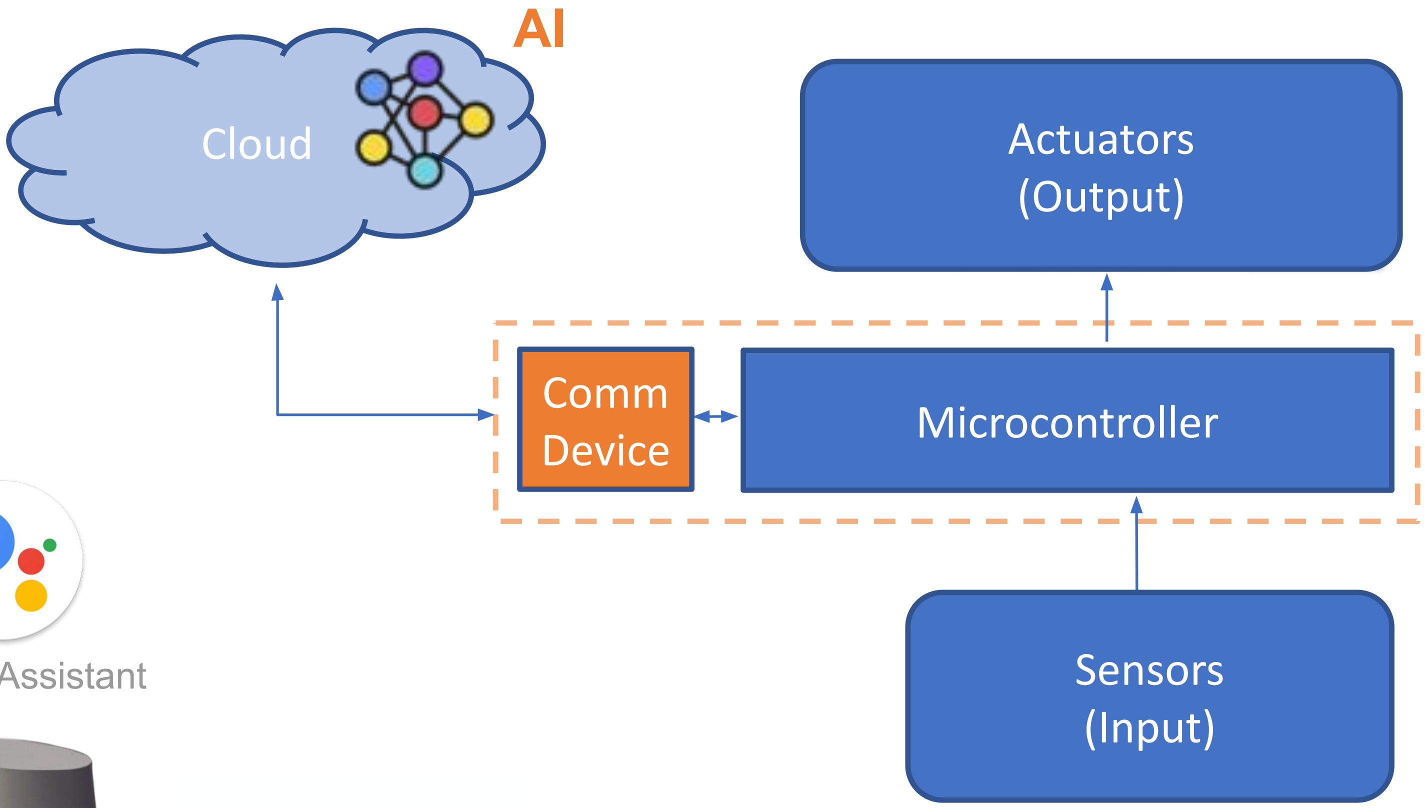
Bandwidth

Latency

Energy

Reliability

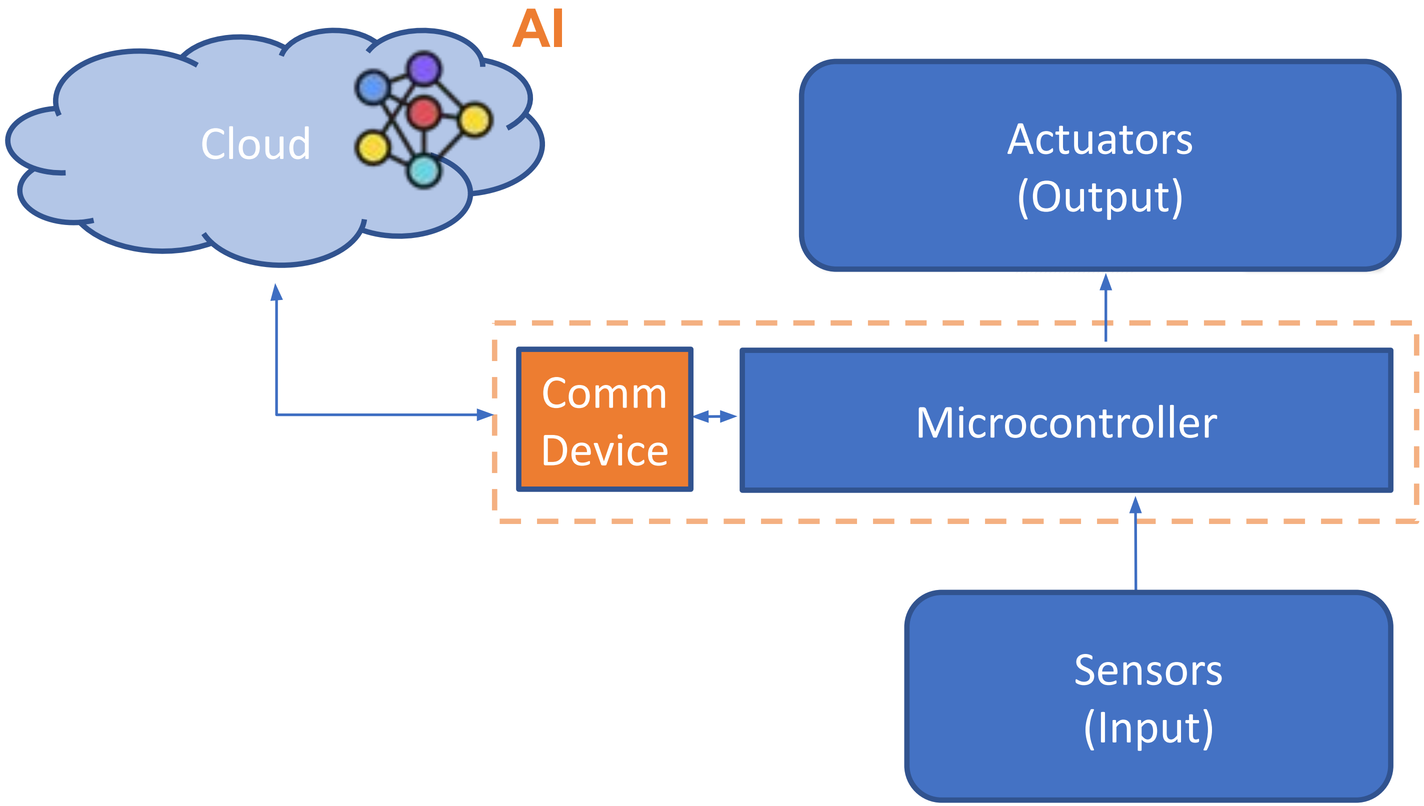
Typical AIoT Project … … Issues



Google Assistant

- Bandwidth
- Latency
- Energy
- Reliability
- Privacy

Typical AIoT Project …



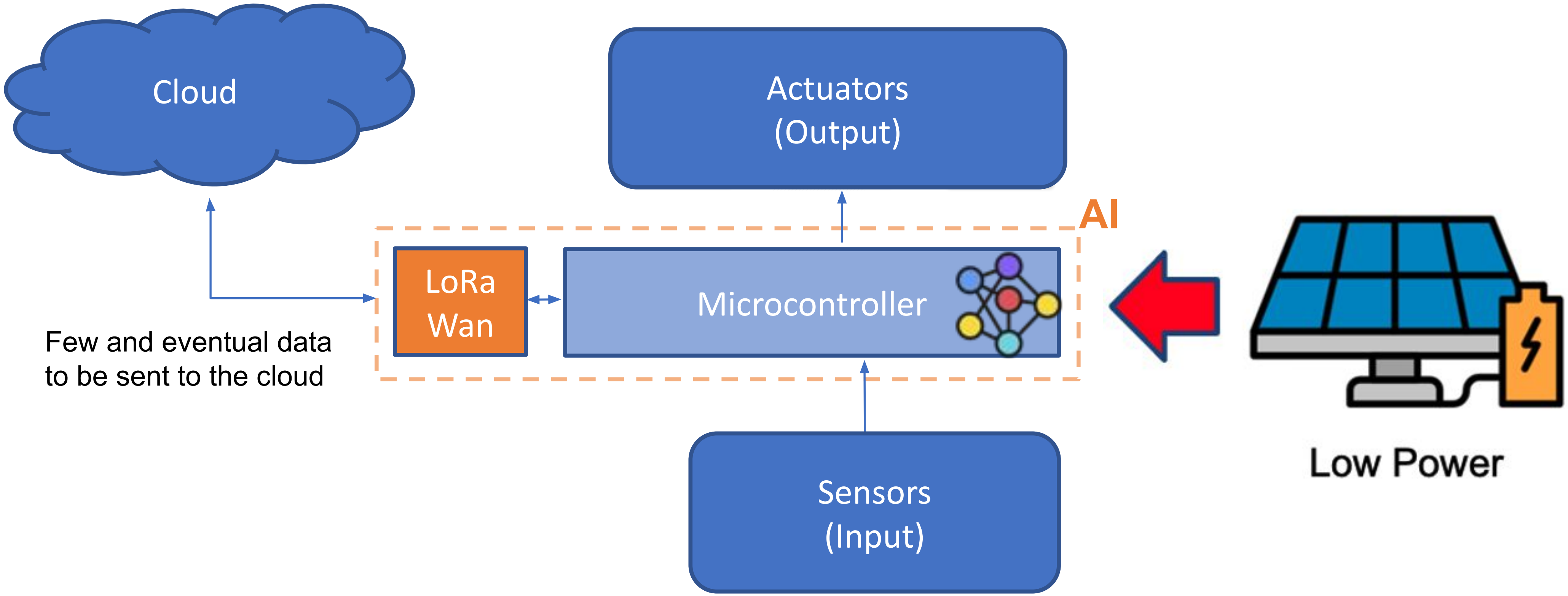
… Issues

Bandwidth
Latency
Energy
Reliability
Privacy

… Solution ?

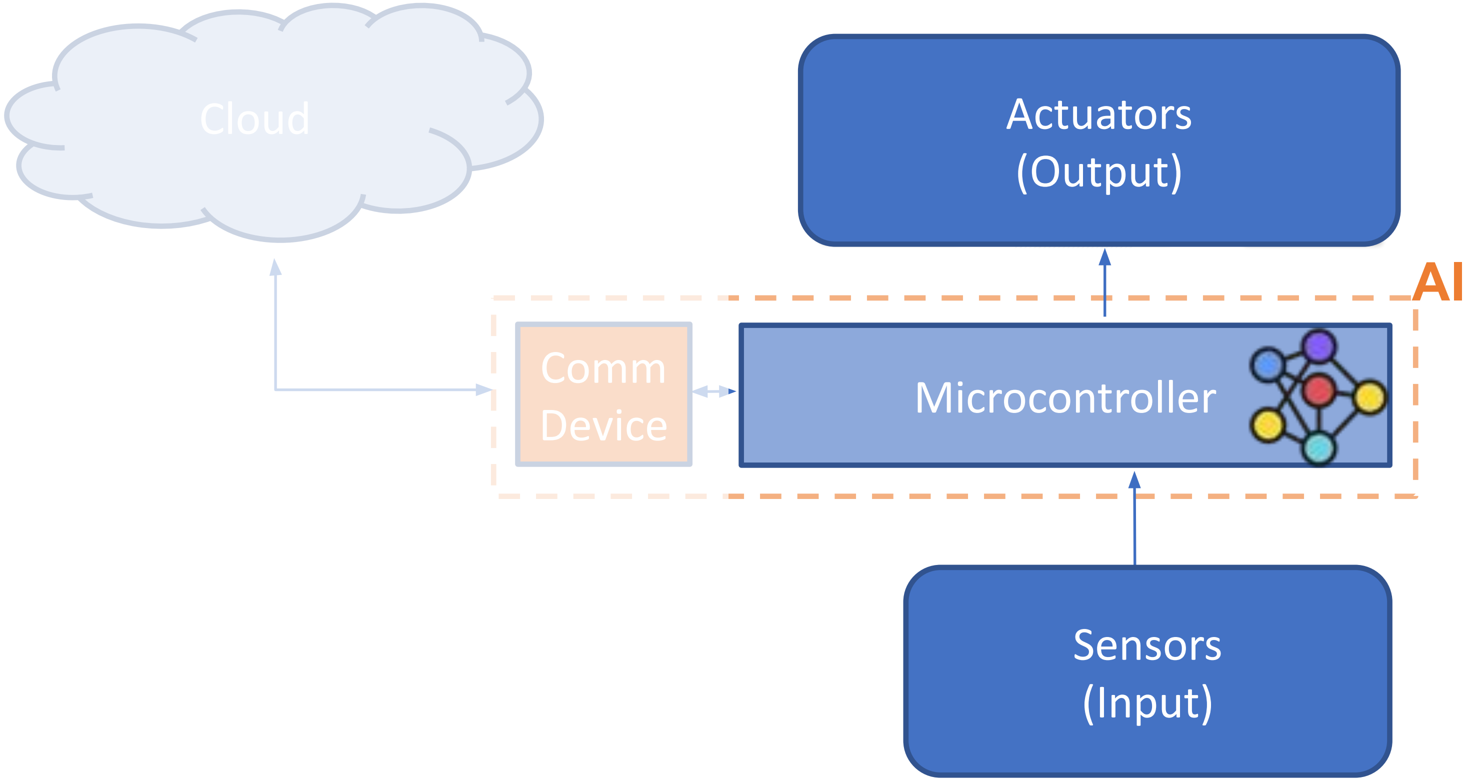
IoT 2.0 * – Edge AI/ML

* Intelligence of Things



… Solution -> ML goes close to data

When to use an Edge AI/ML approach:



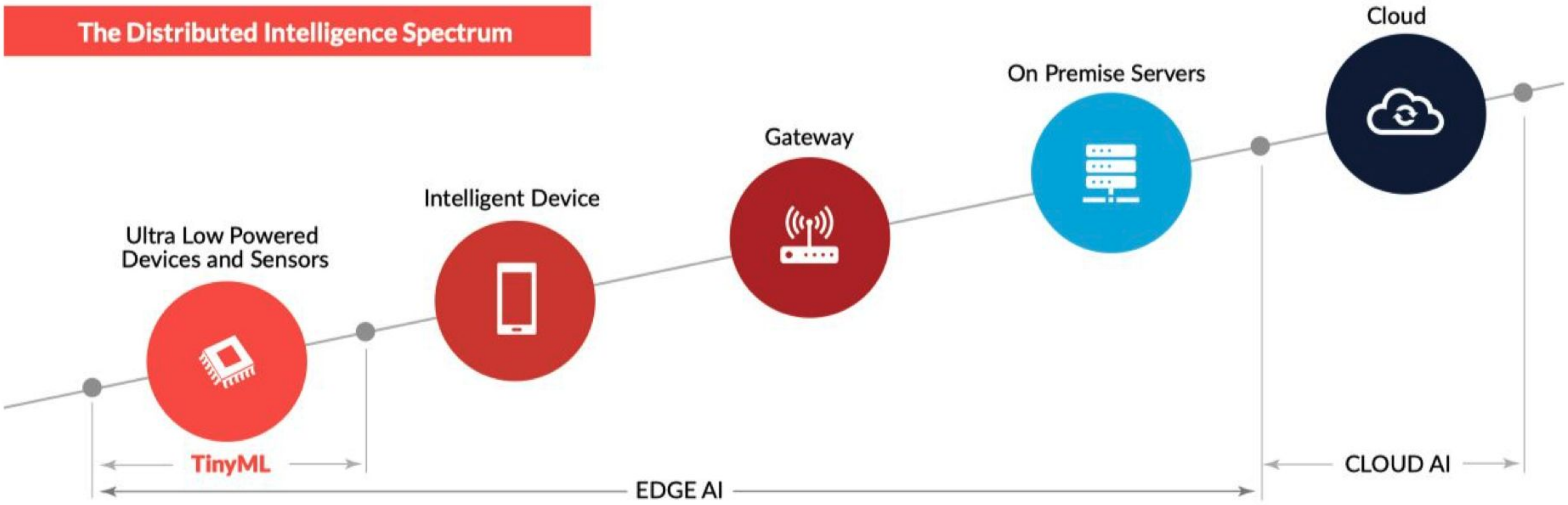
Bandwidth

Latency

Energy

Reliability

Privacy



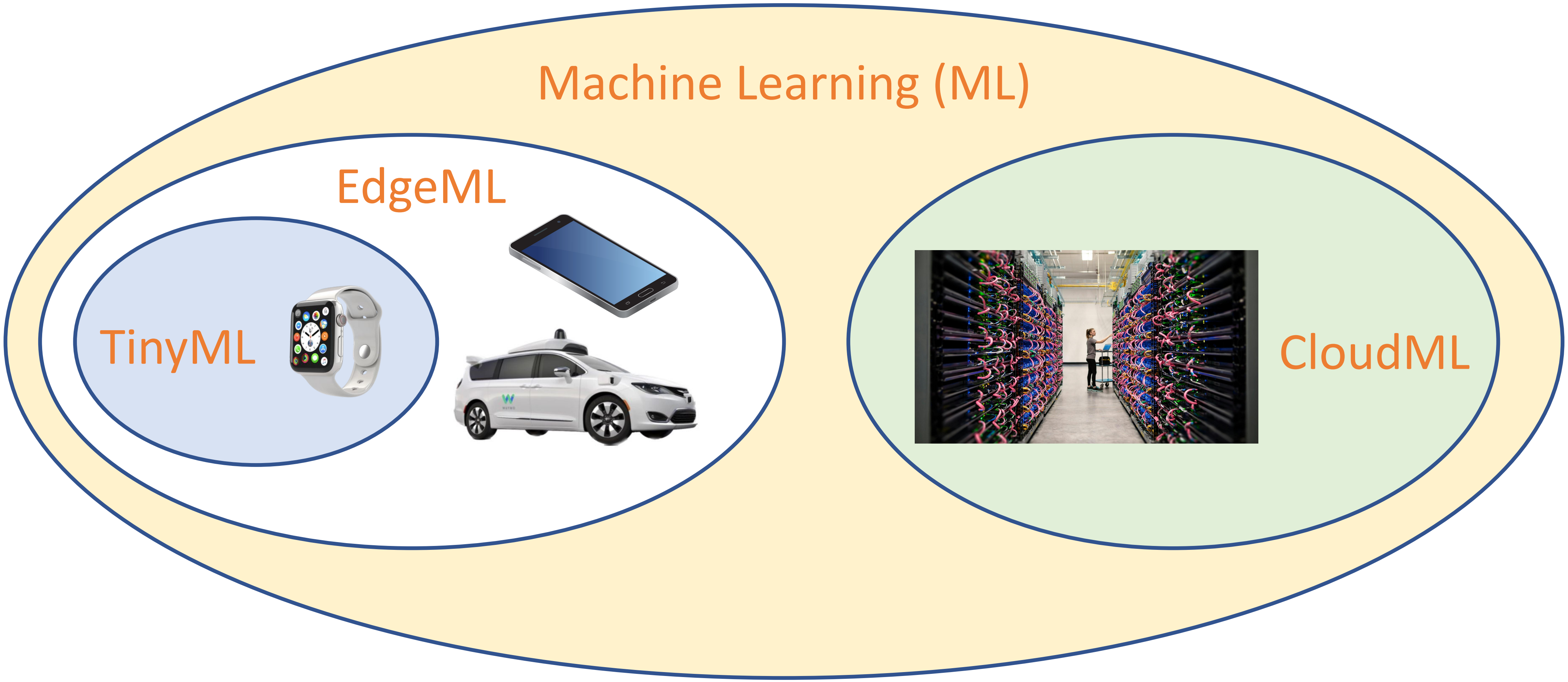
Source: ABI Research: TinyML

Machine Learning

CloudML: is essential for tasks requiring massive computational power or large-scale data analysis.

EdgeML (or EdgeAI) is the processing of Artificial Intelligence algorithms on edge, that is, on users' devices.

TinyML is where sensors are generating data with ultra-low power consumption (batteries) ("always on devices")


Machine Learning (ML)
EdgeML
TinyML
CloudML

Embedded ML (TinyML)

Introduction

What is AI?

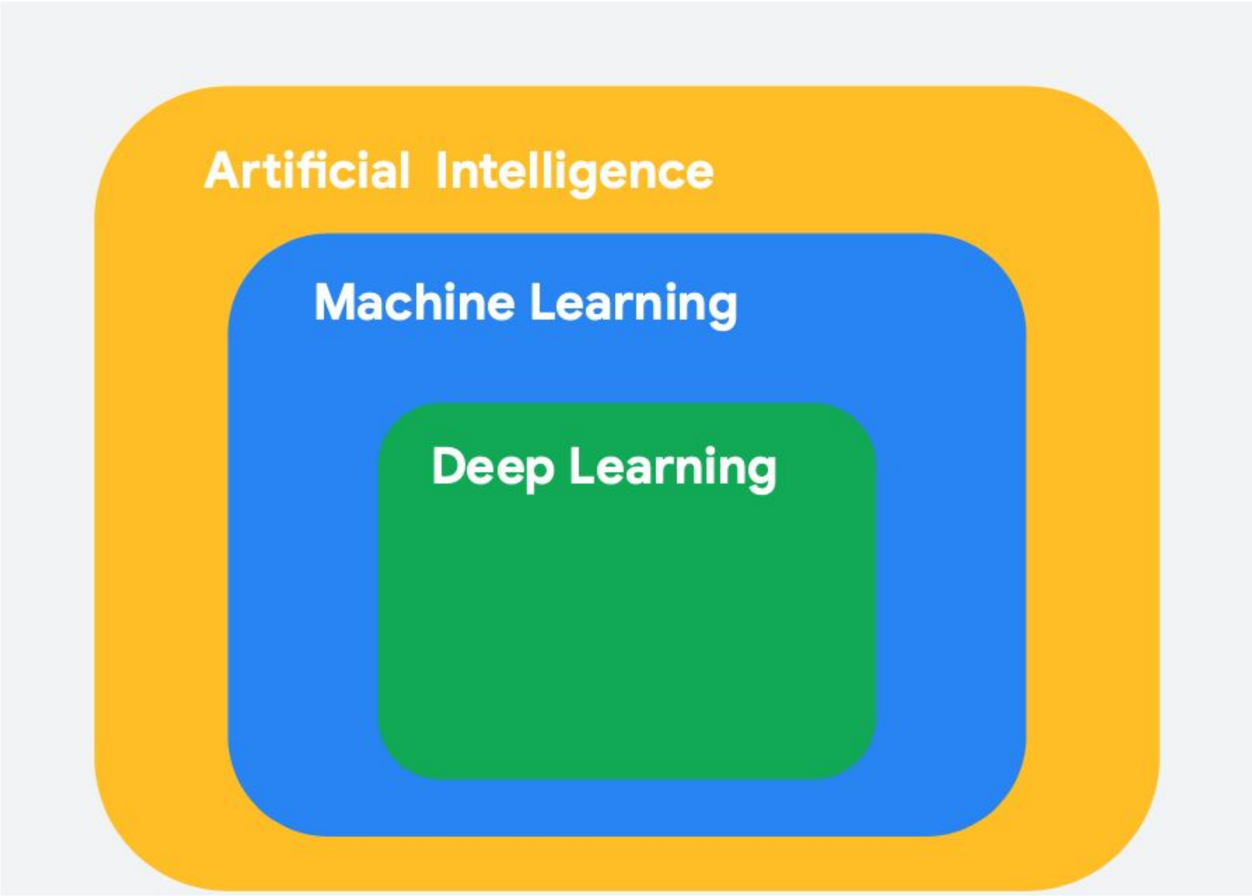
Artificial Intelligence (AI) is the simulation of human intelligence in machines that can perform tasks like learning, reasoning, and problem-solving.

Recommendation

Personal Assistants

LLMs
(Chat Bots)

Teaching computers how to learn a task directly from raw data



AI: Any technique that enables computers to mimic human behavior

ML: Ability to learn without explicitly being programed

DL: Extract patterns from data using neural networks

What is Tiny Machine Learning (**TinyML**)?

TinyML

Fastest-growing field of **ML**

23

What is Tiny Machine Learning (TinyML)?

TinyML

Fastest-growing field of **ML**

Algorithms, hardware, software

24

What is Tiny Machine Learning (**TinyML**)?

