”Datasheets for Machine Learning Sensors”

Matthew Stewart – Harvard University

July 11, 2023
Thank you, tinyML Strategic Partners, for committing to take tinyML to the next Level, together
Executive Strategic Partners
Advancing AI research to make efficient AI ubiquitous

Power efficiency
- Model design, compression, quantization, algorithms, efficient hardware, software tool

Personalization
- Continuous learning, contextual, always-on, privacy-preserved, distributed learning

Efficient learning
- Robust learning through minimal data, unsupervised learning, on-device learning

Perception
- Object detection, speech recognition, contextual fusion

Reasoning
- Scene understanding, language understanding, behavior prediction

Action
- Reinforcement learning for decision making

A platform to scale AI across the industry
Accelerate Your Edge Compute
SYNTIANT
Making Edge AI A Reality
www.syntiant.com
Platinum Strategic Partners
Renesas is enabling the next generation of AI-powered solutions that will revolutionize every industry sector.
DEPLOY VISION AI
AT THE EDGE AT SCALE
Gold Strategic Partners
Where what if becomes what is.

Witness potential made possible at analog.com.
Easily deploy your tinyML solutions with Arduino Pro

arduino.cc/pro
Build the Future of tinyML on Arm
The Leading Development Platform for Edge ML

edgeimpulse.com
Driving decarbonization and digitalization. Together.

Infineon serving all target markets as Leader in Power Systems and IoT

www.infineon.com
NEUROMORPHIC INTELLIGENCE FOR THE SENSOR-EDGE

www.innatera.com
The Right Edge AI Tools Can Make or Break Your Next Smart IoT Product

Analytics Toolkit Suite

AutoML

Data Collection

Tool & Validation

Data Labeling

Code Generation

Model Building

Version Control and Model Management

sensiml.com/tinyML
STMicroelectronics provides extensive solutions to make tiny Machine Learning easy
ENGINEERING
EXCEPTIONAL
EXPERIENCES

We engineer exceptional experiences for consumers in the home, at work, in the car, or on the go.

www.synaptics.com
Silver Strategic Partners
Join Growing tinyML Communities:

tinyML - Enabling ultra-low Power ML at the Edge

The tinyML Community
https://www.linkedin.com/groups/13694488/

15.6k members in 49 Groups in 41 Countries

4k members & 12.4k followers
Subscribe to tinyML YouTube Channel for updates and notifications (including this video)
www.youtube.com/tinyML
Reminders

Slides & Videos will be posted tomorrow

tinyml.org/forums  youtube.com/tinyml

Please use the Q&A window for your questions
Matthew Stewart

Matthew Stewart is a postdoctoral researcher in the Edge Computing Lab at Harvard University. He holds a Ph.D. and MSc in Engineering Sciences and Data Science from Harvard University, and an integrated BEng/MEng in Mechanical Engineering from Imperial College London and the National University of Singapore. Matthew's research work is highly interdisciplinary, encompassing embedded machine learning, autonomous vehicles, benchmarking tools for reinforcement learning and robotics, sustainable computing, and machine learning sensors. Matthew is also a part-time blogger for Towards Data Science, a co-creator of the HarvardX tinyML courses, and a research coordinator at MLCommons.
Datasheets for Machine Learning Sensors

Acknowledgements: B. Brown, Y. Omri, J. Santos, J. MacArthur, B. Plancher, S. Prakash, N. Jeffries, V. J. Reddi, P. Warden & the Useful Sensors Team

Matthew Stewart, Ph. D. | Postdoctoral Researcher |
John A. Paulson School of Engineering and Applied Sciences | Harvard University |
Web: https://mpstewart.io
Miniaturization of Compute

Data Centers -> Edge Devices -> Internet-of-Things -> TinyML
5 Quintillion bytes of data produced every day by IoT

<1% of unstructured data is analyzed or used at all

Cisco, *Internet of Things (IoT) Data Continues to Explode Exponentially. Who Is Using That Data and How?*, Feb 5, 2018
The “Classic” TinyML Paradigm

Physical Sensor → Processor → Cloud

Sensor 1.0

Raw data

Optional connection to cloud-based system
How to Stop Your Smart TV From Spying on You

A voice command starts your TV, identification, and viewing data interconnectivity has privacy implications, smart TV spying and how the

Google Calls Hidden Microphone in Its Nest Home Security Devices an 'Error'

The company says it was an oversight, but it does little to stem paranoia.

FBI warns about snoopy smart TVs spying on you

An FBI branch office warns smart TV users that they can be gateways for hackers to come into your home. Meanwhile, the smart TV OEMs are already spying on you.