TinyML

Fastest-growing field of **ML**

Algorithms, hardware, software

On-device sensor analytics

25

What is Tiny Machine Learning (TinyML)?

TinyML

Fastest-growing field of ML

Algorithms, hardware, software

On-device sensor analytics

Low power consumption

26

What is Tiny Machine Learning (TinyML)?

TinyML

Fastest-growing field of ML

Algorithms, hardware, software

On-device sensor analytics

Low power consumption

Battery-operated

27

What is Tiny Machine Learning (TinyML)?

TinyML

Fastest-growing field of **ML**

Algorithms, hardware, software

On-device sensor analytics

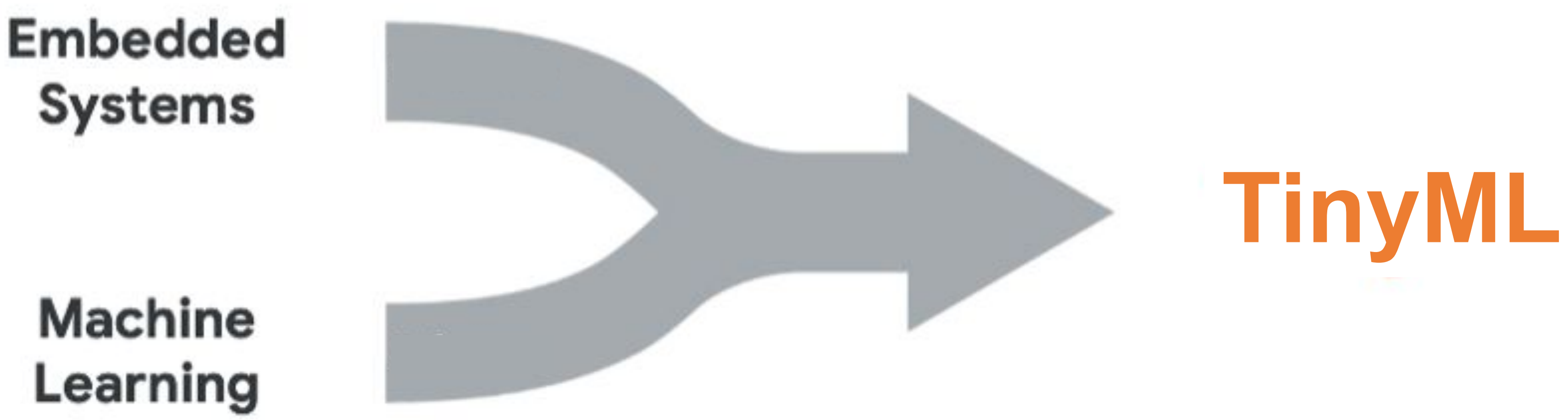
Low power consumption

Battery-operated

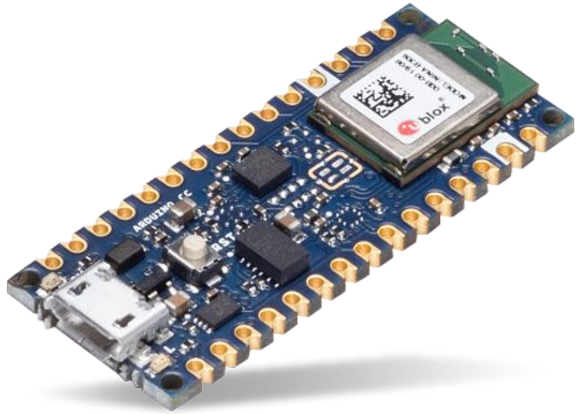
Always-on ML

28

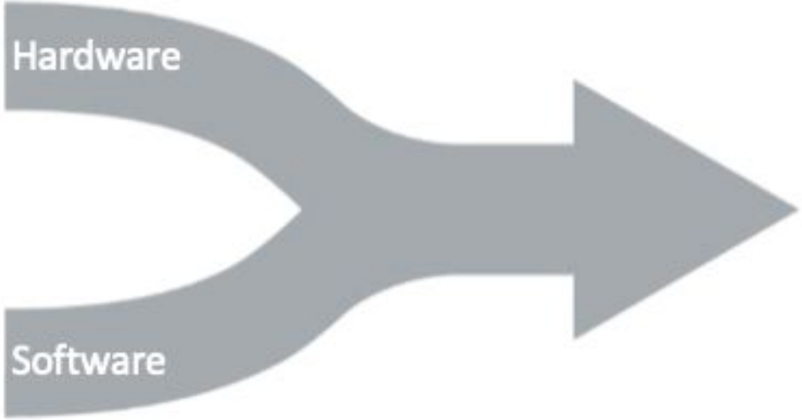
What Makes TinyML ?



What Makes TinyML ?

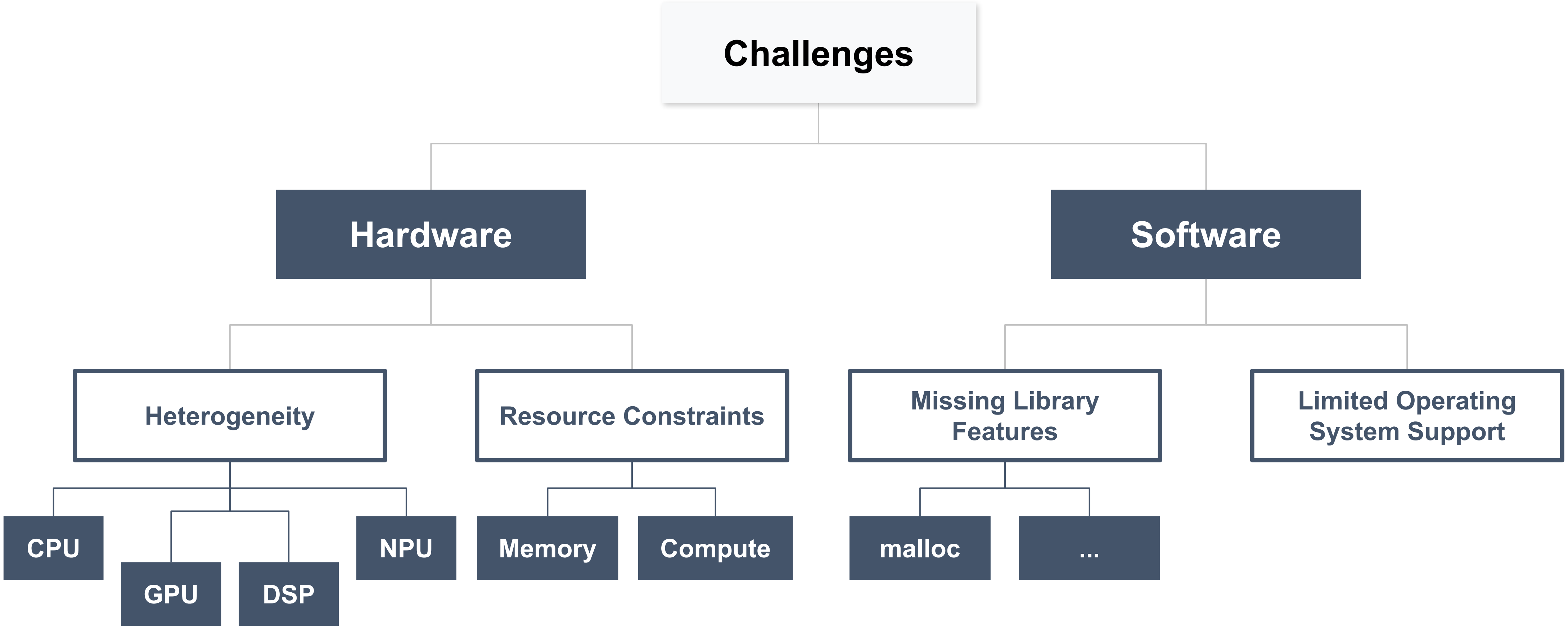


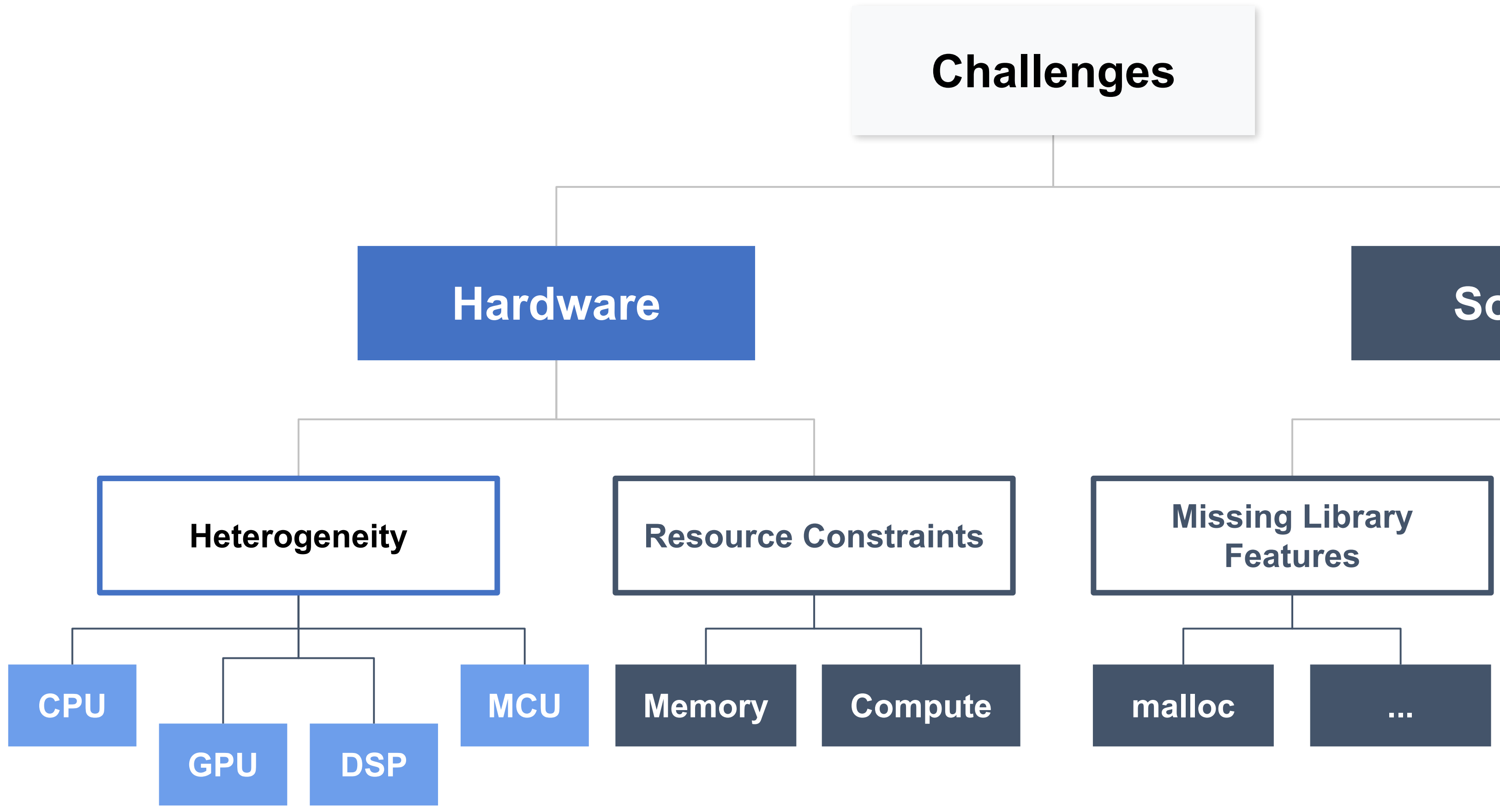
TensorFlow Lite



TinyML

TinyML Challenges


Challenges
Hardware
Software
Heterogeneity
Resource Constraints
Missing Library Features
Limited Operating System Support
CPU
GPU
DSP
NPU
Memory
Compute
malloc
...

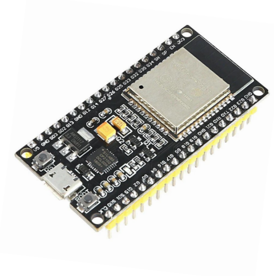

Challenges
Hardware
Software
Heterogeneity
Resource Constraints
Missing Library Features
Limited Operating System Support
CPU
GPU
DSP
MCU
Memory
Compute
malloc
...

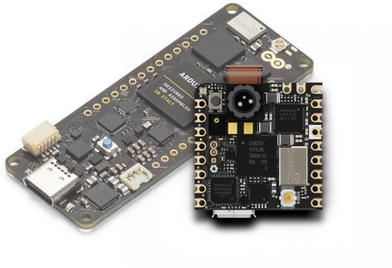
250 Billion

MCUs today

34

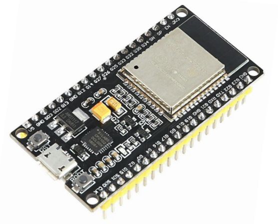
Hardware

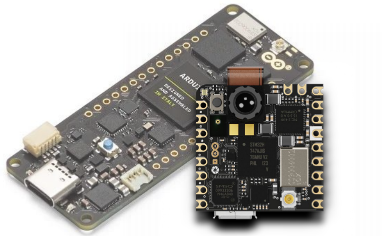




Hardware (Development Boards)

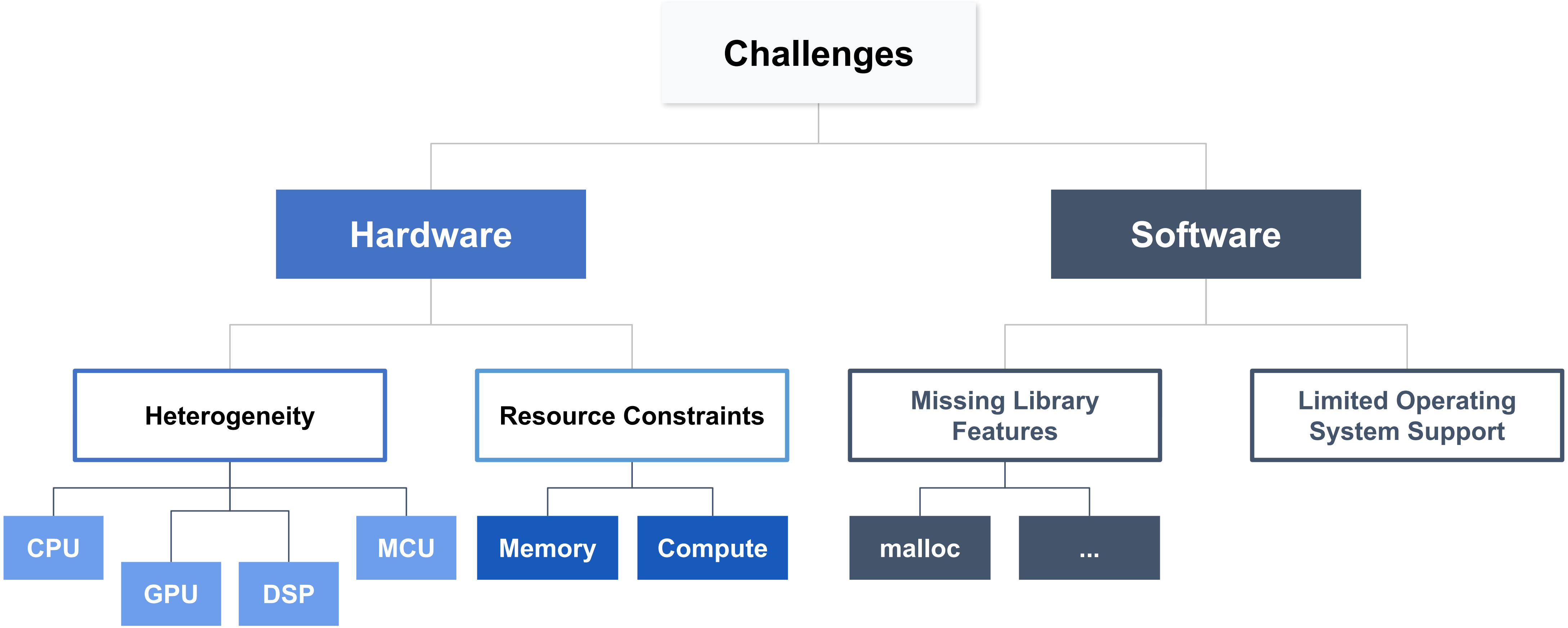






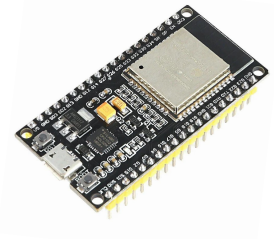
	Raspberry Pico (W)	Arduino Nano Sense	ESP 32	Seeed XIAO Sense / ESP32S3	Arduino Pro
32Bits CPU	Dual-core Arm Cortex-M0+	Arm Cortex-M4F	Xtensa LX6 Dual Core	Arm Cortex-M4F (BLE) Xtensa LX7 Dual Core	Dual Core Arm Cortex M7/M4
CLOCK	133MHz	64MHz	240MHz	64 / 240MHz	480/240MHz
RAM	264KB	256KB	520KB (part available)	256KB / 8MB	1MB
ROM	2MB	1MB	2MB	2MB / 8MB	2MB
Radio	(Yes for W)	BLE	BLE/WiFi	BLE / WiFi (ESP32S3)	BLE/WiFi
Sensors	No	Yes	No	Yes (Sense)	Yes (Nicla)
Bat. Power Manag.	No	No	No	Yes	Yes
Price	$	$$$	$	$$	$$$$$

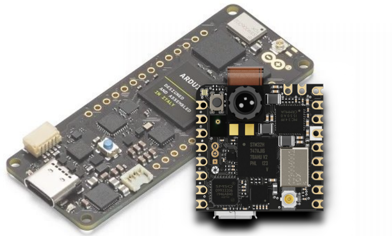
https://media.digikey.com/Resources/Maker/the-original-guide-to-boards-2022.pdf


Challenges
Hardware
Software
Heterogeneity
Resource Constraints
Missing Library Features
Limited Operating System Support
CPU
GPU
DSP
MCU
Memory
Compute
malloc
...

Hardware



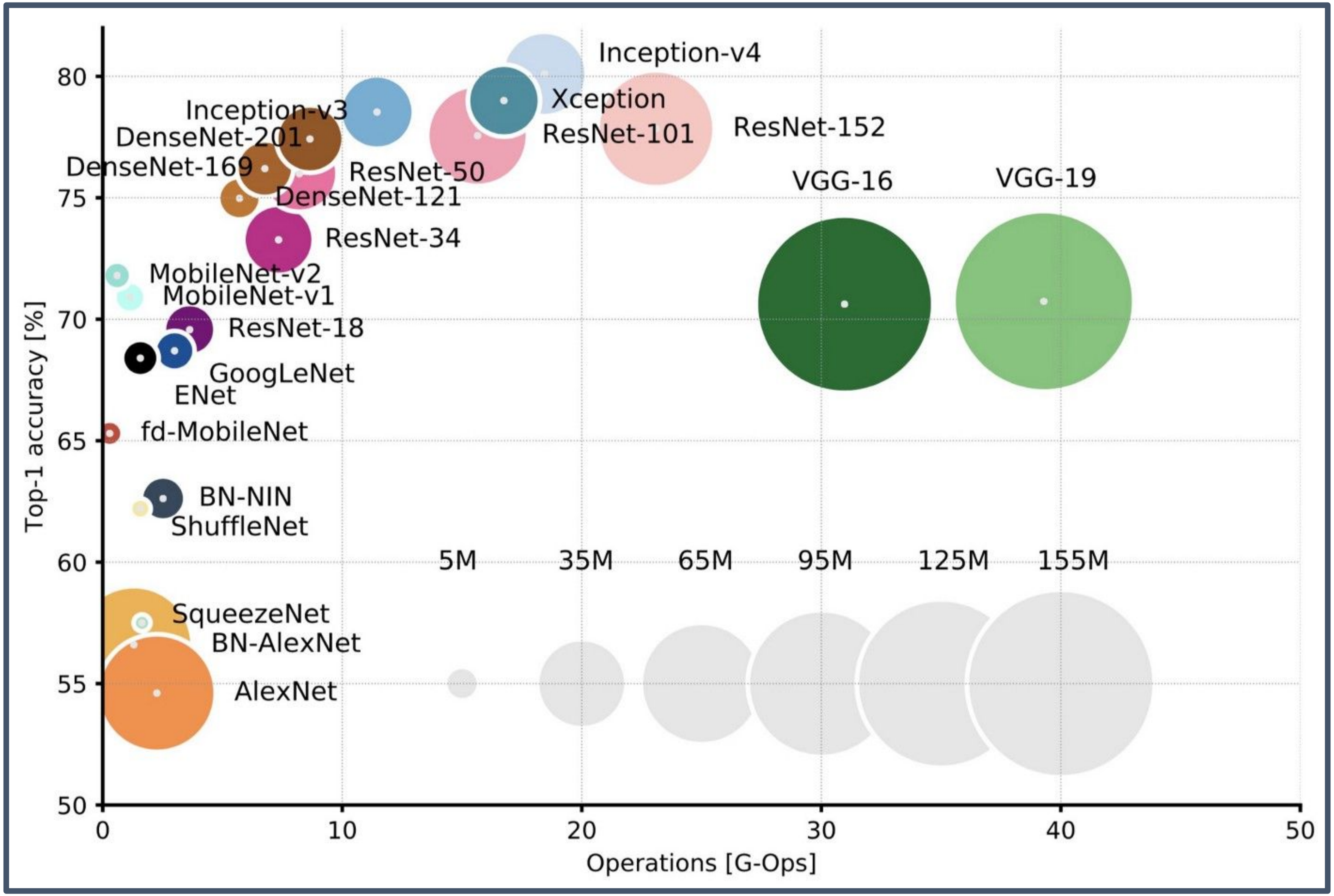


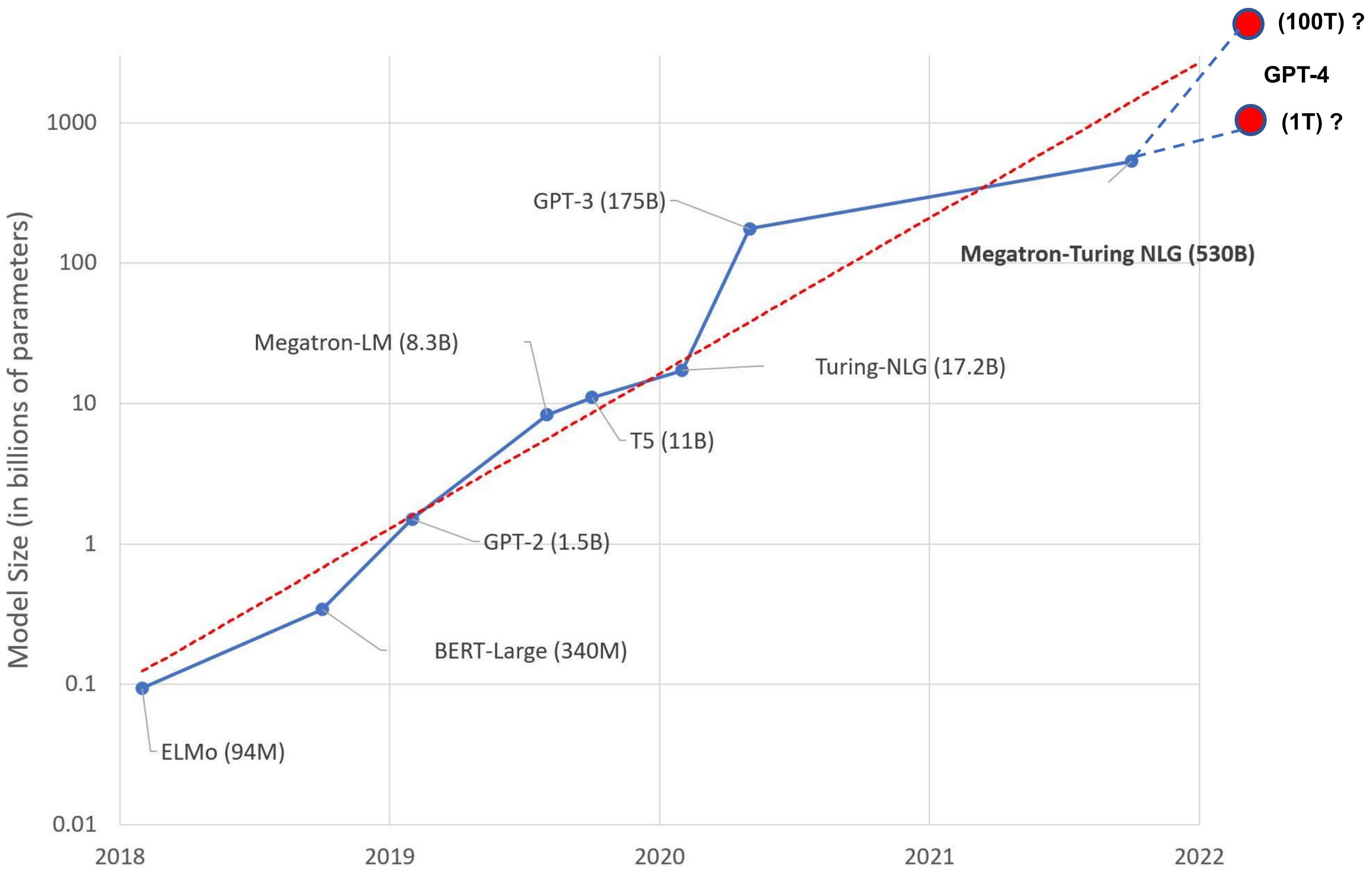


	Raspberry Pico (W)	Arduino Nano Sense	ESP 32	Seeed XIAO Sense / ESP32S3	Arduino Pro
32Bits CPU	Dual-core Arm Cortex-M0+	Arm Cortex-M4F	Xtensa LX6 Dual Core	Arm Cortex-M4F (BLE) Xtensa LX7 Dual Core	Dual Core Arm Cortex M7/M4
CLOCK	133MHz	64MHz	240MHz	64 / 240MHz	480/240MHz
RAM	264KB	256KB	520KB (part available)	256KB / 8MB	1MB
ROM	2MB	1MB	2MB	2MB / 8MB	2MB
Radio	(Yes for W)	BLE	BLE/WiFi	BLE / WiFi (ESP32S3)	BLE/WiFi
Sensors	No	Yes	No	Yes (Sense)	Yes (Nicla)
Bat. Power Manag.	No	No	No	Yes	Yes
Price	$	$$$	$	$$	$$$$$