Support the Guardian
Make a year-end gift today

The Observer
Smart homes
How to stop your smart home spying on you

Everything in your smart home, from the lightbulbs to the thermostat, could be recording you or collecting data about you.
How do we architect future Tiny Machine Learning (tinyML) sensors efficiently, effectively and robustly into the embedded ecosystem?
Machine Learning Sensors
An ML sensor is a self-contained system that utilizes on-device machine learning to extract useful information by observing some complex set of phenomena in the physical world and reports it through a simple interface to a wider system.
Machine Learning Sensors

Physical Sensor → Processor → Cloud

Raw data

Optional connection to cloud-based system

Sensor 1.0
Machine Learning Sensors

ML Sensor → Processor → Cloud

- High-level features/labels
- Optional connection to cloud-based system
ML Sensors - Guiding Set of Principles

1. We need to **raise the level of abstraction** to enable ease of use for scalable deployment of ML sensors; not everyone should be required to be a developer or an engineer to leverage ML sensors into their ecosystem.

2. The ML sensor’s **design should be inherently data-centric** and defined by its input-output behavior instead of exposing the underlying hardware and software mechanisms that support ML model execution.

1. An ML sensor’s **implementation must be clean and complexity-free**. Features such as reusability, software updates, and networking must be thought through to ensure data privacy and secure execution.

2. ML sensors **must be transparent, indicating in a publicly and freely accessible ML sensor datasheet** all the relevant information to supplement the traditional information available for hardware sensors.

3. We as a community should aim to **foster an open ML sensors ecosystem by maximizing data, model, and hardware transparency** where possible, without necessarily relinquishing any claim to intellectual property.
ML Sensor Principles — Abstraction

1. We need to **raise the level of abstraction** to enable ease of use for scalable deployment of ML sensors; not everyone should be required to be a developer or an engineer to leverage ML sensors into their ecosystem.

2. The ML sensor’s **design should be inherently data-centric** and defined by its input-output behavior instead of exposing the underlying hardware and software mechanisms that support ML model execution.

1. An ML sensor’s **implementation must be clean and complexity-free**. Features such as reusability, software updates, and networking must be thought through to ensure data privacy and secure execution.

2. ML sensors **must be transparent**, indicating in a publicly and freely accessible ML sensor datasheet all the relevant information to supplement the traditional information available for hardware sensors.

3. We as a community should aim to **foster an open ML sensors ecosystem by maximizing data, model, and hardware transparency** where possible, without necessarily relinquishing any claim to intellectual property.
ML Sensor Principles — Abstraction

1. We need to raise the level of abstraction to enable ease of use for scalable deployment of ML sensors; not everyone should be required to be a developer or an engineer to leverage ML sensors into their ecosystem.

2. The ML sensor’s design should be inherently data-centric and defined by its input-output behavior instead of exposing the underlying hardware and software mechanisms that support ML model execution.

1. An ML sensor’s implementation must be clean and complexity-free. Features such as reusability, software updates, and networking must be thought through to ensure data privacy and secure execution.

2. ML sensors must be transparent, indicating in a publicly and freely accessible ML sensor datasheet all the relevant information to supplement the traditional information available for hardware sensors.