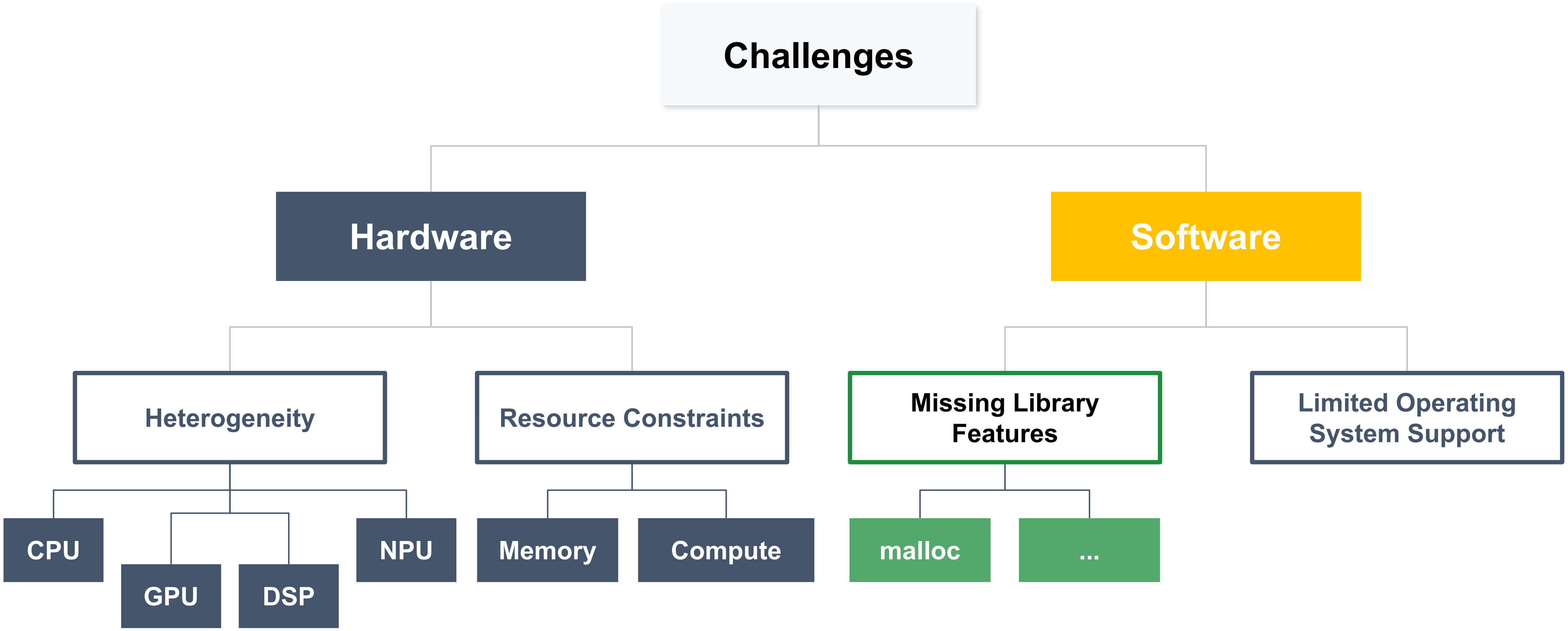
https://media.digikey.com/Resources/Maker/the-original-guide-to-boards-2022.pdf

https://arxiv.org/pdf/1910.01108.pdf
https://towardsdatascience.com/neural-network-architectures-156e5bad51ba





https://huggingface.co/blog/large-language-models https://arxiv.org/abs/2303.08774


Challenges
Hardware
Software
Heterogeneity
Resource Constraints
Missing Library Features
Limited Operating System Support
CPU
GPU
DSP
NPU
Memory
Compute
malloc
...

Datasets Preprocessing

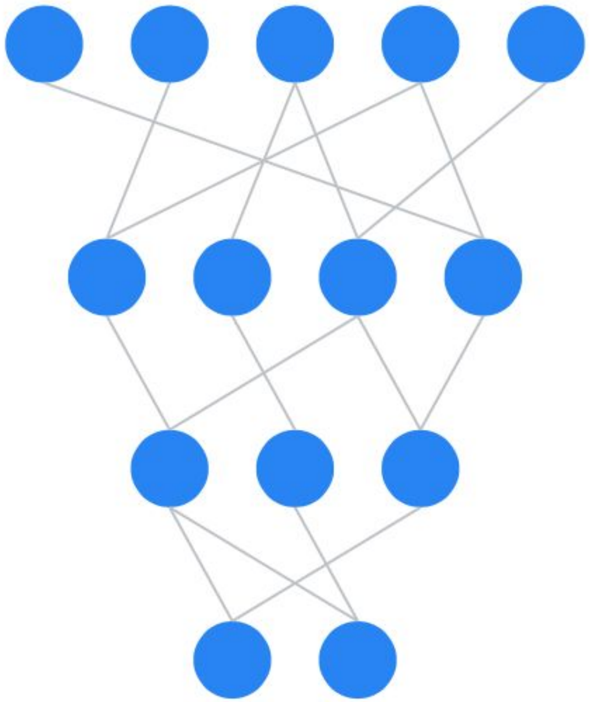
Quantization Pruning, Distillation

Resource constraints

Sound

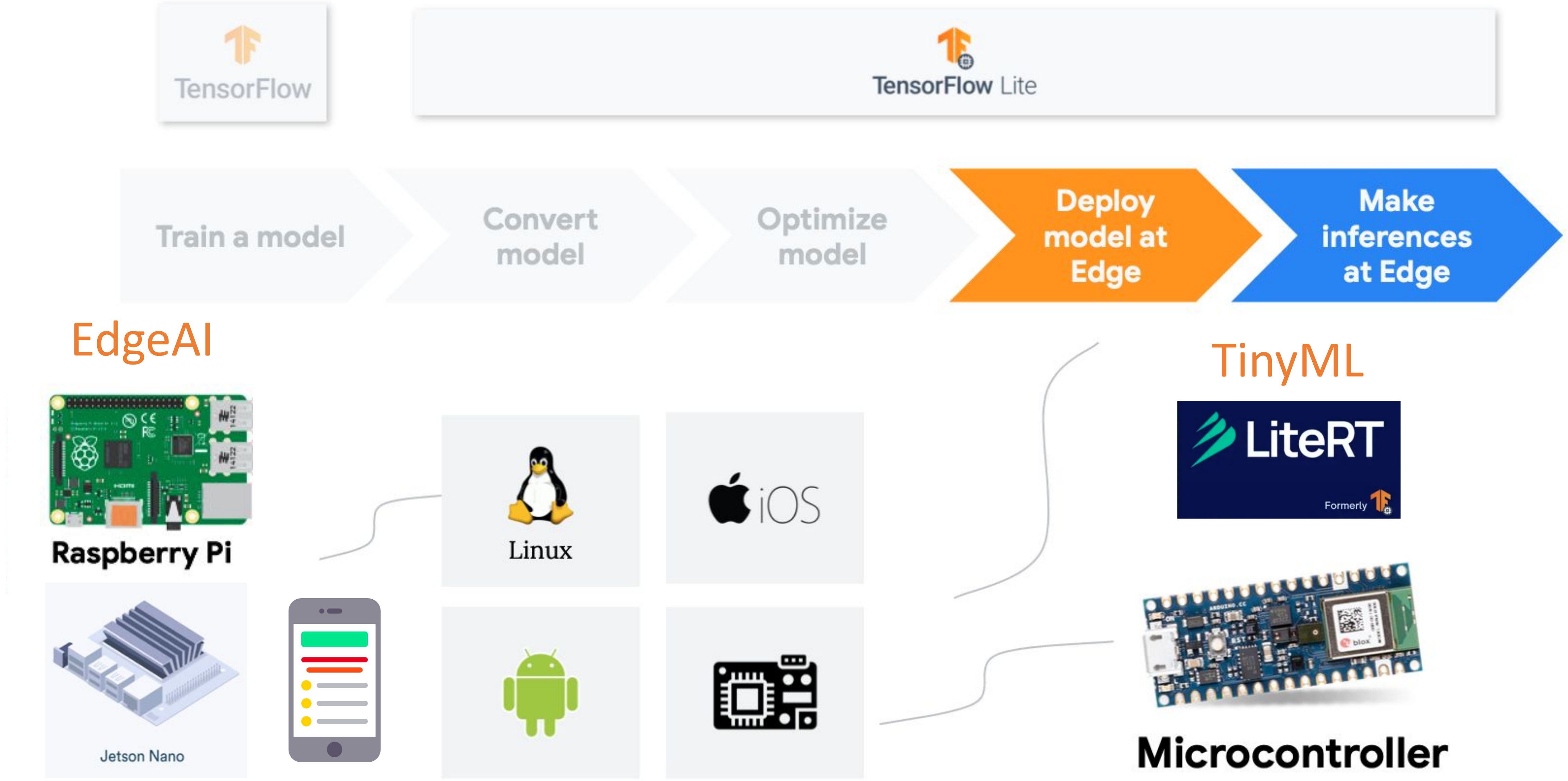
Vision

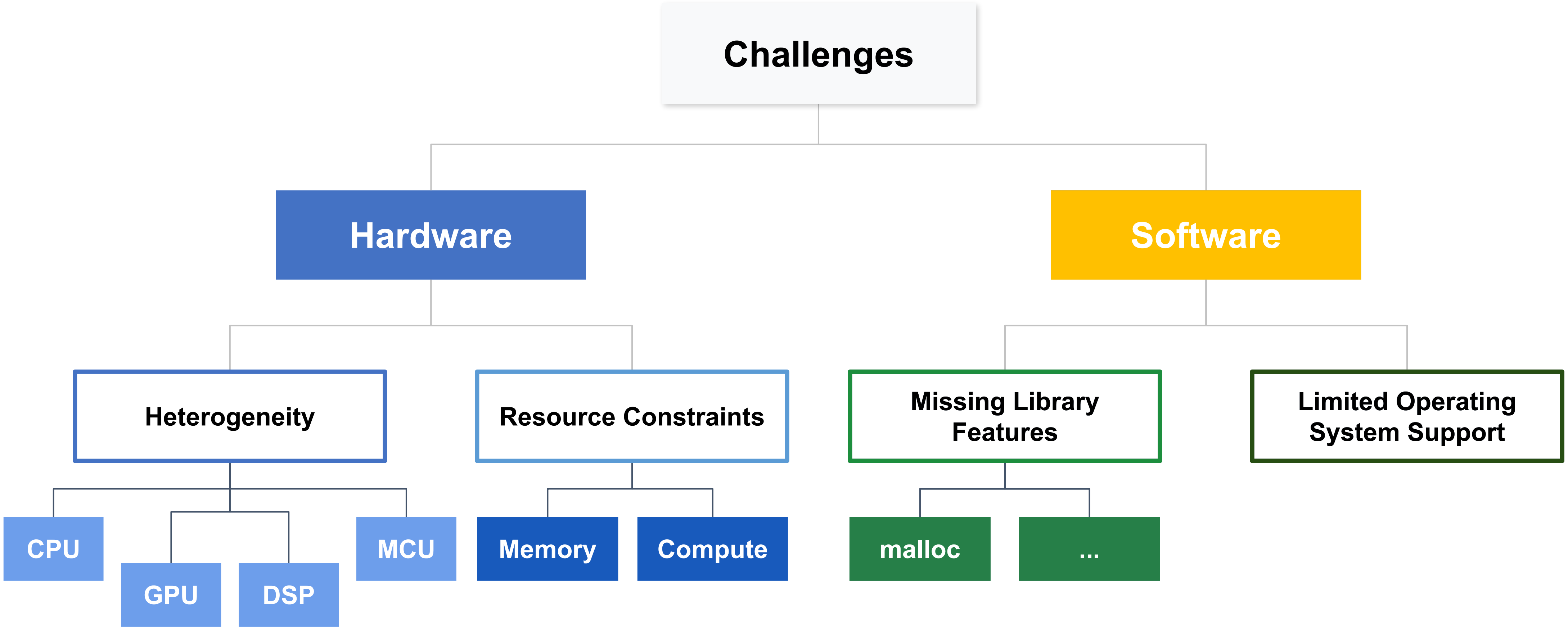
Vibration



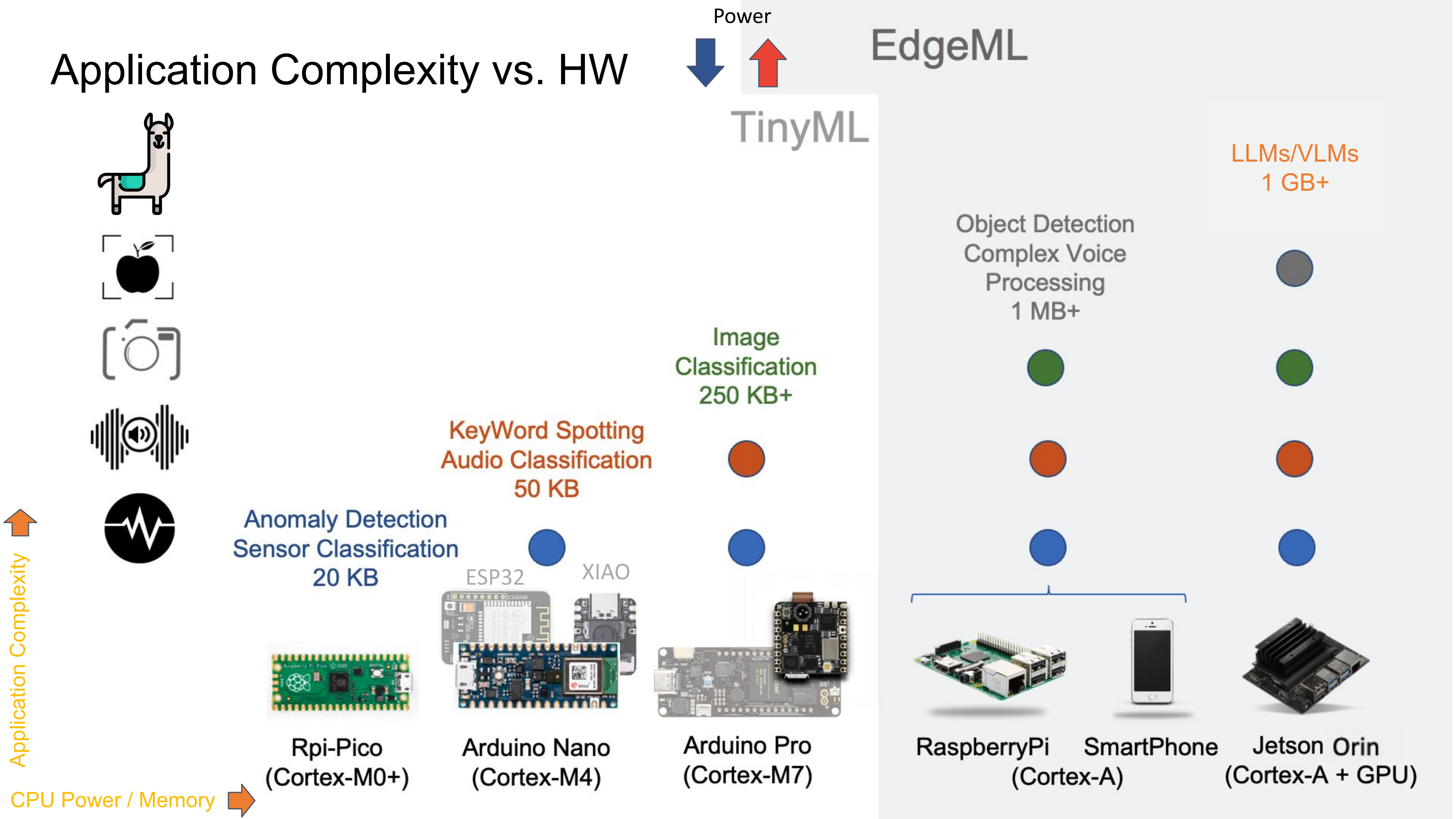
End-to-end **TinyML** application design

TinyML/EdgeAI Software

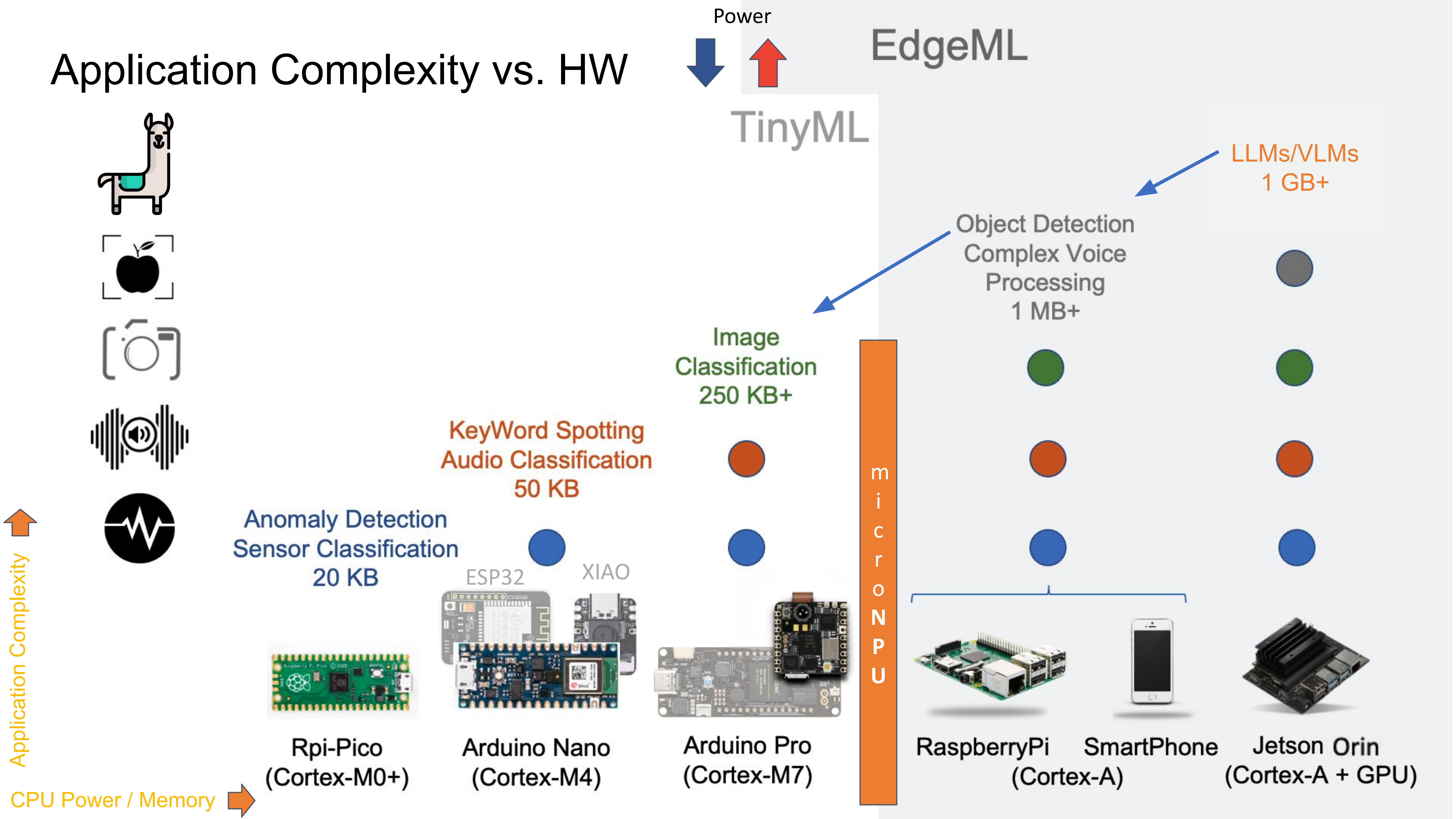



Challenges
Hardware
Software
Heterogeneity
Resource Constraints
Missing Library Features
Limited Operating System Support
CPU
GPU
DSP
MCU
Memory
Compute
malloc
...

Application Complexity vs. HW

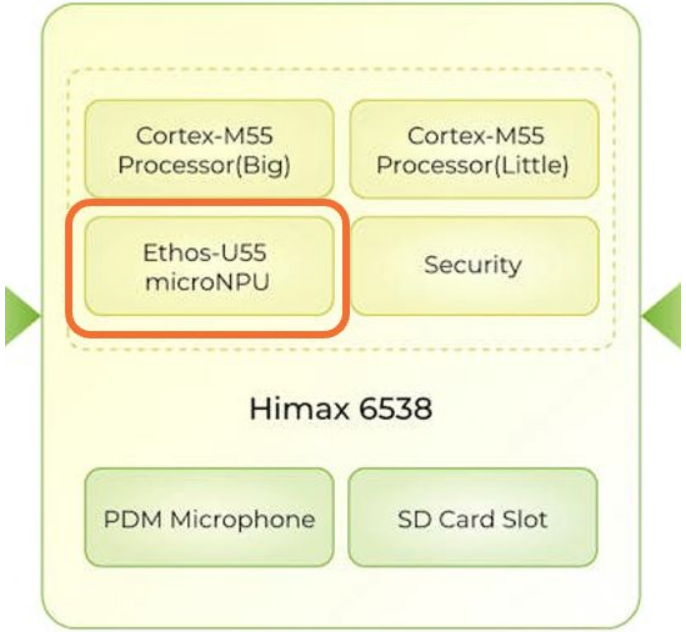


Application Complexity vs. HW



microNPU

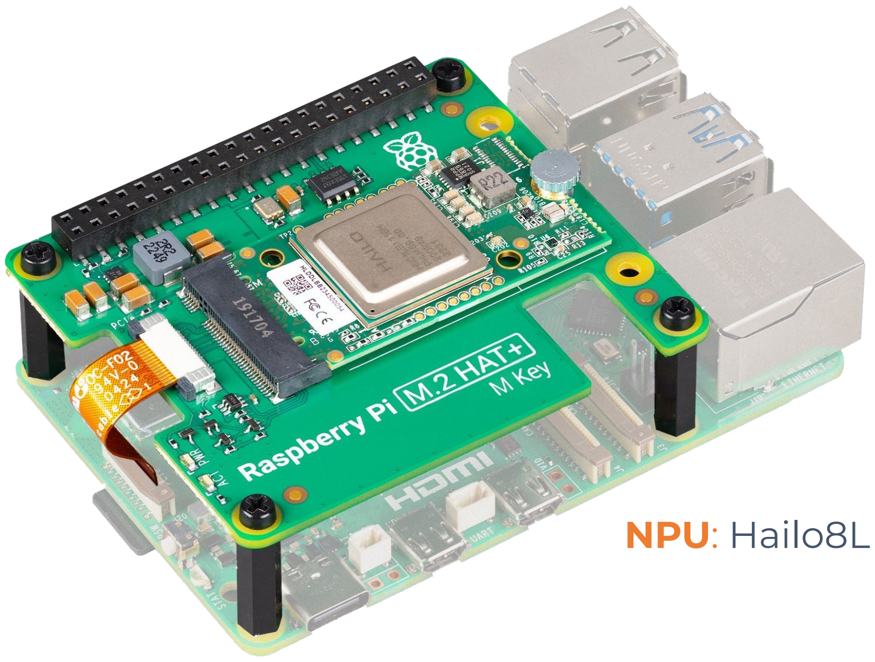
Grove Vision AI v2



0.5 TOPS

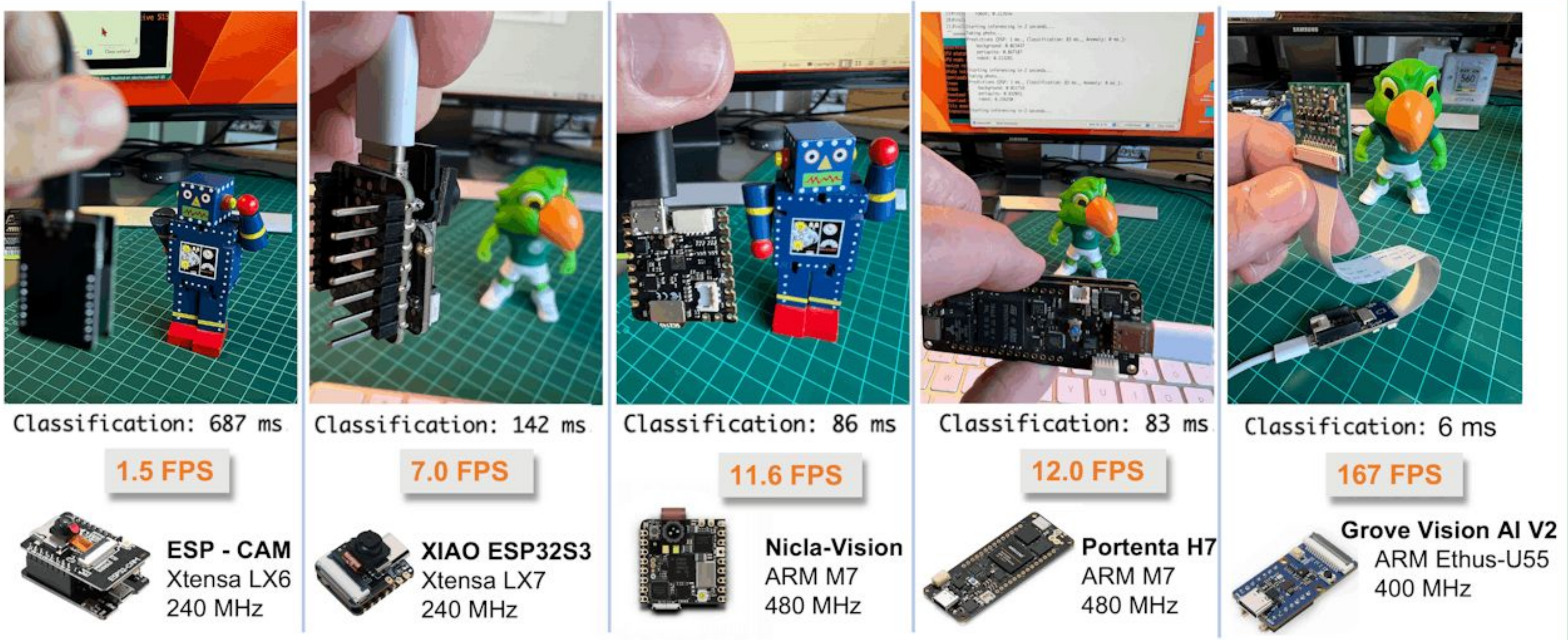
NN Inference Accelerator

Raspberry Pi AI Kit

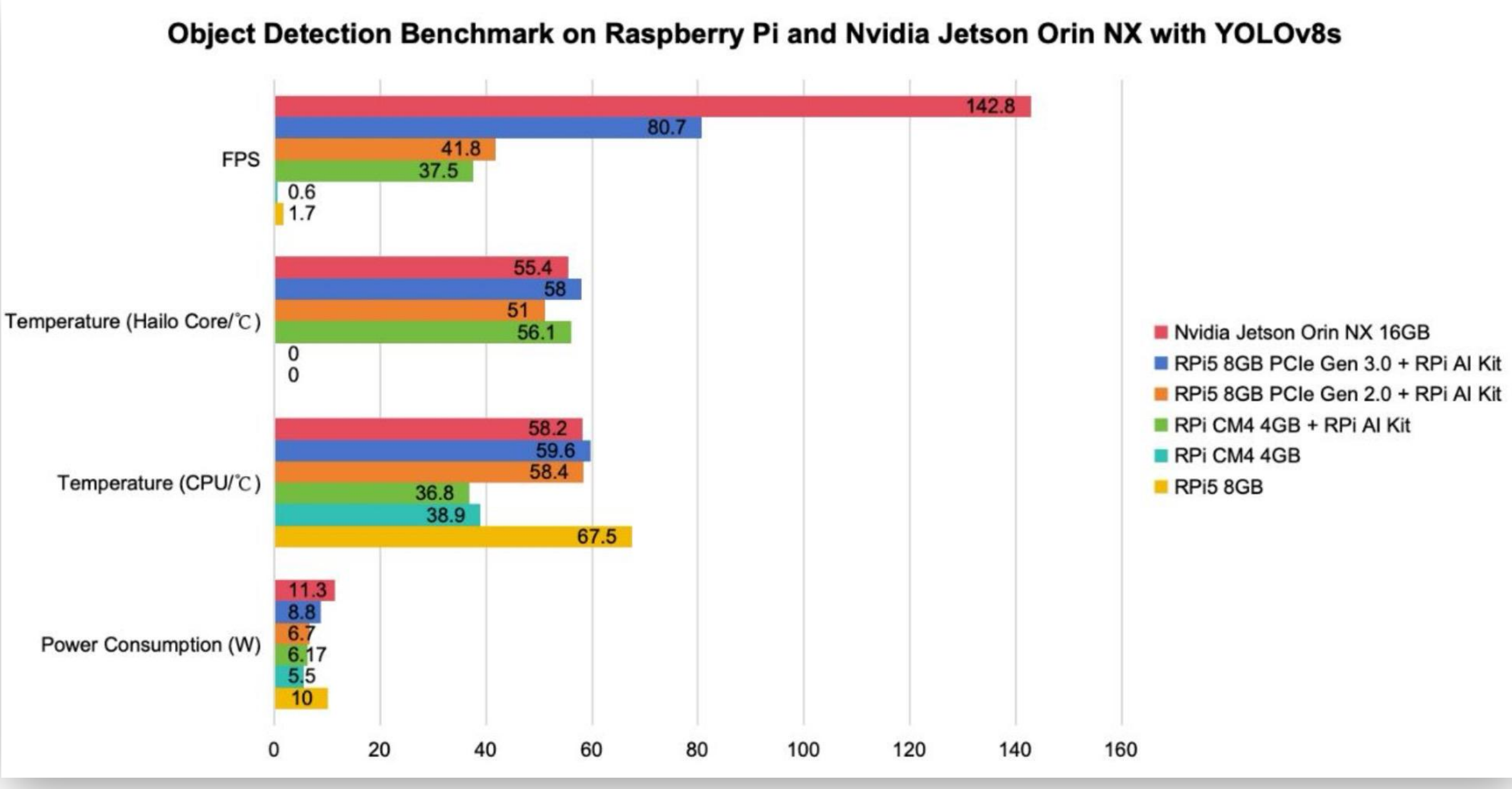


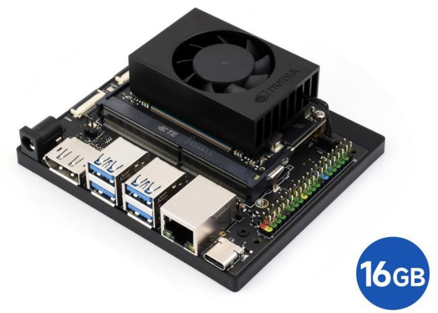
NPU: Hailo8L

13 - 25 TOPS

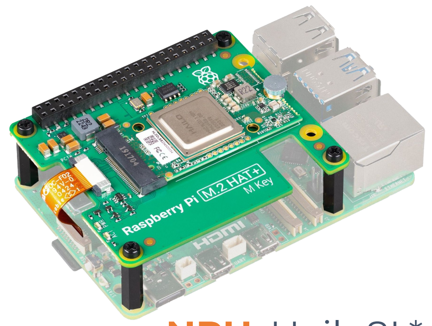


450mW 590mW 350mW





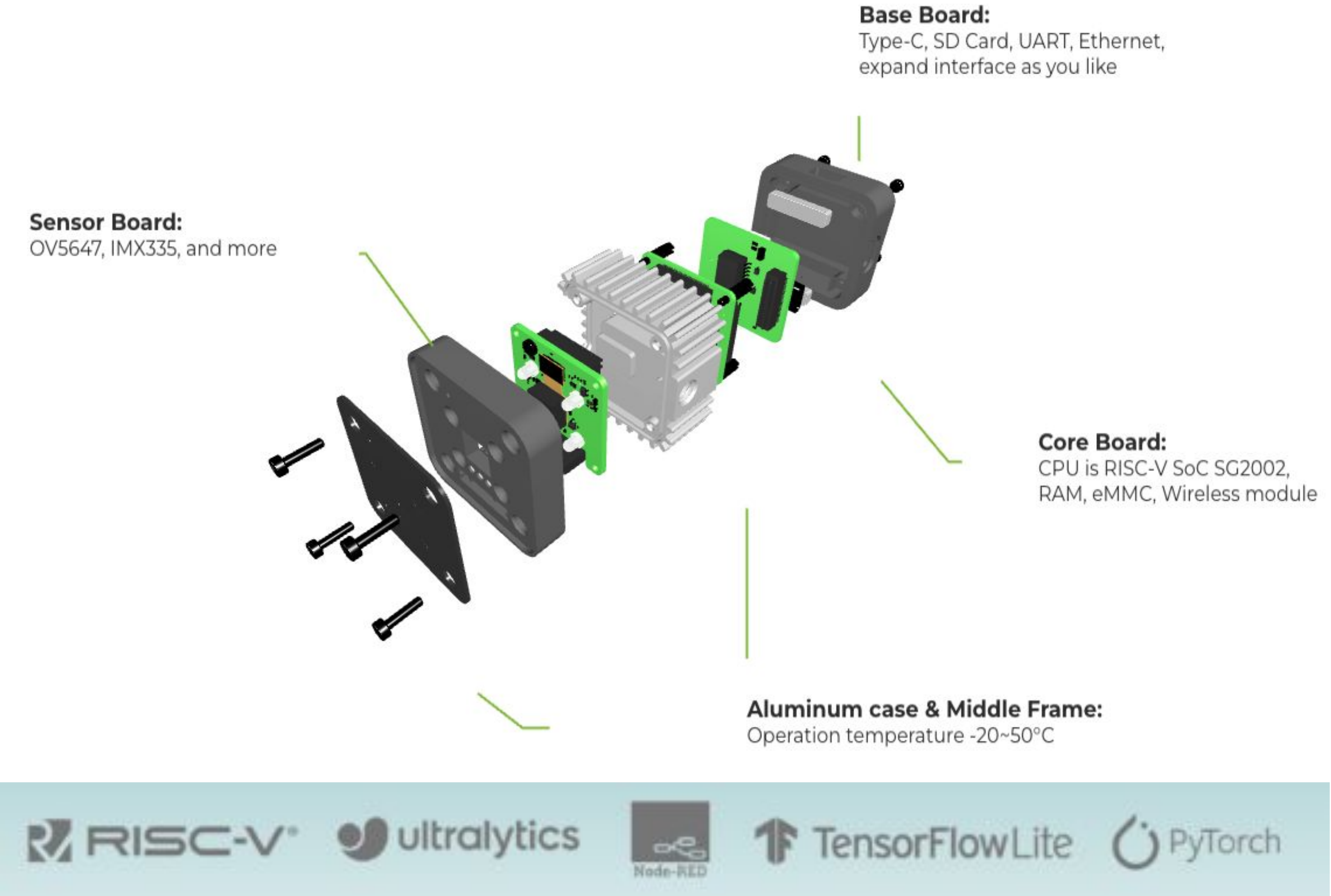
GPU: NVIDIA Ampere
100 TOPS



NPU: Hailo8L*

* **Note:** Hailo-10H for LLMs is planned for future release

seeed studio reCamera



Real-World Applications

Examples

Real-World Applications

Agriculture

Healthcare

Industry

Environment

Real-World Applications

- Agriculture
- Healthcare
- Industry
- Environment

Classifying mosquito wingbeat sound using TinyML

Moez Altayeb
University of Khartoum, Sudan
ICTP, Trieste, Italy
mohedahmed@hotmail.com

Marcelo Rovai
Universidade Federal de Itajubá
Itajubá, Brazil
rovai@unifei.edu.br

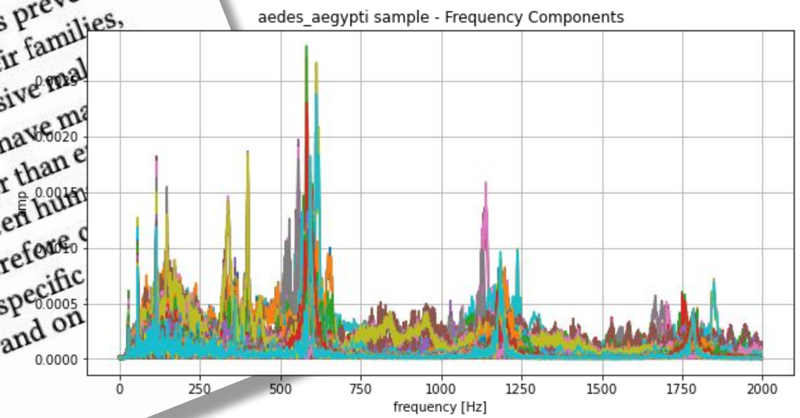
Marco Zennaro
ICTP
Trieste, Italy
mzennaro@ictp.it

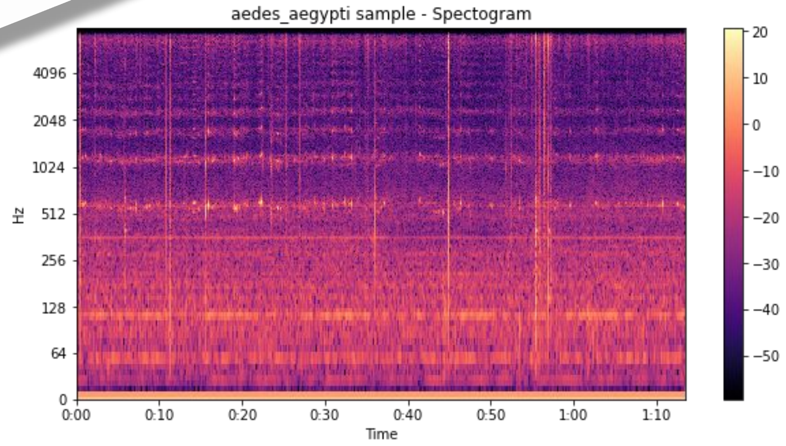
ABSTRACT

Every year more than one billion people are infected and more than one million people die from vector-borne diseases including malaria, dengue, zika and chikungunya. Mosquitoes are the best known disease vector and are geographically spread worldwide. It is important to raise awareness of mosquito proliferation by monitoring their incidence, especially in poor regions. Acoustic detection of mosquitoes has been studied for long and ML can be used to automatically identify mosquito species by their wingbeat. We present a prototype solution based on an openly available dataset, on the Edge Impulse platform and on three commercially-available TinyML devices. The proposed solution is low-power, low-cost and can run without human intervention in resource-constrained areas. This insect monitoring system can reach a global scale.

affected. People from poor communities with little access to health care and clean water sources are also at risk. Although anti-malarial drugs exist, there's currently no malaria vaccine.

Vector-borne diseases also exacerbate poverty. Illness prevent people from working and supporting themselves and their families, impeding economic development. Countries with intensive mal have much lower income levels than those that don't have ma Countries affected by malaria turn to control rather than e tion. Vector control means decreasing contact between hum disease carriers on an area-by-area basis. It is therefore c be able to detect the presence of mosquitoes in a specific paper presents an approach based on TinyML and on embedded devices.



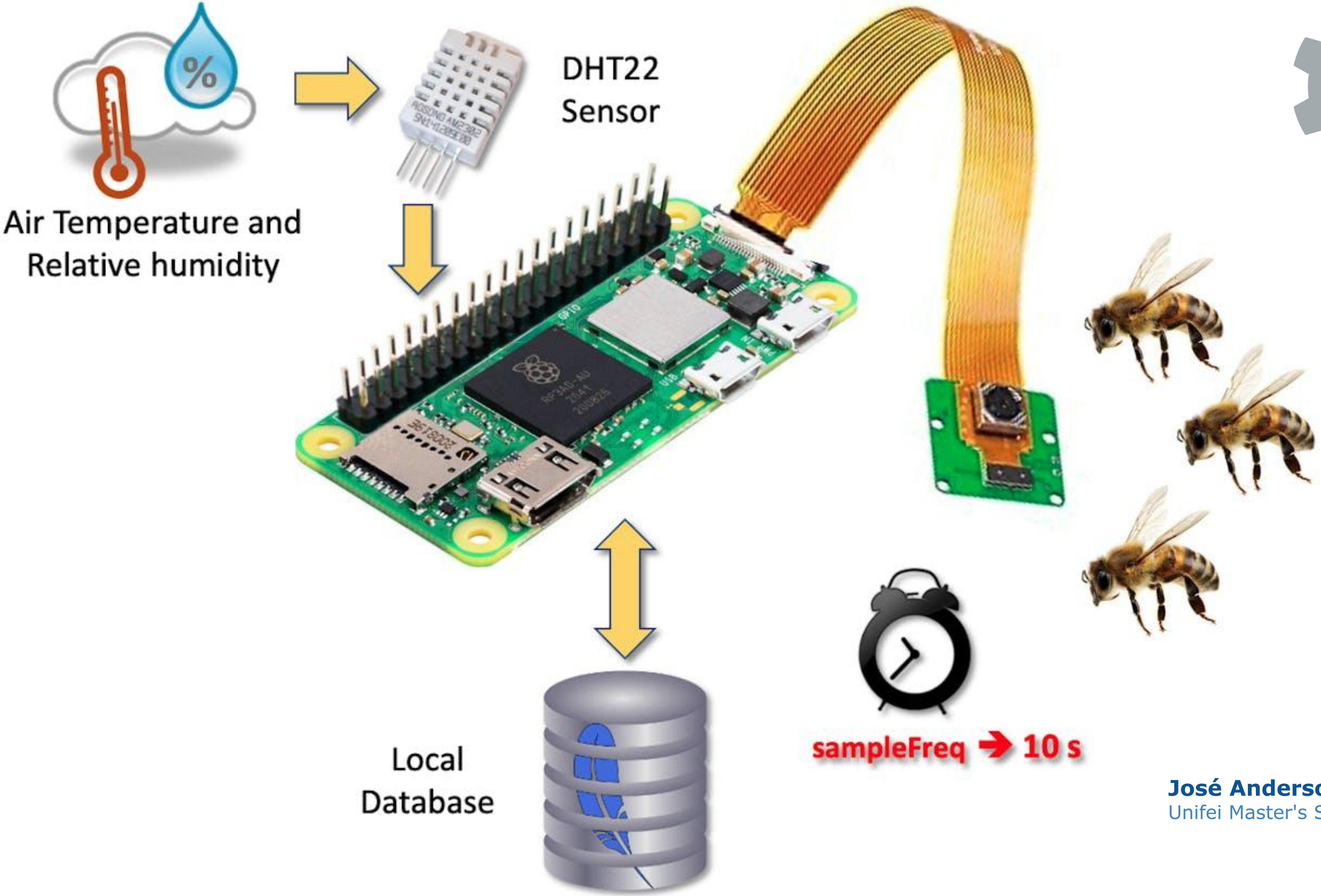


https://www.edgeimpulse.com/blog/the-world-is-all-abuzz-about-tinyml

Coffee Disease Classification

João Vitor Yukio Bordin Yamashita
Engenheiro - UNIFEI

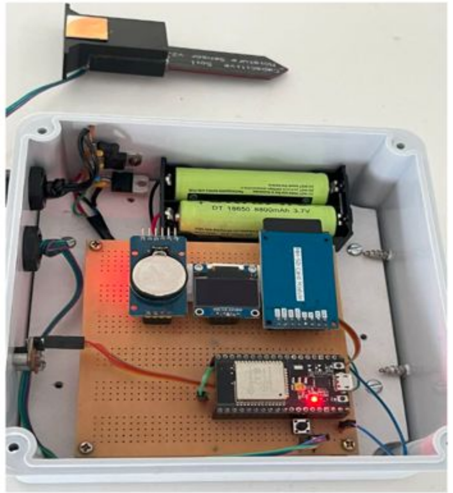
https://www.hackster.io/Yukio/coffee-disease-classification-with-ml-b0a3fc