3. We as a community should aim to foster an open ML sensors ecosystem by maximizing data, model, and hardware transparency where possible, without necessarily relinquishing any claim to intellectual property.
<table>
<thead>
<tr>
<th>Part Number</th>
<th>Description</th>
<th>Texas Instruments</th>
<th>Price</th>
<th>Package</th>
<th>Secondary Device(s)</th>
<th>Voltage</th>
<th>Resolution</th>
<th>Features</th>
<th>Accuracy</th>
<th>Temperature Range</th>
<th>Test Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>TMP2342D22X7</td>
<td>SENSOR TEMPERATURE Analyzer</td>
<td>Texas Instruments</td>
<td>1.632</td>
<td>In stock</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1.8V to 5.5V</td>
<td>50°C to 125°C</td>
</tr>
<tr>
<td>TMP2342C22X7</td>
<td>SENSOR TEMPERATURE Analyzer</td>
<td>Texas Instruments</td>
<td>1.678</td>
<td>In stock</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1.8V to 5.5V</td>
<td>50°C to 125°C</td>
</tr>
<tr>
<td>TMP2343C22X7</td>
<td>SENSOR TEMPERATURE Analyzer</td>
<td>Texas Instruments</td>
<td>2.07</td>
<td>In stock</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1.8V to 5.5V</td>
<td>50°C to 125°C</td>
</tr>
<tr>
<td>TMP401L200F01</td>
<td>SENSOR TEMPERATURE Analyzer</td>
<td>Texas Instruments</td>
<td>340.00</td>
<td>In stock</td>
<td>Automatic, Analog-DC-003</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>TMP2342D22X7</td>
<td>SENSOR TEMPERATURE Analyzer</td>
<td>Texas Instruments</td>
<td>596.00</td>
<td>In stock</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1.8V to 5.5V</td>
<td>50°C to 125°C</td>
</tr>
<tr>
<td>TMP401L200F01</td>
<td>SENSOR TEMPERATURE Analyzer</td>
<td>Texas Instruments</td>
<td>227.00</td>
<td>In stock</td>
<td>Automatic, Analog-DC-003</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>TMP401L200F01</td>
<td>SENSOR TEMPERATURE Analyzer</td>
<td>Texas Instruments</td>
<td>227.00</td>
<td>In stock</td>
<td>Automatic, Analog-DC-003</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>TMP401L200F01</td>
<td>SENSOR TEMPERATURE Analyzer</td>
<td>Texas Instruments</td>
<td>227.00</td>
<td>In stock</td>
<td>Automatic, Analog-DC-003</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>TMP152P</td>
<td>ANALOG TEMPERATURE SENSOR</td>
<td>Analog Devices Inc</td>
<td>1.025</td>
<td>In stock</td>
<td>Back</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>MAX66500M8F7-T</td>
<td>DIGITAL TEMPERATURE SENSOR</td>
<td>Analog Devices Inc</td>
<td>5.996</td>
<td>In stock</td>
<td>Back</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>TMP52F7-2S</td>
<td>ANALOG TEMPERATURE SENSOR</td>
<td>Analog Devices Inc</td>
<td>28.36</td>
<td>In stock</td>
<td>Back</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>MAX66500M8F7-T</td>
<td>DIGITAL TEMPERATURE SENSOR</td>
<td>Analog Devices Inc</td>
<td>6.682</td>
<td>In stock</td>
<td>Back</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>AD2102W-RED</td>
<td>ANALOG TEMPERATURE SENSOR</td>
<td>Analog Devices Inc</td>
<td>3.990</td>
<td>In stock</td>
<td>Back</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
ML Sensor Principles — Abstraction

1. We need to **raise the level of abstraction** to enable ease of use for scalable deployment of ML sensors; not everyone should be required to be a developer or an engineer to leverage ML sensors into their ecosystem.
1. We need to **raise the level of abstraction** to enable ease of use for scalable deployment of ML sensors; not everyone should be required to be a developer or an engineer to leverage ML sensors into their ecosystem.

2. The ML sensor’s **design should be inherently data-centric** and defined by its input-output behavior instead of exposing the underlying hardware and software mechanisms that support ML model execution.

1. An ML sensor’s **implementation must be clean and complexity-free**. Features such as reusability, software updates, and networking must be thought through to ensure data privacy and secure execution.

2. ML sensors **must be transparent, indicating in a publicly and freely accessible ML sensor datasheet** all the relevant information to supplement the traditional information available for hardware sensors.

3. We as a community should aim to **foster an open ML sensors ecosystem by maximizing data, model, and hardware transparency** where possible, without necessarily relinquishing any claim to intellectual property.
1. We need to raise the level of abstraction to enable ease of use for scalable deployment of ML sensors;

2. The ML sensor’s **design should be inherently data-centric** and defined by its input-output behavior instead of exposing the underlying hardware and software mechanisms that support ML model execution.

ML Sensor Principles — Data-centric

---

Sensor 1.0

Sensor 2.0
ML Sensor Principles — Simplicity

1. We need to raise the level of abstraction to enable ease of use for scalable deployment of ML sensors; not everyone should be required to be a developer or an engineer to leverage ML sensors into their ecosystem.

2. The ML sensor’s design should be inherently data-centric and defined by its input-output behavior instead of exposing the underlying hardware and software mechanisms that support ML model execution.

1. An ML sensor’s implementation must be clean and complexity-free. Features such as reusability, software updates, and networking must be thought through to ensure data privacy and secure execution.

2. ML sensors must be transparent, indicating in a publicly and freely accessible ML sensor datasheet all the relevant information to supplement the traditional information available for hardware sensors.

3. We as a community should aim to foster an open ML sensors ecosystem by maximizing data, model, and hardware transparency where possible, without necessarily relinquishing any claim to intellectual property.
ML Sensor Principles — Simplicity

1. We need to raise the level of abstraction to enable ease of use for scalable deployment of ML sensors; not everyone should be required to be a developer or an engineer to leverage ML sensors into their ecosystem.

2. The ML sensor’s design should be inherently data-centric and defined by its input-output behavior instead of exposing the underlying hardware and software mechanisms that support ML model execution.

1. An ML sensor’s implementation must be clean and complexity-free. Features such as reusability, software updates, and networking must be thought through to ensure data privacy and secure execution.

We need to define or rely on standard interfaces and mechanisms for communication with sensors.

Source: https://github.com/usefulsensors/person_sensor_docs
ML Sensor Principles — Simplicity

1. We need to **raise the level of abstraction** to enable ease of use for scalable deployment of ML sensors; not everyone should be required to be a developer or an engineer to leverage ML sensors into their ecosystem.