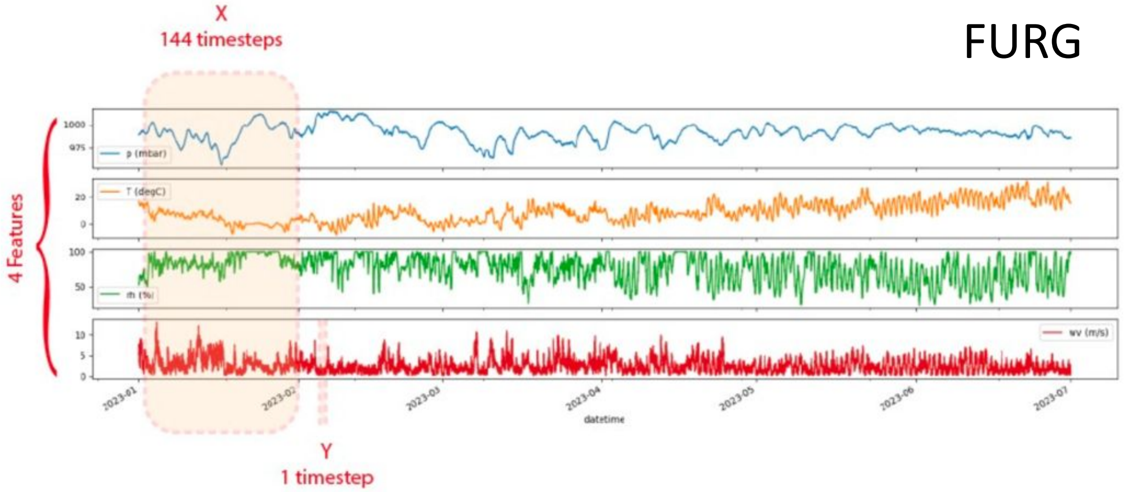


José Anderson Reis
Unifei Master's Student

LSTM

FURG UTEC UNRaf





ESP32 LSTM Phenolic Sponge Moisture

YOLO

FURG UTEC UNRaf

Ant Detection

Real-World Applications

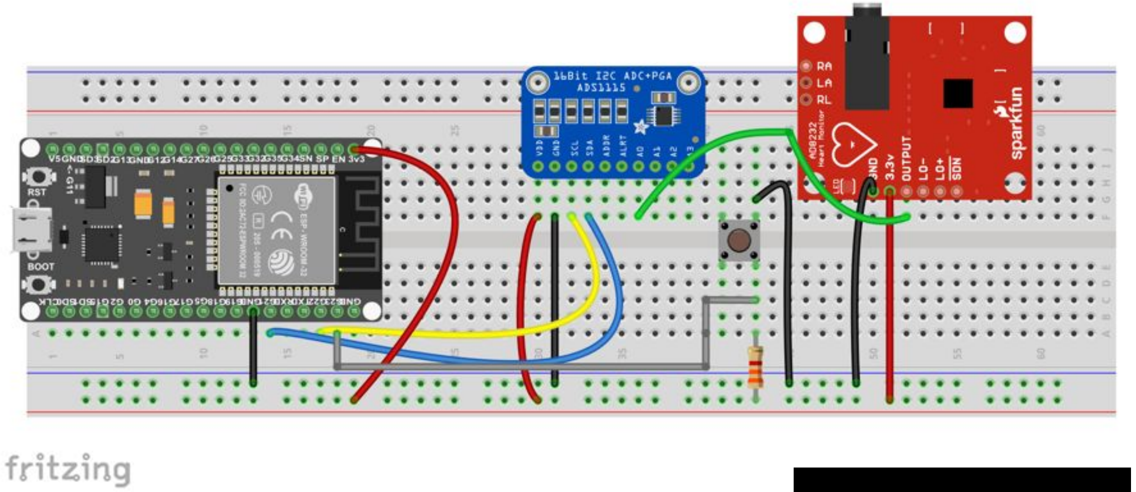
Agriculture

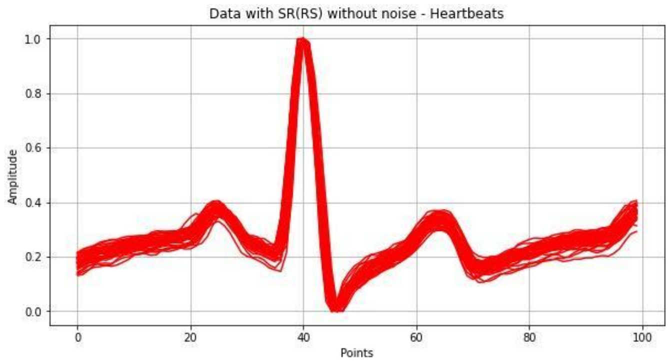
Healthcare

Industry

Environment

AD8232 - Single Lead Heart Rate Monitor

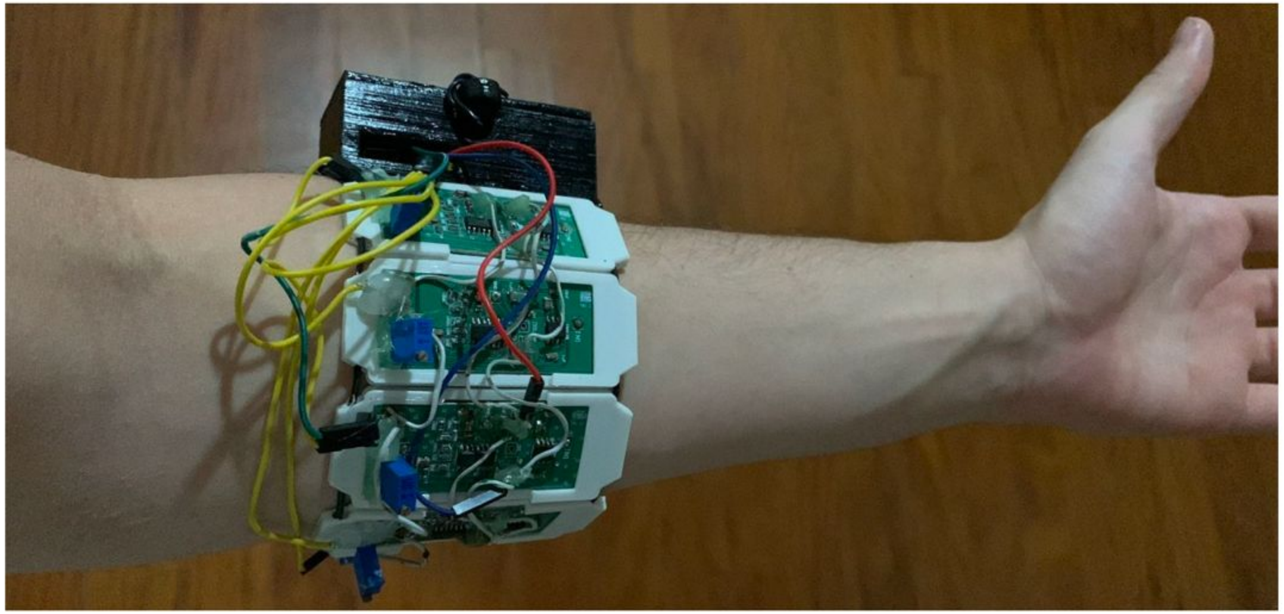


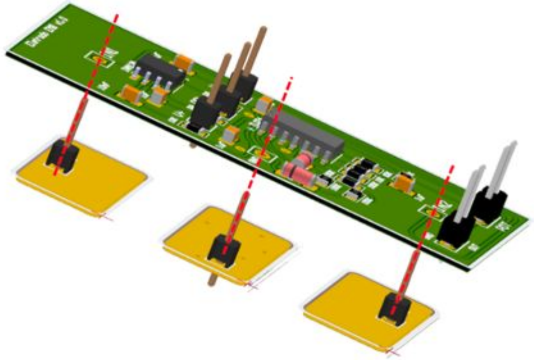


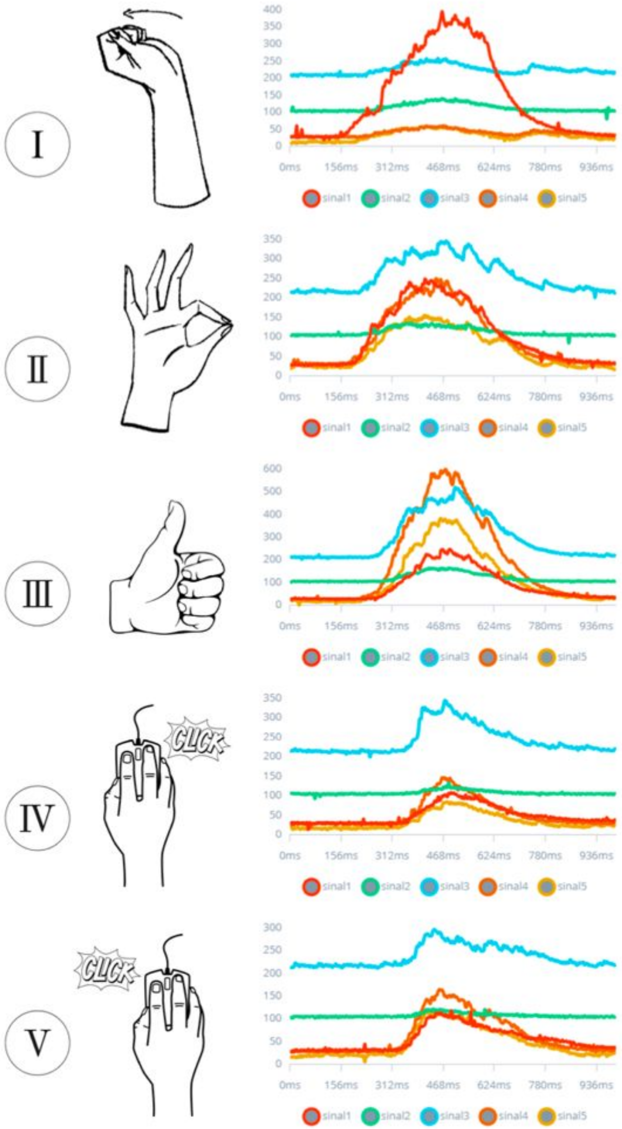
[Atrial Fibrillation Detection on ECG using TinyML
Silva et al. UNIFEI 2021]

Guilherme Silva
Matheus Lima
Engenheiros - UNIFEI

Surface electromyography

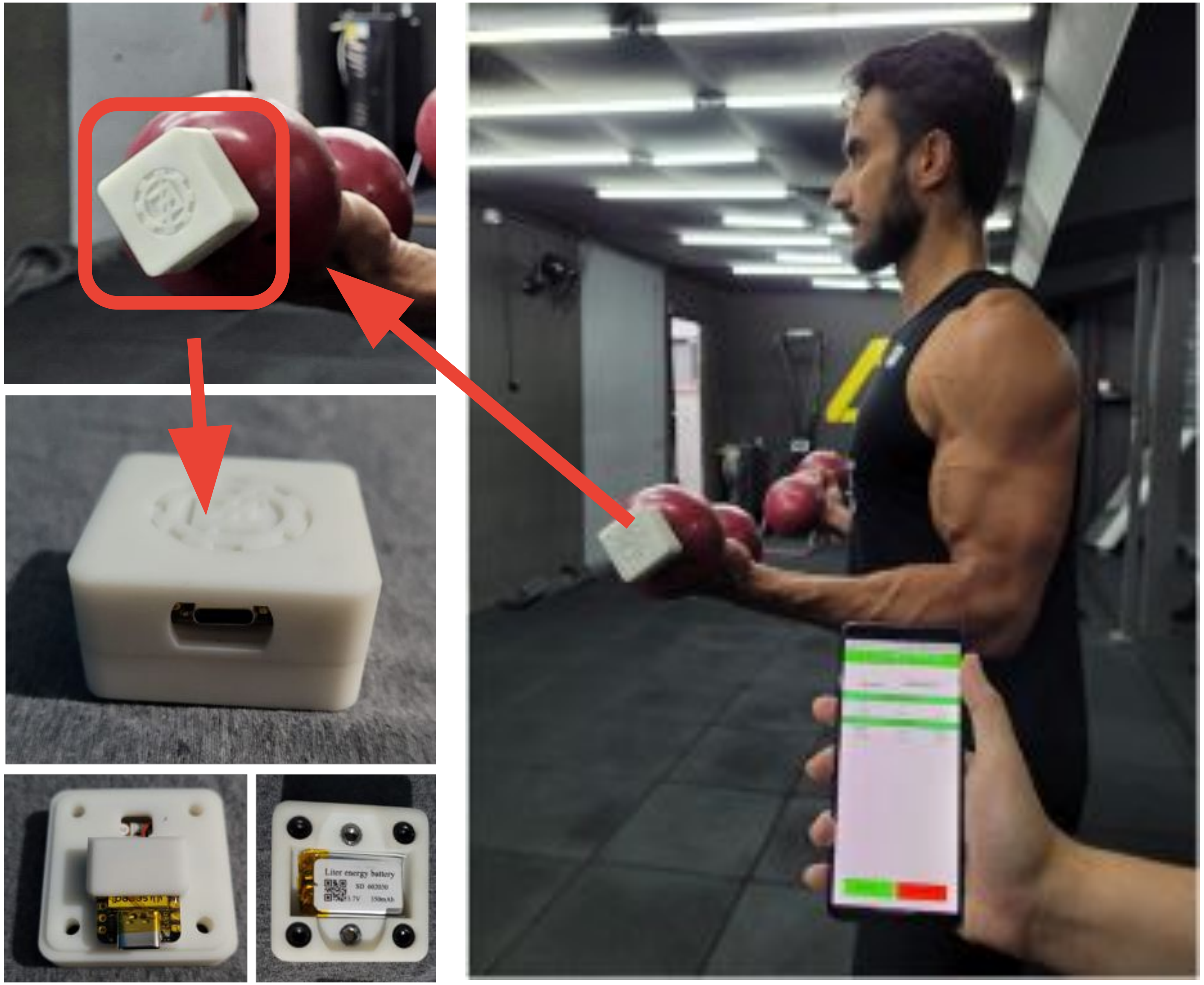






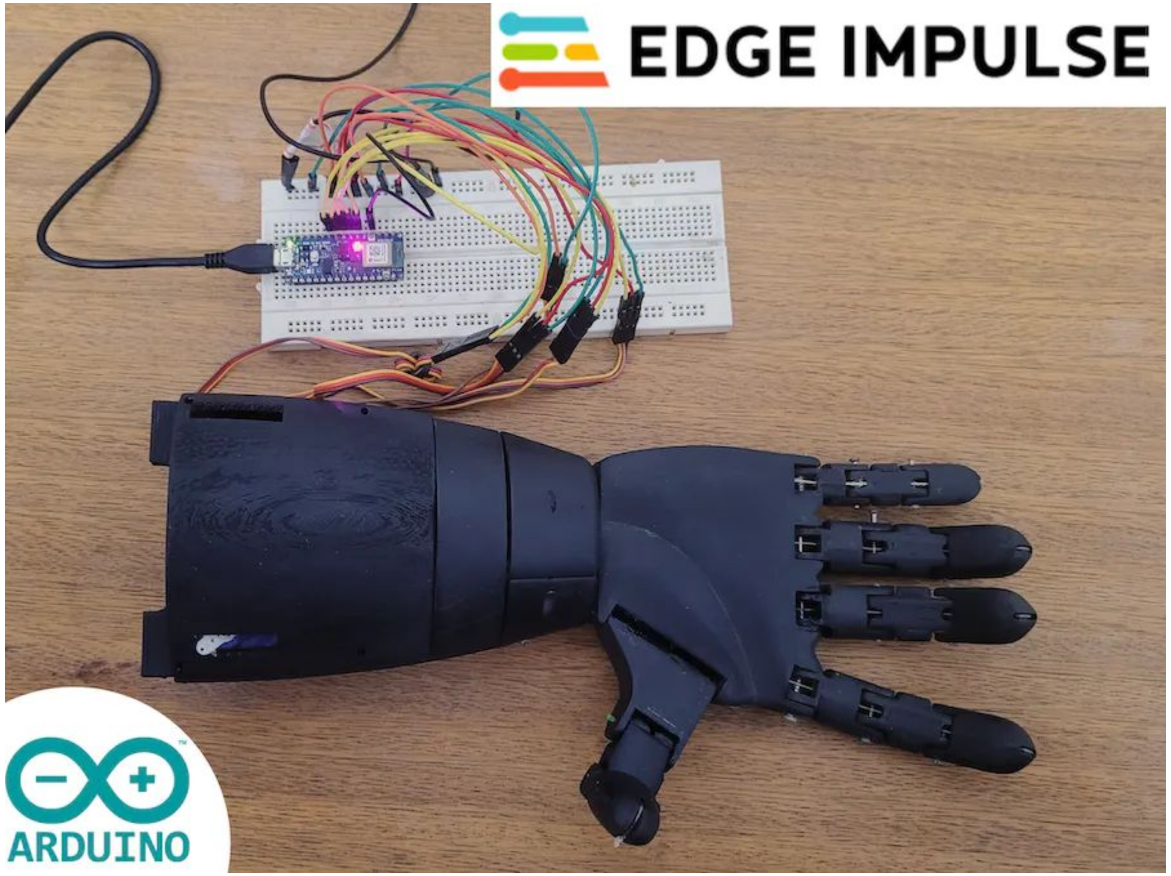
Mateus F. Delangélica e
Renato M. Neto, UNIFEI

Personal Trainer



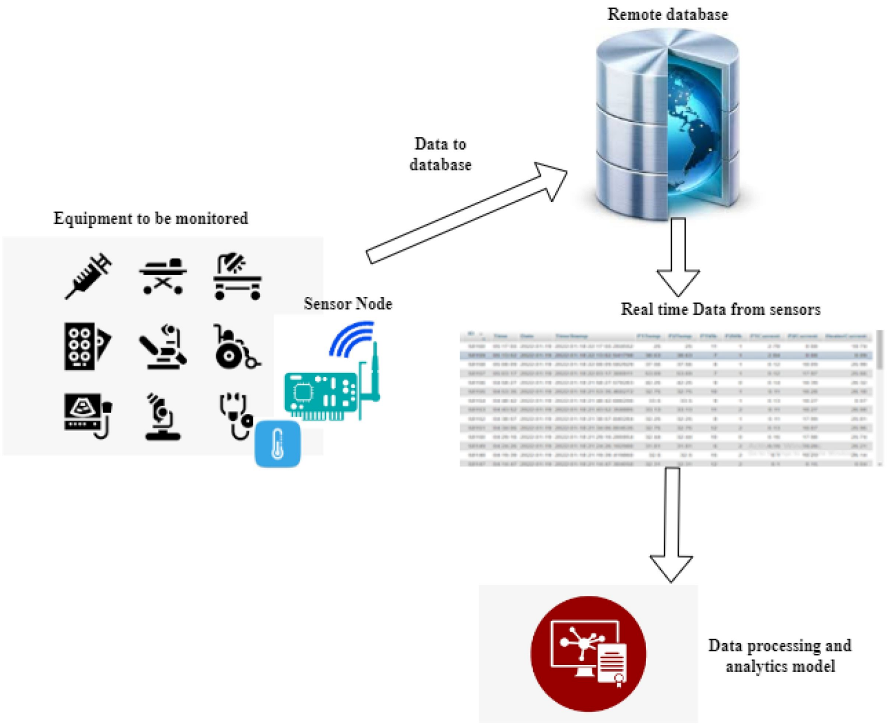
Ricardo Magno C. Junior
Luiz Fernando Kikuchi
UNIFEI

Bionic Hand Voice Commands



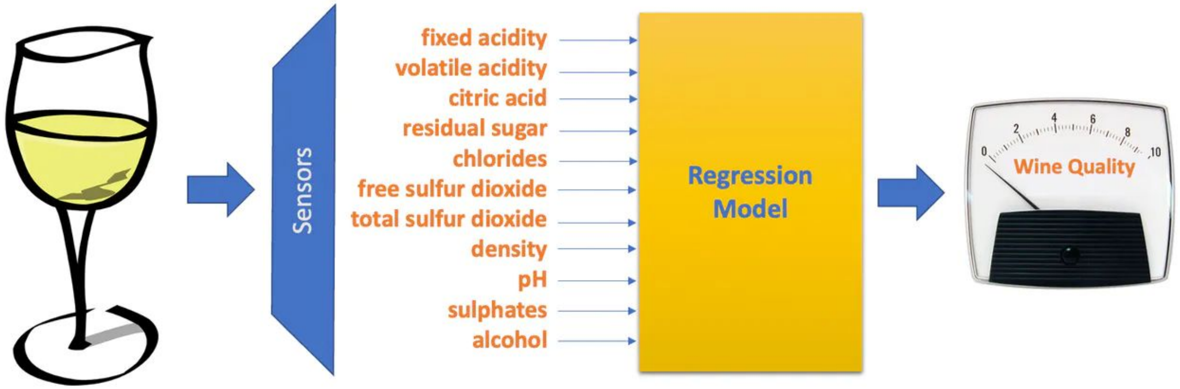
https://www.hackster.io/ex-machina/bionic-hand-voice-commands-module-w-edge-impulse-arduino-aa97e3

Regression on TinyML



[On-Device IoT-Based Predictive Maintenance Analytics Model: Comparing TinyLSTM and TinyModel from Edge Impulse]()

Sensor fusion



[TinyML Made Easy: Exploring Regression - White Wine Quality]()

Real-World Applications

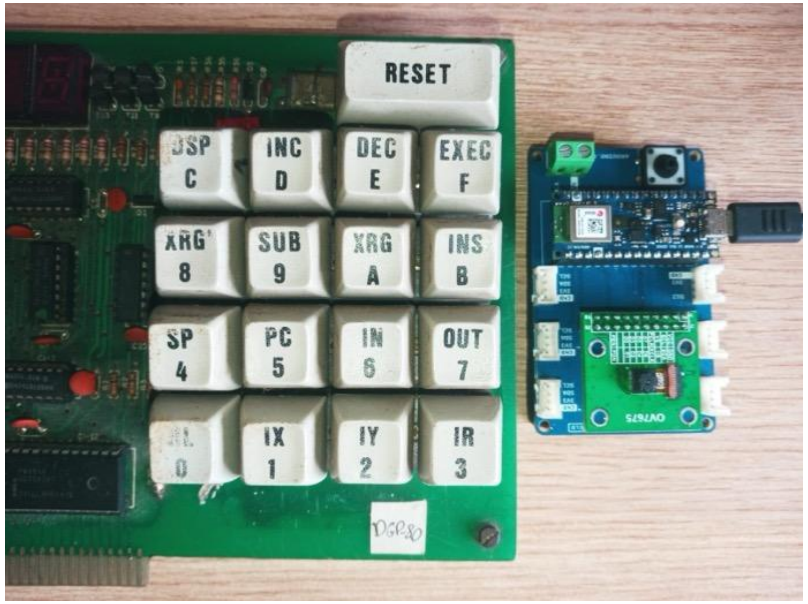
Agriculture

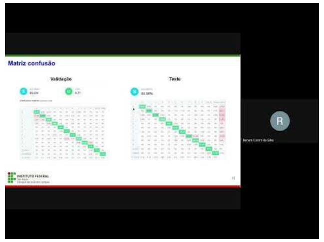
Healthcare

Industry

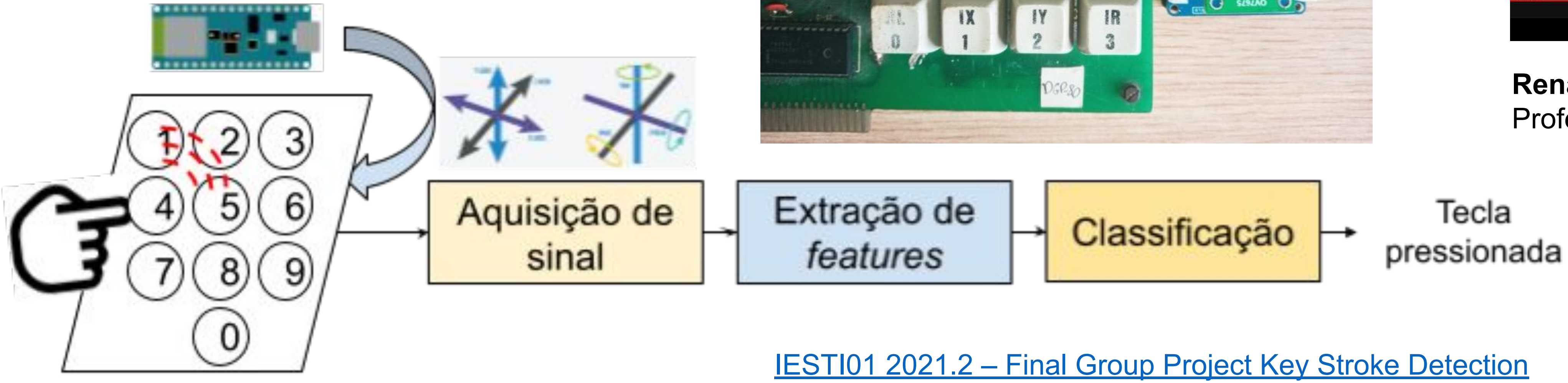
Environment

Keystroke **Sound** Detection



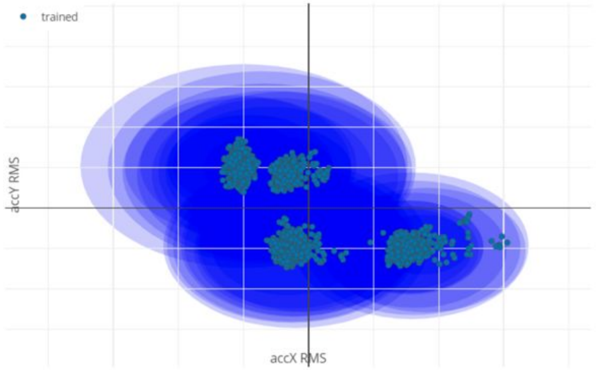


Renam Castro
Professor IFESP



[IESTI01 2021.2 – Final Group Project Key Stroke Detection]()

Industrial – Anomaly Detection



[IESTI01 2021.2 - Final Group Project: Bearing Failure Detection]

Reinforcement on TinyML

Deep Reinforcement Learning for Autonomous Source Seeking on a Nano Drone

Bardienus P. Duisterhof[1,3] Srivatsan Krishnan[1] Jonathan J. Cruz[1] Colby R. Banbury[1] William Fu[1]

Aleksandra Faust[2] Guido C. H. E. de Croon[3] Vijay Janapa Reddi[1,4]

[1]Harvard University, [2]Robotics at Google, [3]Delft University of Technology, [4]The University of Texas at Austin

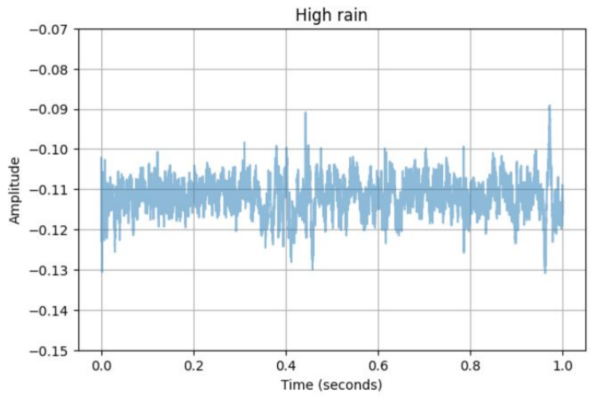
https://arxiv.org/abs/1909.11236

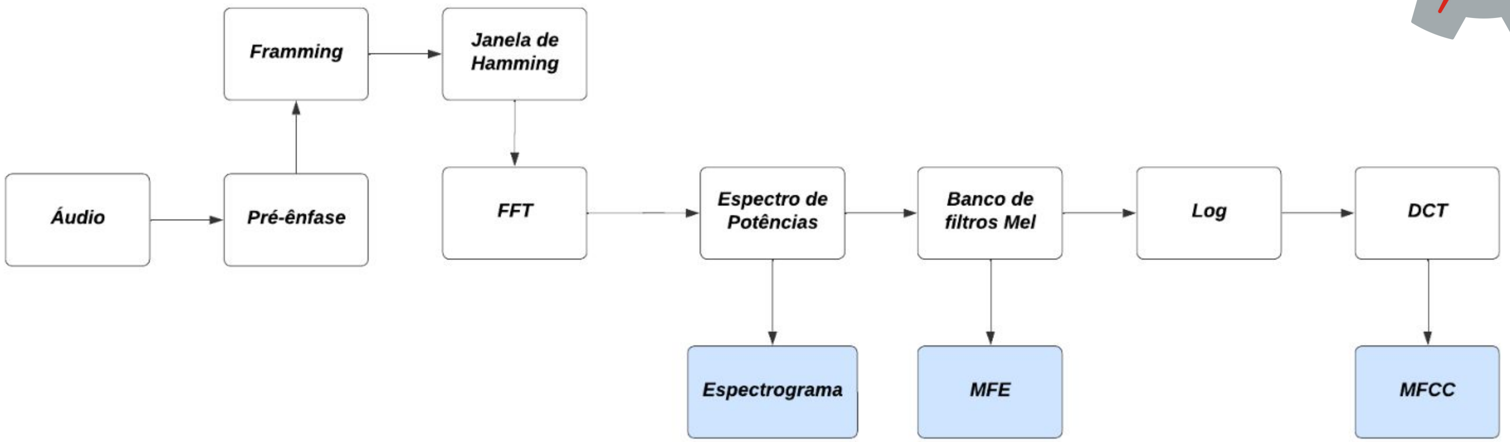
https://youtu.be/wmVKbX7MOnU

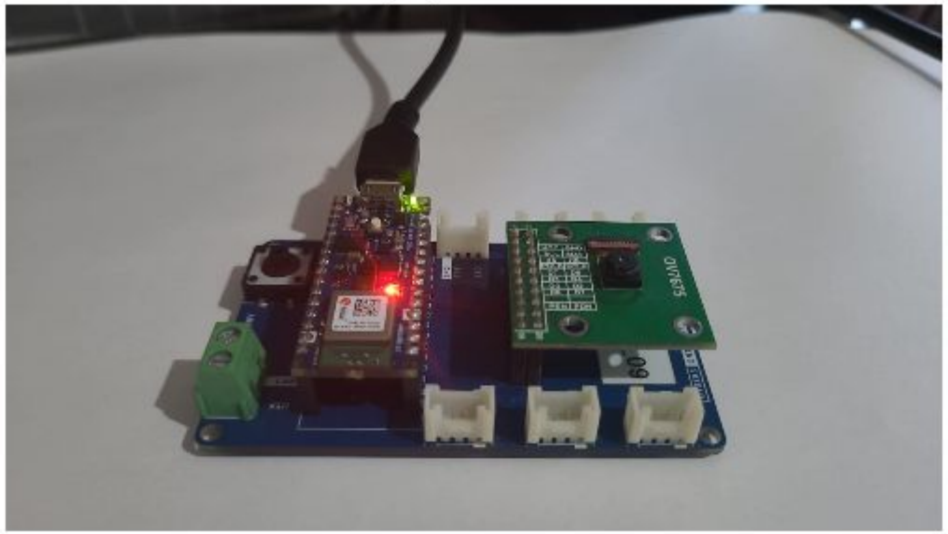
Real-World Applications

- Agriculture
- Healthcare
- Industry
- Environment

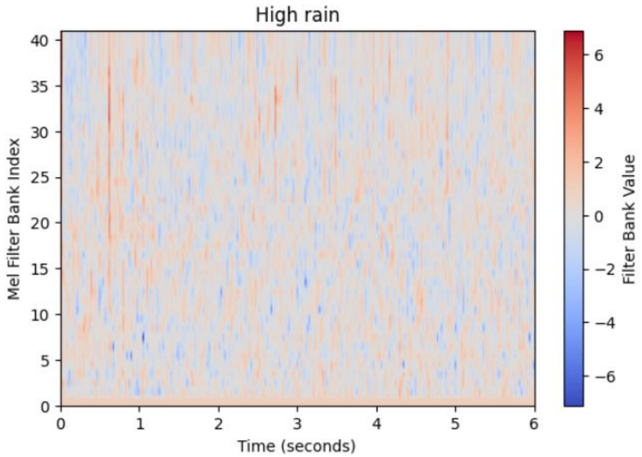
Measure rainfall using sound detection



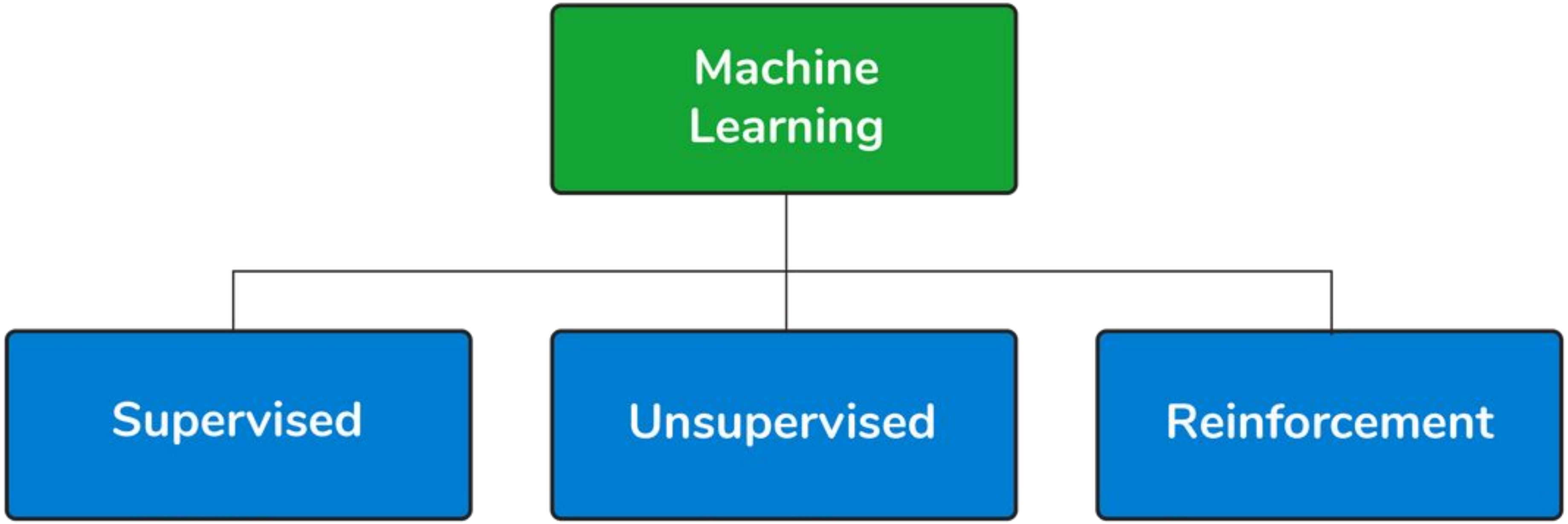




```
Edge Impulse standalone inferencing
run_classifier returned: 0
Timing: DSP 630 ms, inference 47 ms,
Predictions:
  HighRain: 0.99609
  LowRain: 0.00000
  MediumRain: 0.00000
  NoRain: 0.00000
```



Bruno V. Batista , Felipe S. Fernandes, Rodrigo N. Pereira - UNIFEI



Task driven

Data driven

Learns to react to an environment

- Regression
- Classification
- Object Detection

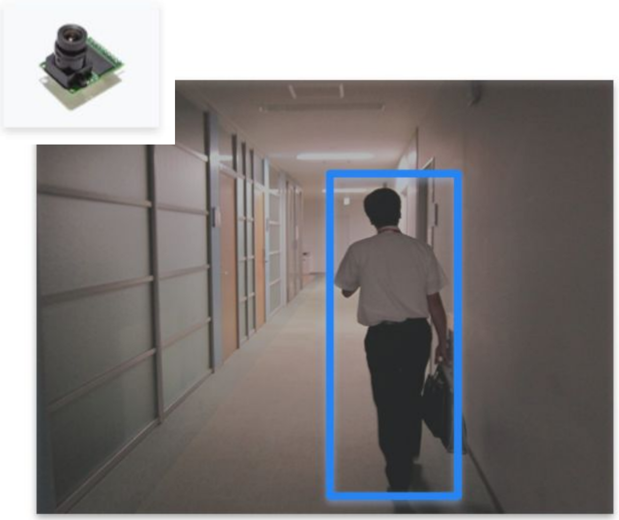
- Anomaly Detection

- Autonomous Navigation

Sound

Vibration

Vision



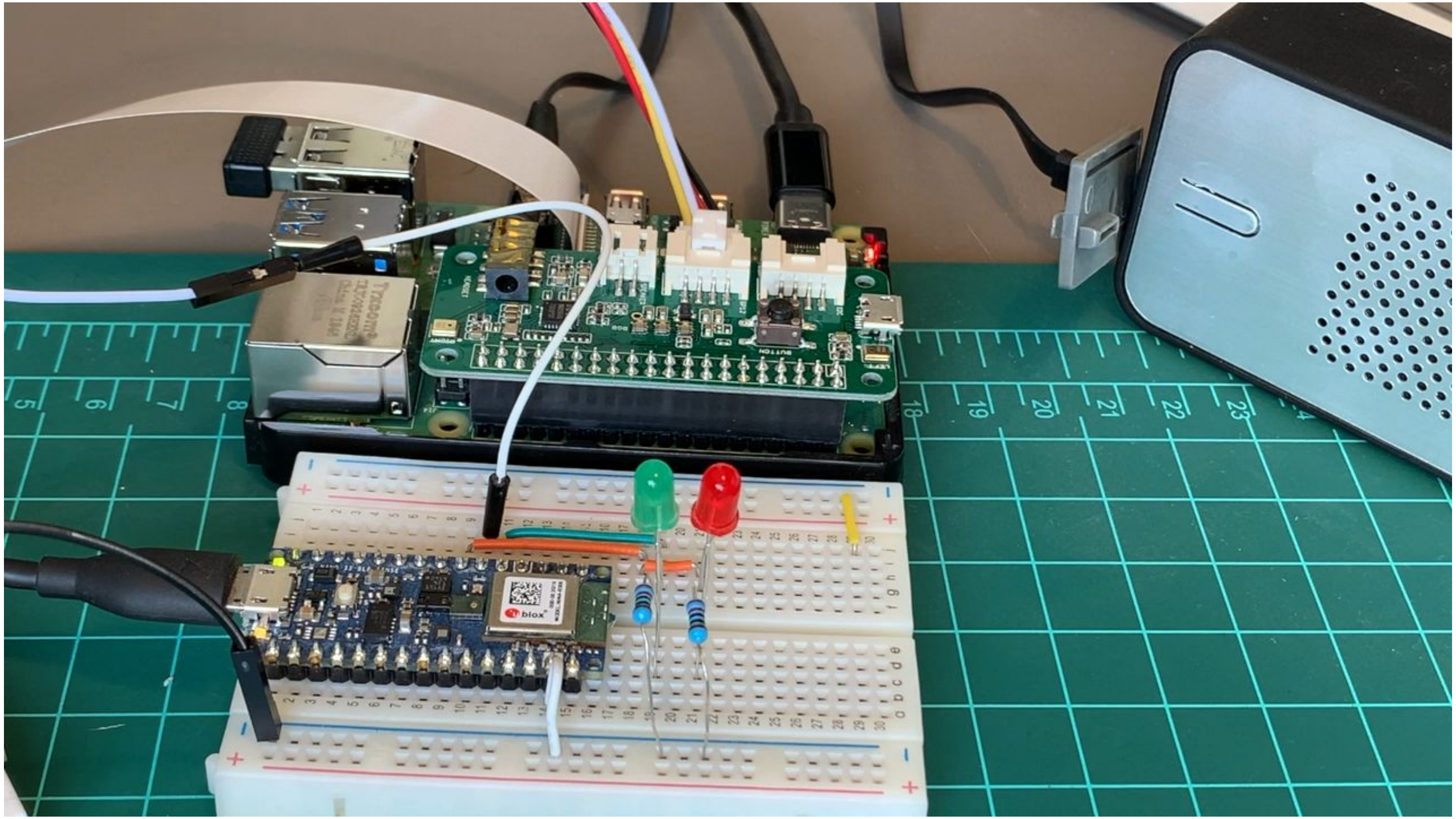
Sound

Vibration

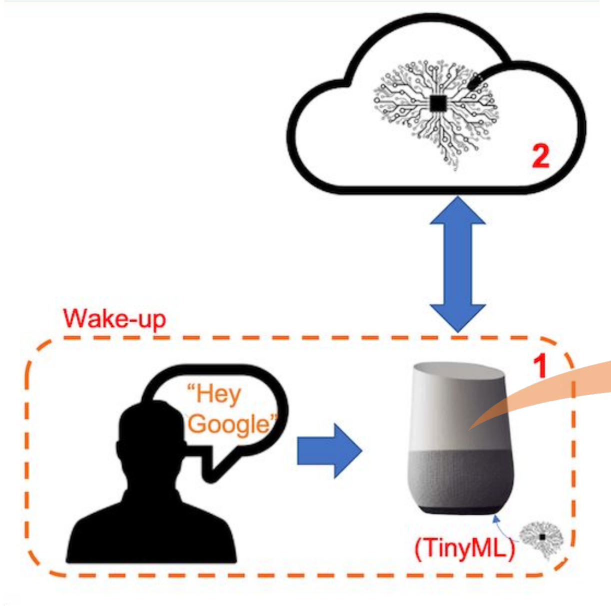
Vision

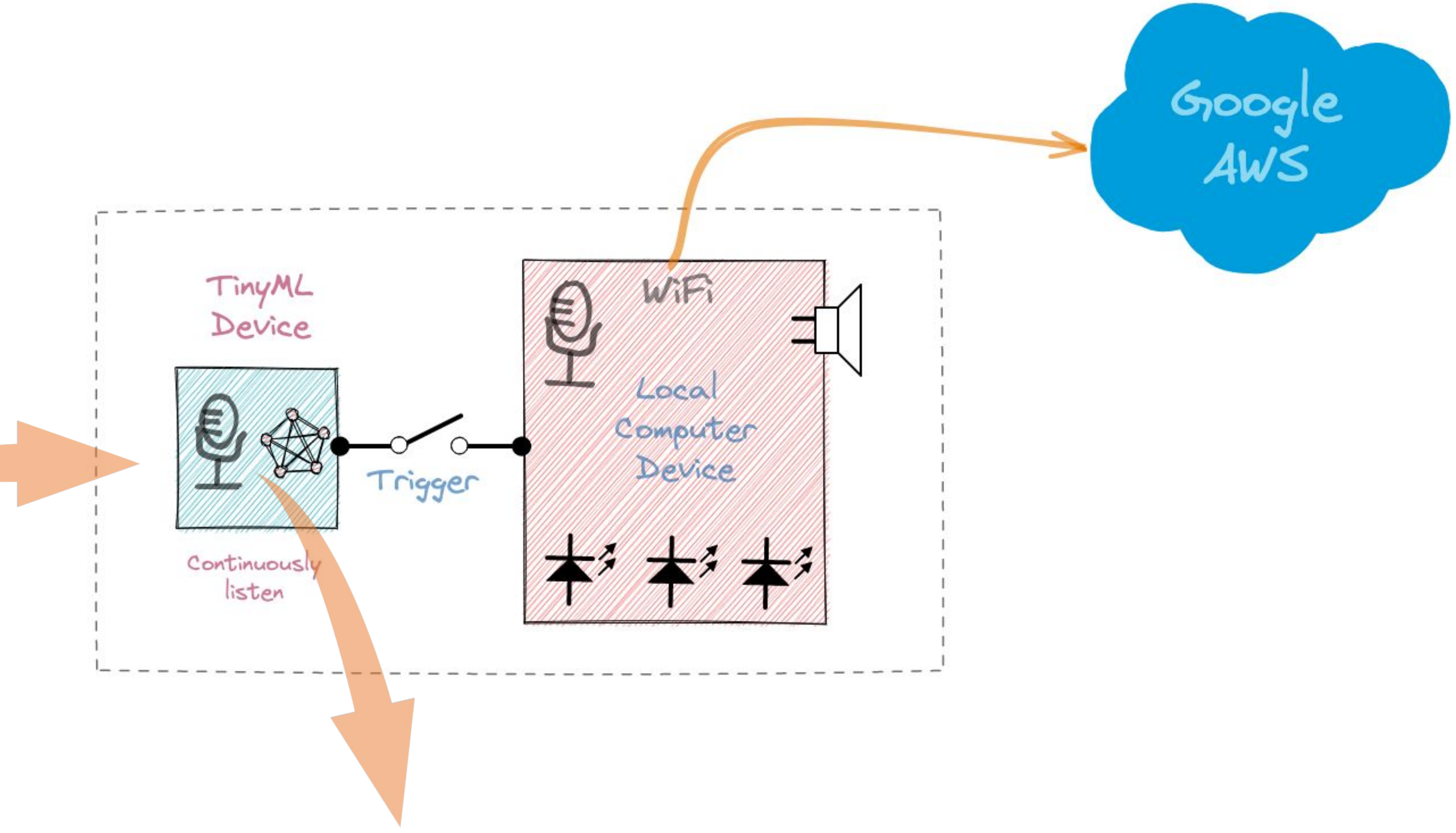
Personal Assistant

Personal Assistant



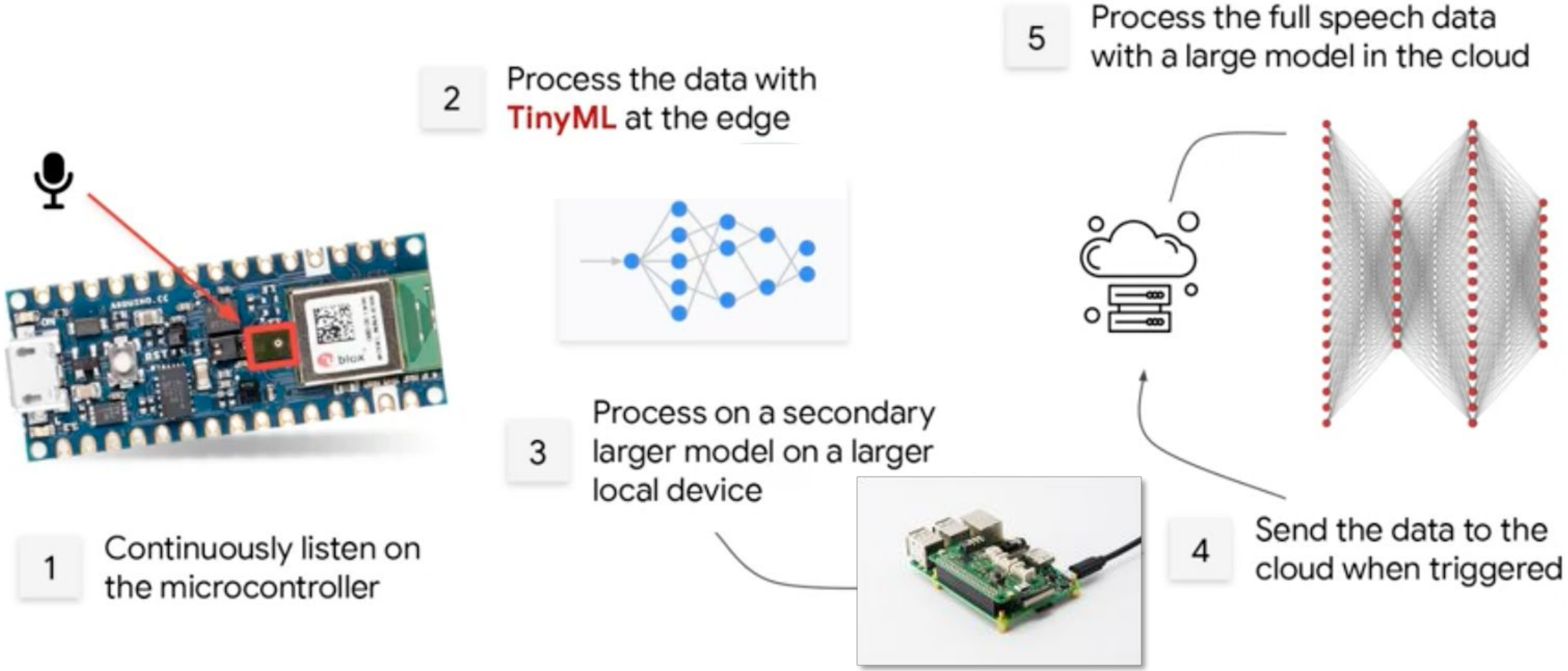
Personal Assistant



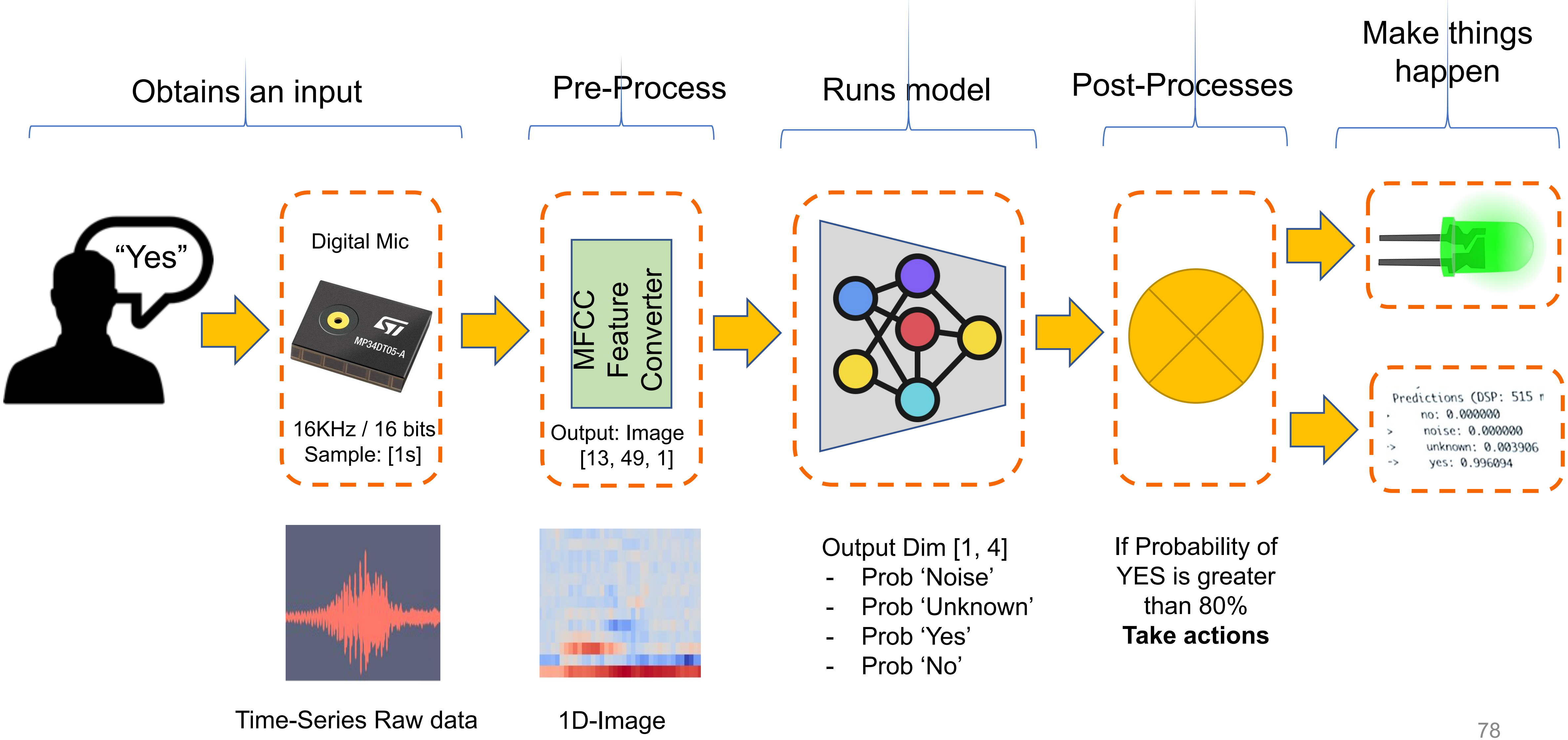


KWS

“Cascade” Detection: multi-stage model



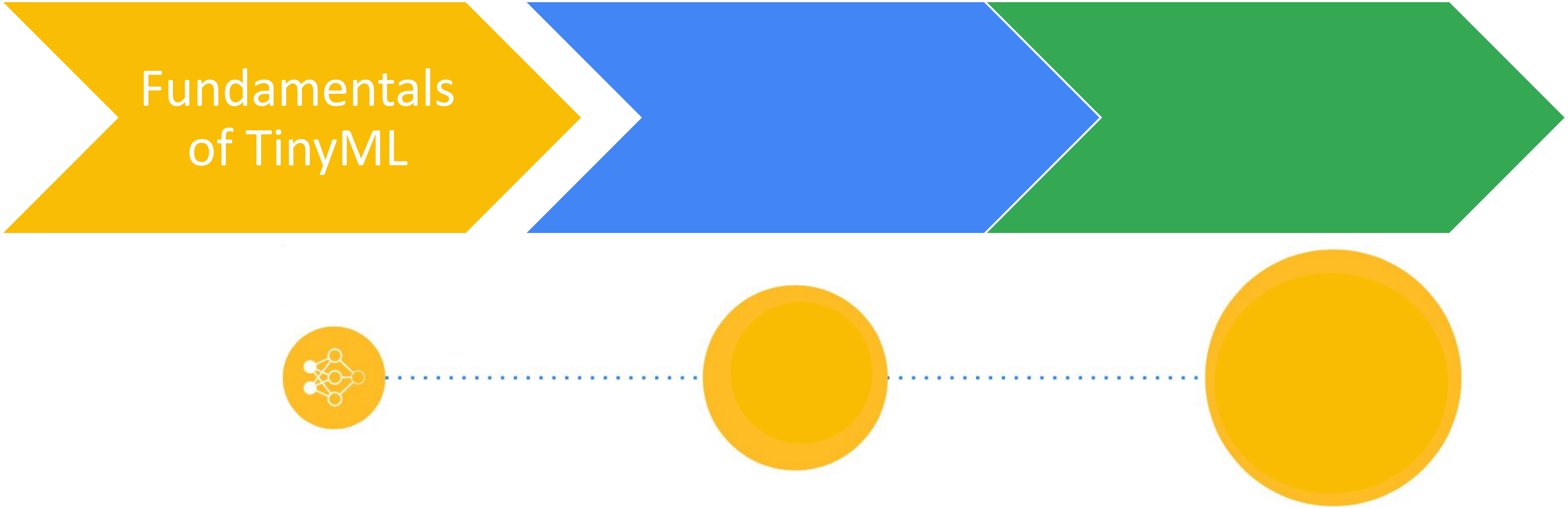
KeyWord Spotting (KWS) - **Inference**



What will We learn?

What will We learn?

Part 1



Part 1 is all about talking about what is the language of machine learning

What we will learn?

Part 1 Part 2

In Part 2, we will get a sneak peek into the variety of different TinyML applications, as keyword spotting ("Alexa"), gesture recognition, understand how to leverage the sensors, and so forth.

What we will learn?

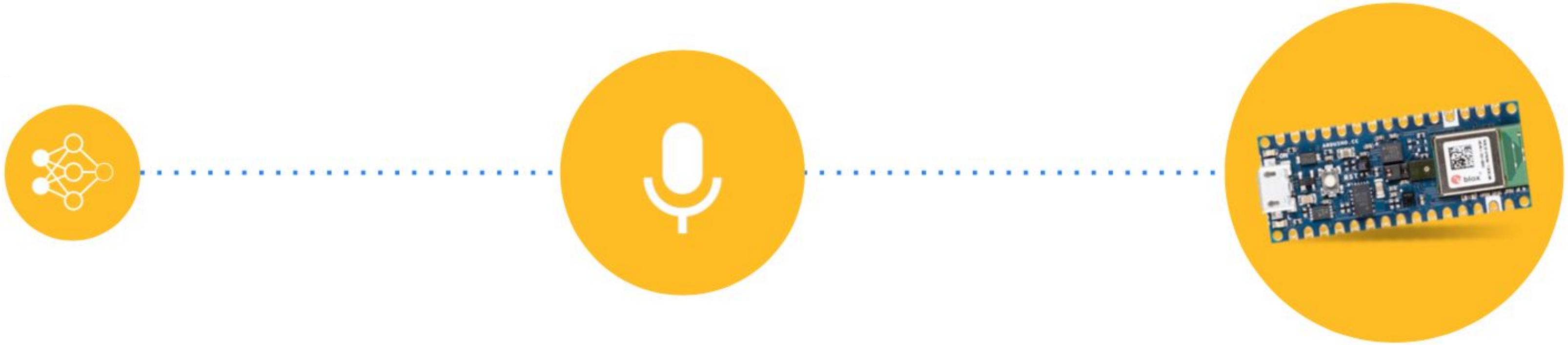
Part 1

Part 2

Fundamentals of TinyML

Applications of TinyML

Deploying TinyML



In Part 2, we will **also** learn how to deploy models on a real microcontroller. Along the way we will explore the challenges unique to and amplified by TinyML (e.g., preprocessing, post-processing, dealing with resource constraints).

Background Requirements

Part 1

Part 2

Fundamentals of TinyML

Applications of TinyML

Deploying TinyML

Fundamentals of TinyML
- Python
- TensorFlow
- Google Colab
    - Jupyter Notebook

Applications of TinyML
- Python
- TensorFlow (Lite)
- Google Colab
- Edge Impulse Studio

Deploying TinyML
- Python
- TensorFlow (Lite-Micro - RTLite)
- Google Colab
- Edge Impulse Studio
- IDE (as Arduino)
- C/C++

This course combines **computer science** with **engineering** to feature real-world application case studies that examine the challenges facing **TinyML deployments**.

What areas we will learn?

Machine Learning

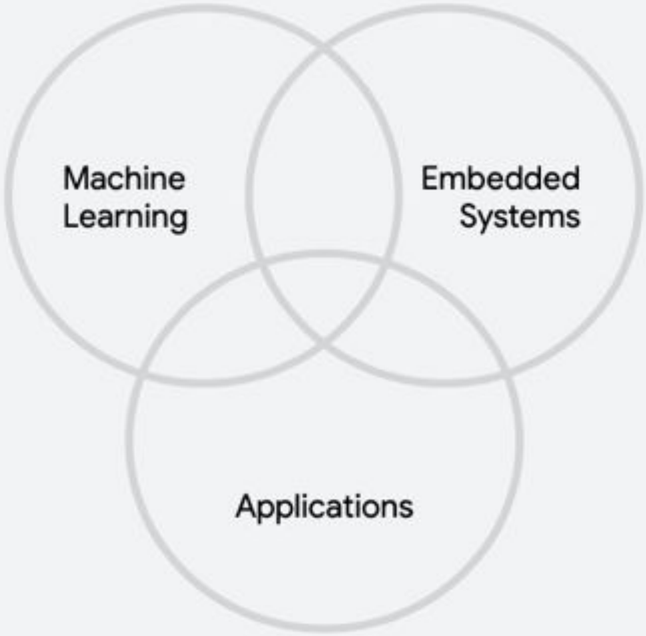
Applications

Embedded Systems

We will learn the **fundamentals of each of these areas**, just enough to focus on the goal of being able to build **TinyML applications**.

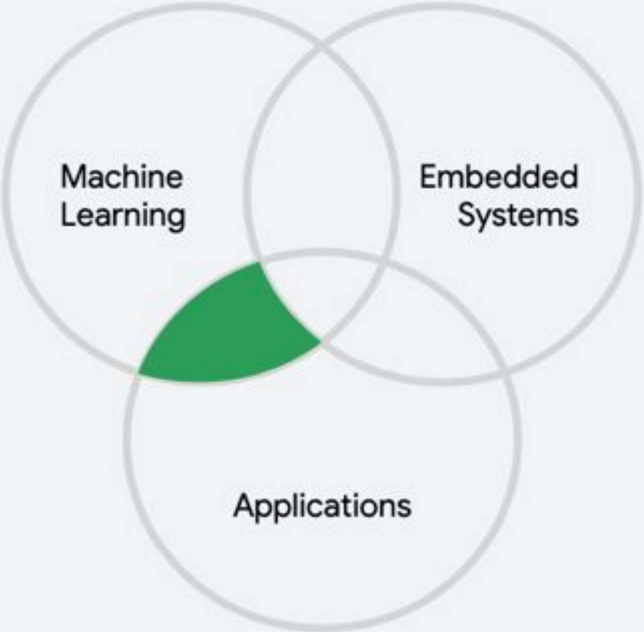
Interactions

In addition, we will bring these diverse topics together to reveal the interesting learnings at the various **intersections**



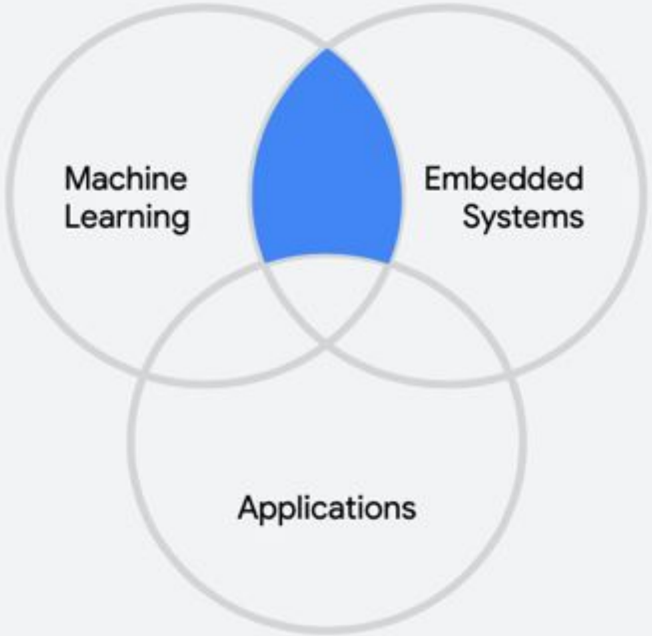
Interactions

How **machine learning** can enable new and interesting **TinyML applications**?



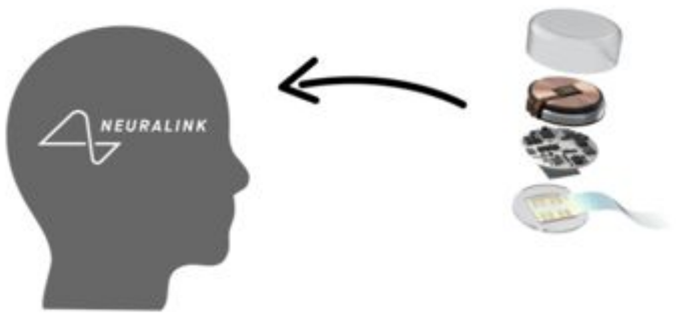
Interactions

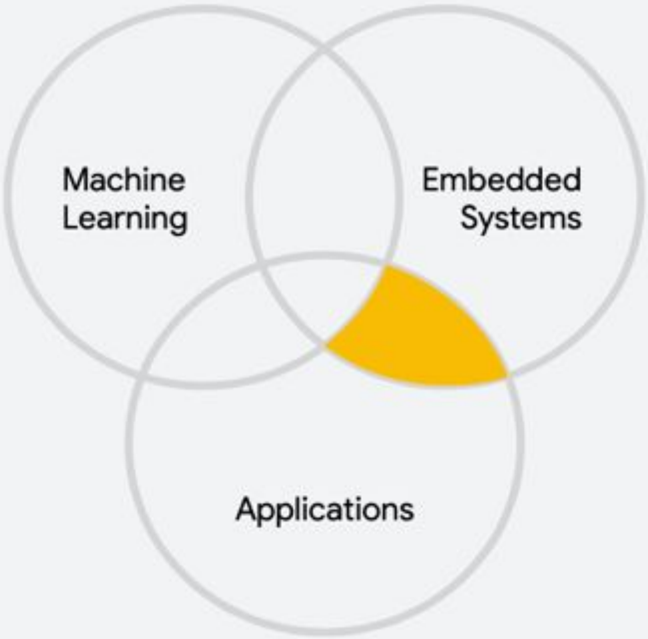
What are the **challenges** with enabling **machine learning** on **tiny**, resource-constrained **embedded devices**?



Interactions

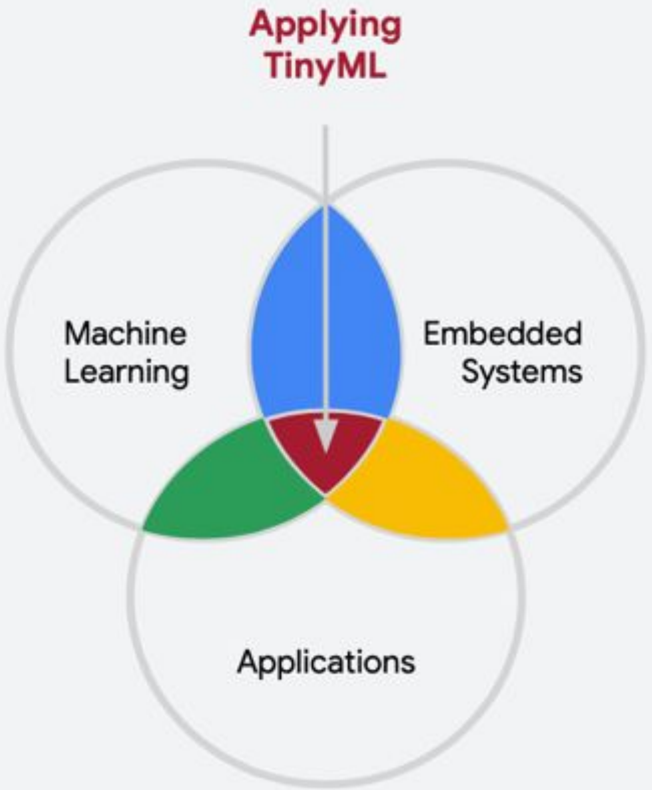
What type of new **use cases** can we possibly enable on **embedded systems** that we could not otherwise do before?





At the End of the Day

Given your understanding of things at these various intersections, you will have a deep understanding for **how to apply TinyML**



How are we going to get there?

Hands-on Learning

- **Software**
  - Machine Learning (TensorFlow)
  - Programming environments (Google Colab or Jupyter)
  - Edge Impulse Studio
- **Hardware**
  - Arduino Nano 33 BLE Sense
  - Wio Terminal (optional)
  - ESP32S3 (optional)

TinyML Arduino Kit
(Principal)

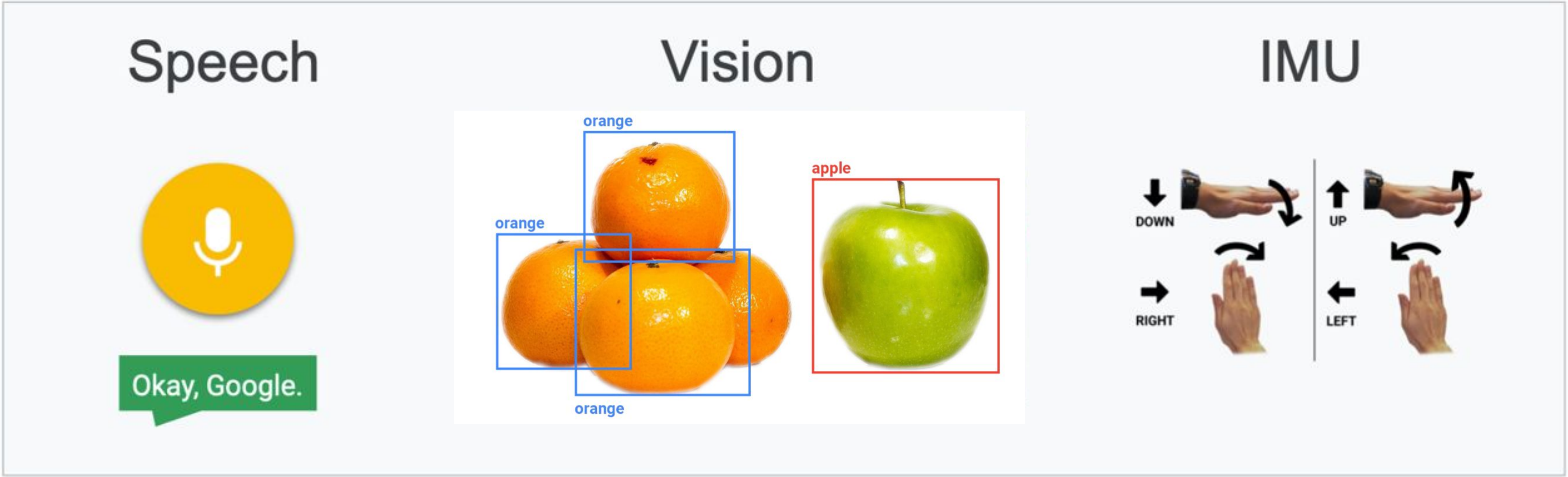
Thanks to:

Wio Terminal Kit
(Optional)

Thanks to:

XIAO ESP32S3 Kit
(Optional)

Hands-on Activities



How is the course structured?

Course Structure

- Weekly video-recorded lectures (15 weeks)
  - Slides
  - Hands-on coding (by teacher & students)
- Weekly Additional Readings
- Assignments
  - Quizzes (Weekly)
  - Notebooks with codes (4)
  - Hands-on lab reports (4)
- Final Project (Groups of 3 students)
  - Report
  - Presentation

https://github.com/Mjrovai/UNIFEI-IESTI01-TinyML

Class planning and approval process

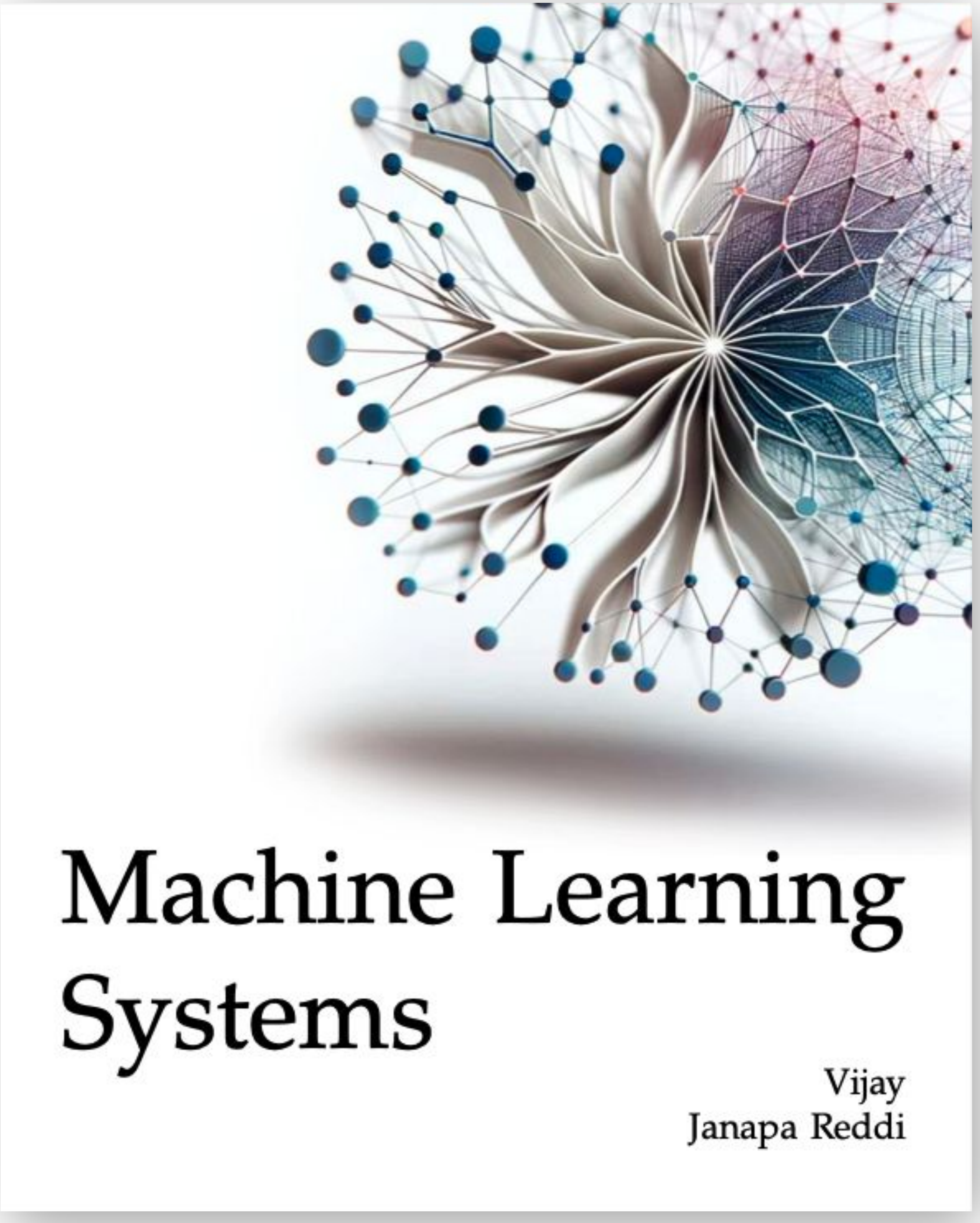
- Minimal suggested Workload (4 hours per week):
  - 30 hours (Weekly recorded classes of about 2h, for 15 weeks)
  - 15 hours of assignments/coding/labs
  - 15 hours in research, individual studies, and final project (in a group)

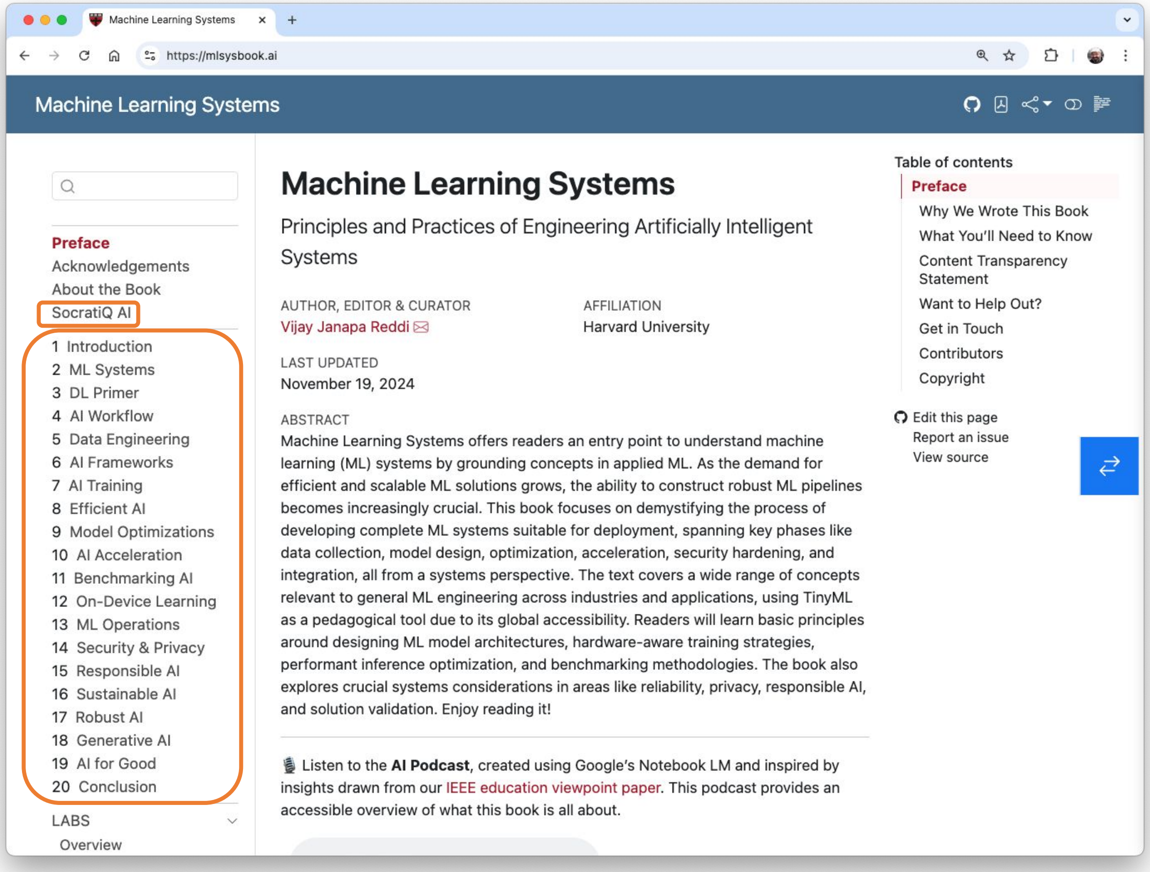
- Approval process:
  - 1st Evaluation:
    - Individual **Quizzes**: 10%
    - Individual **Exercise Lists** (Notebooks): 25%
    - Group **Project Proposal**: 15%
  - 2nd Evaluation
    - Individual **Quizzes**: 10%
    - Individual **Practical Projects** (Lab reports): 25%
    - Group **Project Presentation (*)** and Final Report: 15%

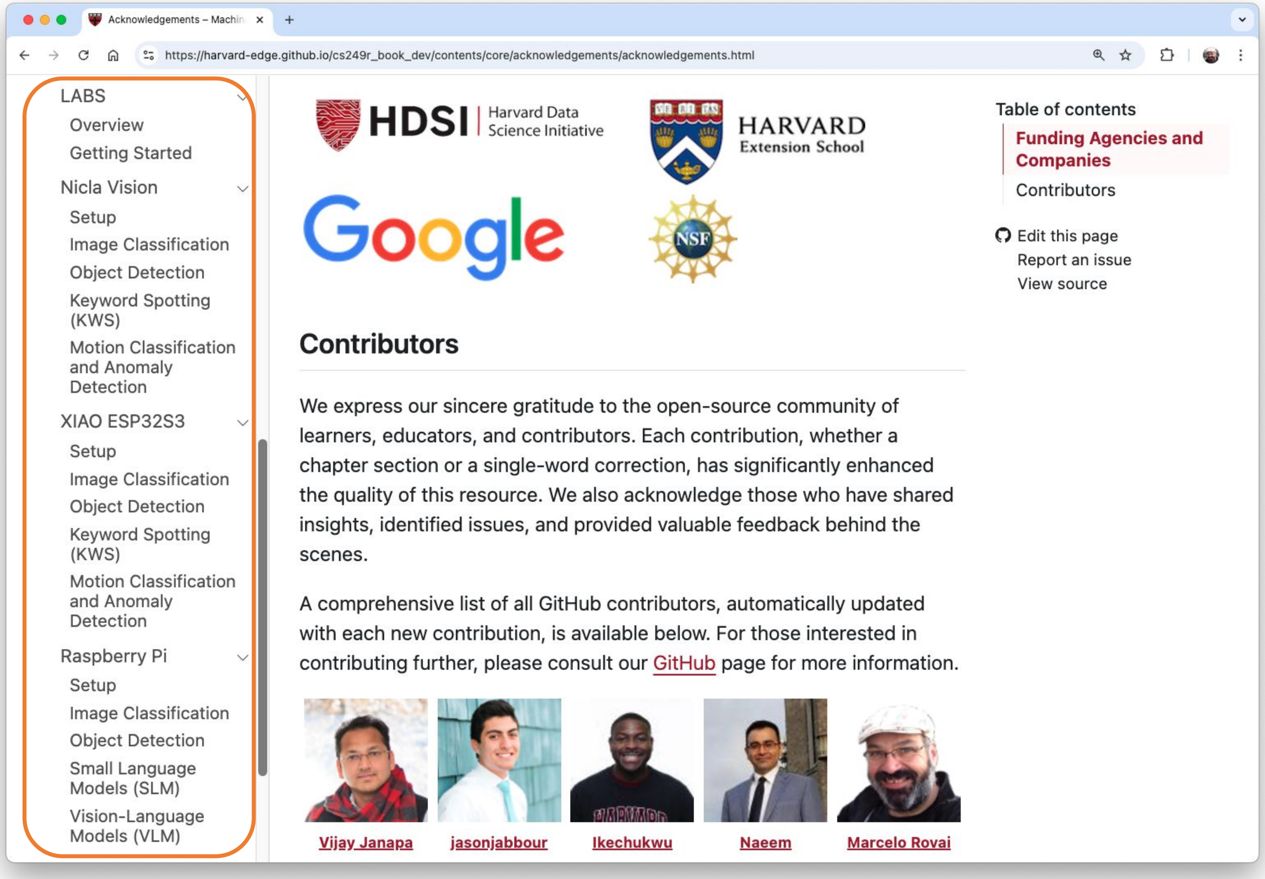
(*) Examples: Harvard CS249r – TinyML - Final Projects
Examples: UNIFEI IESTI01 – TinyML – Final Projects

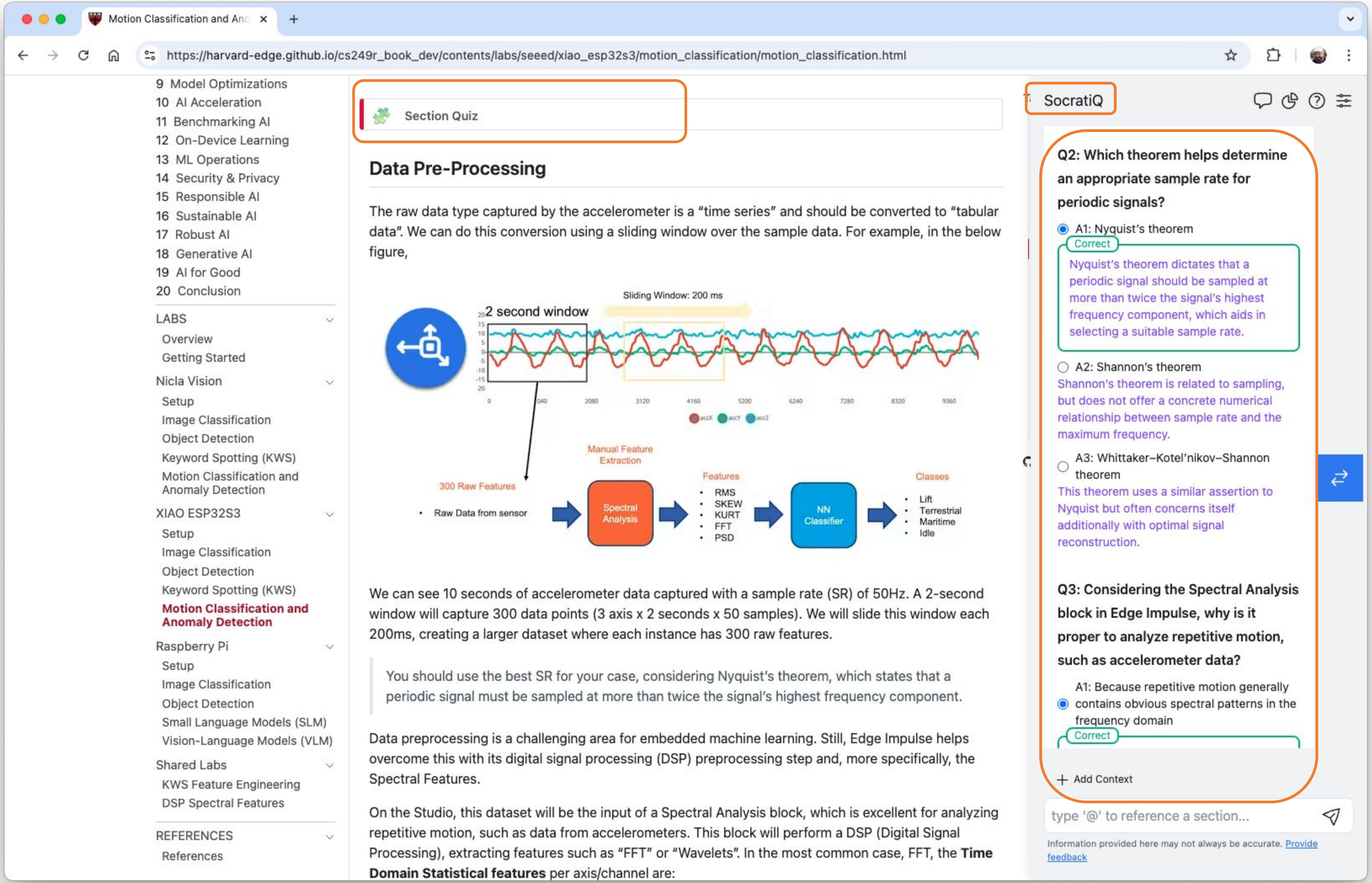
	Date	Class	Content	Assigment Deadline	ML Systems e-Book (Additional Reading)
Fundamentals	11/03/25	1 2	About the Course and Syllabus Introduction to TinyML		Chapter 1 - Introduction
	18/03/25	3 4	TinyML - Challenges - Embedded Systems TinyML Challenges - Machine Learning	Pre-Survey / Quiz 1	Chapter 2- ML Systems
	25/03/25	5 6	The Machine Learning Paradigm The Building Blocks of Deep Learning (DL) - Introduction	Quiz 2	Chapter 3 - Deep Learning Primer (3.1 - 3.3)
	01/04/25	7 8	The Building Blocks of DL - Regression with DSS The Building Blocks of DL - Classification with DSS	List 1 / Quiz 3	Chapter 3 - Deep Learning Primer (3.4 - 3.8)
	08/04/25	9 10	The Building Blocks of DL - DNN Recap, Datasets and Model Performance Metrics Introducing Convolutions (CNN)	List 2 / Quiz 4	Chapter 4 - DNN Architectetures (4.1 - 4.3)
	15/04/25	11 12	Image Classification using CNN Introduction to Edge Impulse – CNN with Cifar-10	List 3 / Quiz 5	Chapter 8 - AI Training
	22/04/25	13 14	Preventing Overfitting Fundamentals wrap-up and Application's preview	List 4 / Quiz 6	Chapter 19 - AI For Good
Application & Deploy	29/04/25	15 16	ML Applications Overview - AI Lifecycle and ML Workflow Introduction to TFLite and TFLite-Micro	Project Proposal / Quiz 7	Chapter 5 - AI Workflow
	06/05/25	17 18	TinyML Kit Overview - HW and SW Installation & Test TFLite-Micro Overview & Hello World Code Walkthrough	Quiz 8	Chapter 7 - AI Frameworks
	13/05/25	19 20	Motion Classification - Introduction Motion Classification using MCU (Nano 33)	Quiz 9	Chapter 6 - Data Engineering
	20/05/25	21 22	K-means Clustering & Anomaly Detection Anomaly Detection Hands-On Lab & Pos-Processing	Lab 1 / Quiz 10	Chapter 9 - Efficient AI
	27/05/25	23 24	Keyword Spotting - Introduction Lab KWS using MCU (Nano 33)	Lab 2 / Quiz 11	Chapter 13 - ML Operations
	03/06/25	25 26	Image Classification Introduction Image Classification using Edge Impulse Studio	Quiz 12	Chapter 15 - Security & Privacy
	10/06/25	27 28	Collecting Data - Alternative ways Responsible AI & TinyML Course Wrapup	Lab 3 / Quiz 13	Chapter 16 - Responsible AI
	17/06/25	29 30	EdgeAI - Going Further Group Presentations (Date to be defined with groups)	Lab 4 and Lab 4a	Chapter 17 - Sustainable AI

https://mlsysbook.ai/

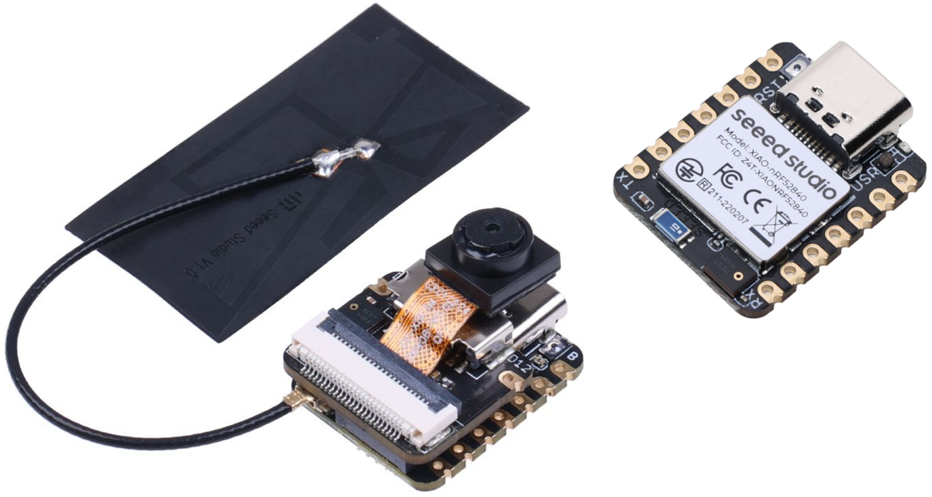



Machine Learning Systems
https://mlsysbook.ai
Machine Learning Systems
Preface
Acknowledgements
About the Book
SocratiQ AI
1 Introduction
2 ML Systems
3 DL Primer
4 AI Workflow
5 Data Engineering
6 AI Frameworks
7 AI Training
8 Efficient AI
9 Model Optimizations
10 AI Acceleration
11 Benchmarking AI
12 On-Device Learning
13 ML Operations
14 Security & Privacy
15 Responsible AI
16 Sustainable AI
17 Robust AI
18 Generative AI
19 AI for Good
20 Conclusion
LABS
Overview
Machine Learning Systems
Principles and Practices of Engineering Artificially Intelligent Systems
AUTHOR, EDITOR & CURATOR
Vijay Janapa Reddi
AFFILIATION
Harvard University
LAST UPDATED
November 19, 2024
ABSTRACT
Machine Learning Systems offers readers an entry point to understand machine learning (ML) systems by grounding concepts in applied ML. As the demand for efficient and scalable ML solutions grows, the ability to construct robust ML pipelines becomes increasingly crucial. This book focuses on demystifying the process of developing complete ML systems suitable for deployment, spanning key phases like data collection, model design, optimization, acceleration, security hardening, and integration, all from a systems perspective. The text covers a wide range of concepts relevant to general ML engineering across industries and applications, using TinyML as a pedagogical tool due to its global accessibility. Readers will learn basic principles around designing ML model architectures, hardware-aware training strategies, performant inference optimization, and benchmarking methodologies. The book also explores crucial systems considerations in areas like reliability, privacy, responsible AI, and solution validation. Enjoy reading it!
Listen to the AI Podcast, created using Google's Notebook LM and inspired by insights drawn from our IEEE education viewpoint paper. This podcast provides an accessible overview of what this book is all about.
Table of contents
Preface
Why We Wrote This Book
What You'll Need to Know
Content Transparency Statement
Want to Help Out?
Get in Touch
Contributors
Copyright
Edit this page
Report an issue
View source

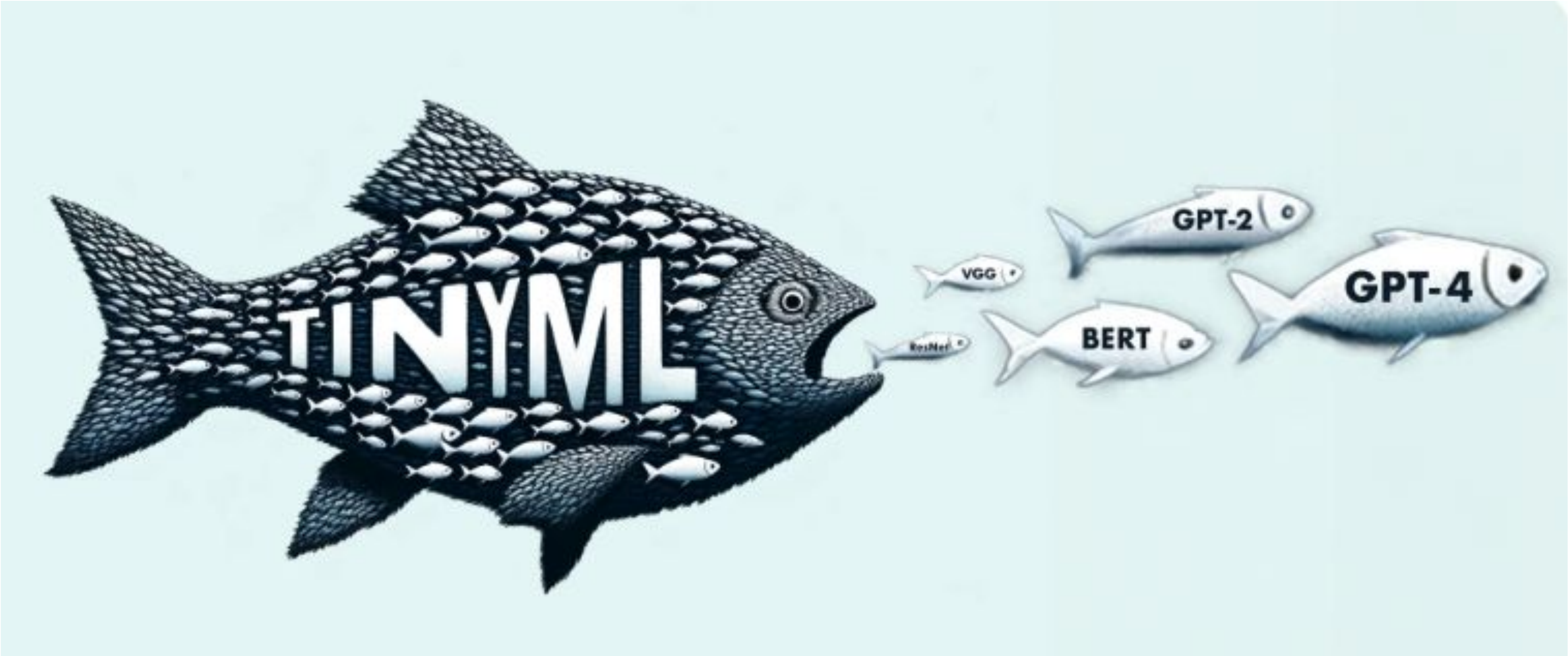

Acknowledgements – Machin
https://harvard-edge.github.io/cs249r_book_dev/contents/core/acknowledgements/acknowledgements.html
LABS
Overview
Getting Started
Nicla Vision
Setup
Image Classification
Object Detection
Keyword Spotting (KWS)
Motion Classification and Anomaly Detection
XIAO ESP32S3
Setup
Image Classification
Object Detection
Keyword Spotting (KWS)
Motion Classification and Anomaly Detection
Raspberry Pi
Setup
Image Classification
Object Detection
Small Language Models (SLM)
Vision-Language Models (VLM)
HDSI Harvard Data Science Initiative
HARVARD Extension School
Google
NSF
Contributors
We express our sincere gratitude to the open-source community of learners, educators, and contributors. Each contribution, whether a chapter section or a single-word correction, has significantly enhanced the quality of this resource. We also acknowledge those who have shared insights, identified issues, and provided valuable feedback behind the scenes.
A comprehensive list of all GitHub contributors, automatically updated with each new contribution, is available below. For those interested in contributing further, please consult our GitHub page for more information.
Vijay Janapa
jasonjabbour
Ikechukwu
Naeem
Marcelo Rovai
Table of contents
Funding Agencies and Companies
Contributors
Edit this page
Report an issue
View source


Motion Classification and Anc
https://harvard-edge.github.io/cs249r_book_dev/contents/labs/seeed/xiao_esp32s3/motion_classification/motion_classification.html
9 Model Optimizations
10 AI Acceleration
11 Benchmarking AI
12 On-Device Learning
13 ML Operations
14 Security & Privacy
15 Responsible AI
16 Sustainable AI
17 Robust AI
18 Generative AI
19 AI for Good
20 Conclusion
LABS
Overview
Getting Started
Nicla Vision
Setup
Image Classification
Object Detection
Keyword Spotting (KWS)
Motion Classification and Anomaly Detection
XIAO ESP32S3
Setup
Image Classification
Object Detection
Keyword Spotting (KWS)
Motion Classification and Anomaly Detection
Raspberry Pi
Setup
Image Classification
Object Detection
Small Language Models (SLM)
Vision-Language Models (VLM)
Shared Labs
KWS Feature Engineering
DSP Spectral Features
REFERENCES
References
Section Quiz
Data Pre-Processing
The raw data type captured by the accelerometer is a "time series" and should be converted to "tabular data". We can do this conversion using a sliding window over the sample data. For example, in the below figure,
Sliding Window: 200 ms
2 second window
300 Raw Features
Raw Data from sensor
Manual Feature Extraction
Spectral Analysis
Features
RMS
SKEW
KURT
FFT
PSD
NN Classifier
Classes
Lift
Terrestrial
Maritime
Idle
We can see 10 seconds of accelerometer data captured with a sample rate (SR) of 50Hz. A 2-second window will capture 300 data points (3 axis x 2 seconds x 50 samples). We will slide this window each 200ms, creating a larger dataset where each instance has 300 raw features.
You should use the best SR for your case, considering Nyquist's theorem, which states that a periodic signal must be sampled at more than twice the signal's highest frequency component.
Data preprocessing is a challenging area for embedded machine learning. Still, Edge Impulse helps overcome this with its digital signal processing (DSP) preprocessing step and, more specifically, the Spectral Features.
On the Studio, this dataset will be the input of a Spectral Analysis block, which is excellent for analyzing repetitive motion, such as data from accelerometers. This block will perform a DSP (Digital Signal Processing), extracting features such as "FFT" or "Wavelets". In the most common case, FFT, the Time Domain Statistical features per axis/channel are:
SocratiQ
Q2: Which theorem helps determine an appropriate sample rate for periodic signals?
A1: Nyquist's theorem
Correct
Nyquist's theorem dictates that a periodic signal should be sampled at more than twice the signal's highest frequency component, which aids in selecting a suitable sample rate.
A2: Shannon's theorem
Shannon's theorem is related to sampling, but does not offer a concrete numerical relationship between sample rate and the maximum frequency.
A3: Whittaker–Kotel'nikov–Shannon theorem
This theorem uses a similar assertion to Nyquist but often concerns itself additionally with optimal signal reconstruction.
Q3: Considering the Spectral Analysis block in Edge Impulse, why is it proper to analyze repetitive motion, such as accelerometer data?
A1: Because repetitive motion generally contains obvious spectral patterns in the frequency domain
Correct
Add Context
type '@' to reference a section...
Information provided here may not always be accurate. Provide feedback

Seeed Studio **XIAO**



TinyML: Why the Future of Machine Learning is Tiny and Bright



Shvetank Prakash, Emil Njor, Colby Banbury, Matthew Stewart, Vijay Janapa Reddi

FEATURES

CUTTING AI DOWN TO SIZE

A $14 chip incorporating tinyML AI models, actual size shown.

https://www.science.org/content/article/what-s-tinyml-global-south-s-alternative-power-hungry-pricey-ai

To learn more …

Online Courses

Harvard School of Engineering and Applied Sciences - CS249r: Tiny Machine Learning
Professional Certificate in Tiny Machine Learning (TinyML) – edX/Harvard
Introduction to Embedded Machine Learning - Coursera/Edge Impulse
Computer Vision with Embedded Machine Learning - Coursera/Edge Impulse
UNIFEI-IESTI01 TinyML: "Machine Learning for Embedding Devices"

Books

"Python for Data Analysis" by Wes McKinney
"Deep Learning with Python" by François Chollet - GitHub Notebooks
"TinyML" by Pete Warden and Daniel Situnayake
"TinyML Cookbook 2nd Edition" by Gian Marco Iodice
"Technical Strategy for AI Engineers, In the Era of Deep Learning" by Andrew Ng
"AI at the Edge" book by Daniel Situnayake and Jenny Plunkett
"XIAO: Big Power, Small Board" by Lei Feng and Marcelo Rovai
"Machine Learning Systems" by Vijay Janapa Reddi

Projects Repository

Edge Impulse Expert Network

On the TinyML4D website, You can find lots of educational materials on TinyML. They are all free and open-source for educational uses – we ask that if you use the material, please cite them! TinyML4D is an initiative to make TinyML education available to everyone globally.

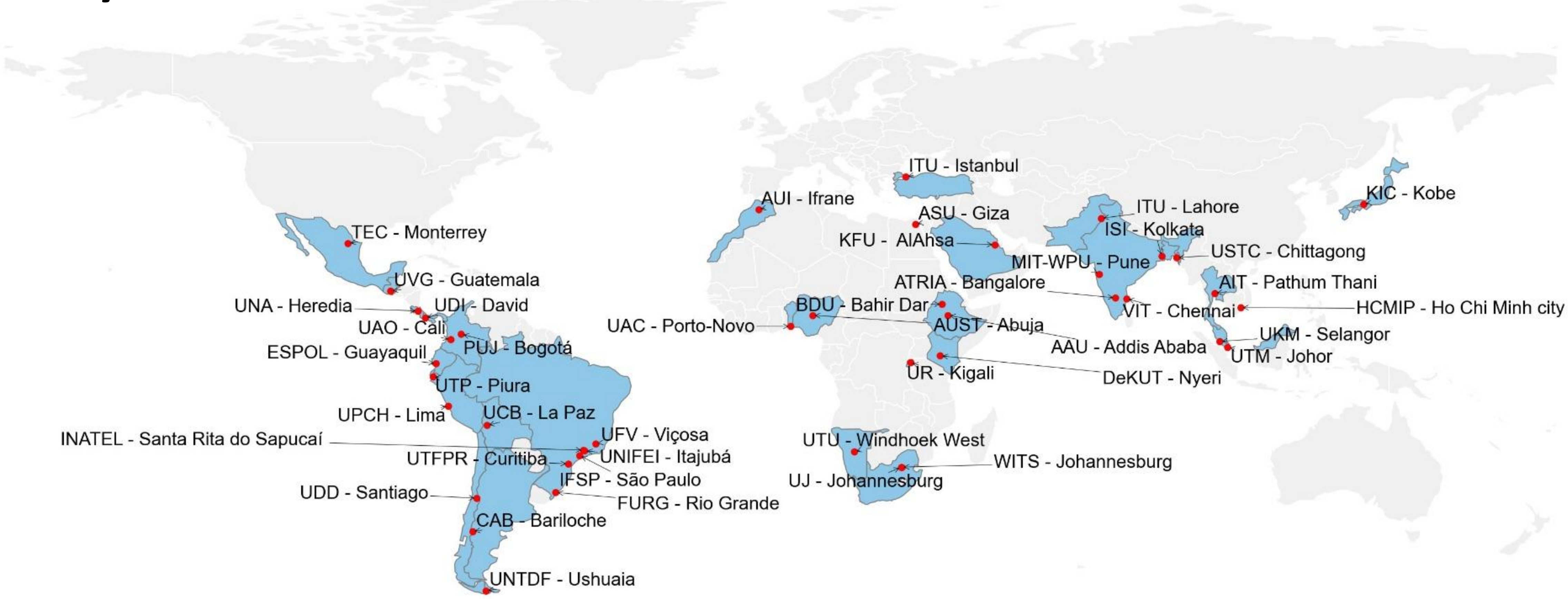
TinyML4D **Show&Tell** Presentations

TinymML4D Academic Network Show and Tell Main Index.

The TinyML4D Academic Network Students should use this form to propose presentations.
https://forms.gle/ic52HZMqVv4pBrkP7 2

The Show and Tell are typically held at 2 pm UTC on the last Thursday of each month and will take place in this Meet link:
https://meet.google.com/rns-yyrx-ggw

TinyML4D **Academic Network**



The IESTI01 course is part of the **TinyML4D**, an initiative to make TinyML education available to everyone globally.

Conclusion

The Future of ML is Tiny and Bright

***Vijay Janapa Reddi**, Ph. D. | Associate Professor |*
John A. Paulson School of Engineering and Applied Sciences | Harvard University |

Questions?

UNIFEI