2. The ML sensor’s **design should be inherently data-centric** and defined by its input-output behavior instead of exposing the underlying hardware and software mechanisms that support ML model execution.

1. An ML sensor’s **implementation must be clean and complexity-free**. Features such as reusability, software updates, and networking must be thought through to ensure data privacy and secure execution.

We need to define data formats to enable interoperability and exchange of ML sensors across manufacturers.

Source: https://github.com/usefulsensors/person_sensor_docs
ML Sensor Principles — Simplicity

1. We need to raise the level of abstraction to enable ease of use for scalable deployment of ML sensors; not everyone should be required to be a developer or an engineer to leverage ML sensors into their ecosystem.

2. The ML sensor’s design should be inherently data-centric and defined by its input-output behavior instead of exposing the underlying hardware and software mechanisms that support ML model execution.

1. An ML sensor’s implementation must be clean and complexity-free. Features such as reusability, software updates, and networking must be thought through to ensure data privacy and secure execution.

Source: https://github.com/usefulsensors/person_sensor_docs
ML Sensor Principles — Documentation

1. We need to raise the level of abstraction to enable ease of use for scalable deployment of ML sensors; not everyone should be required to be a developer or an engineer to leverage ML sensors into their ecosystem.

2. The ML sensor’s design should be inherently data-centric and defined by its input-output behavior instead of exposing the underlying hardware and software mechanisms that support ML model execution.

1. An ML sensor’s implementation must be clean and complexity-free. Features such as reusability, software updates, and networking must be thought through to ensure data privacy and secure execution.

2. ML sensors must be transparent, indicating in a publicly and freely accessible ML sensor datasheet all the relevant information to supplement the traditional information available for hardware sensors.

3. We as a community should aim to foster an open ML sensors ecosystem by maximizing data, model, and hardware transparency where possible, without necessarily relinquishing any claim to intellectual property.
1. We need to raise the level of abstraction to enable ease of use for scalable deployment of ML sensors; not everyone should be required to be a developer or an engineer to leverage ML sensors into their ecosystem.

2. The ML sensor’s design should be inherently data-centric and defined by its input-output behavior instead of exposing the underlying hardware and software mechanisms that support ML model execution.

1. An ML sensor’s implementation must be clean and complexity-free. Features such as reusability, software updates, and networking must be thought through to ensure data privacy and secure execution.

2. ML sensors must be transparent, indicating in a publicly and freely accessible ML sensor datasheet all the relevant information to supplement the traditional information available for hardware sensors.

3. We as a community should aim to foster an open ML sensors ecosystem by maximizing data, model, and hardware transparency where possible, without necessarily relinquishing any claim to intellectual property.
2. ML sensors **must be transparent, indicating in a publicly and freely accessible ML sensor datasheet** all the relevant information to supplement the traditional information available for hardware sensors.

3. We as a community should aim to **foster an open ML sensors ecosystem by maximizing data, model, and hardware transparency** where possible, without necessarily relinquishing any claim to intellectual property.
Datasheet Template

- Description, Features and Use case
- Compliance
- Environmental Impact
- Privacy and Security
- Diagram and Form Factor
- Data Nutrition Label
- Hardware Characteristics
- Communication Specification and Pinout
- Model Characteristics
- End-to-End Performance Analysis

End-to-End

Model Characteristics

Communication Specification and Pinout

Hardware Characteristics

Data Nutrition Label

Privacy and Security

Compliance

Environmental Impact

Description, Features and Use case

End-to-End
Model Characteristics

- Provides a concise, holistic picture of the performance characteristics of a machine learning model
- For our sensor, this is a binary classification person detection model which processes raw image data (Mitchell et al., 2019)
Data Nutrition Label

At a Glance

- Offers a succinct, comprehensive snapshot of dataset attributes used for model training
- For our sensor, this focuses on attributes of the visual wake words dataset derived from MS-COCO.

(Holland et al., 2020; Chmielinski et al., 2022)
Privacy and Security Label

<table>
<thead>
<tr>
<th>Security &amp; Privacy Overview</th>
<th>Privacy and Security Details</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Harvard University</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Security functions:</td>
<td></td>
</tr>
<tr>
<td>Sensor data collection:</td>
<td></td>
</tr>
<tr>
<td>Sensor type:</td>
<td></td>
</tr>
<tr>
<td>Data retained in the device</td>
<td></td>
</tr>
<tr>
<td>Data shared with</td>
<td></td>
</tr>
<tr>
<td>Other collected data</td>
<td></td>
</tr>
<tr>
<td>Privacy policy</td>
<td></td>
</tr>
<tr>
<td>Detailed Security &amp; Privacy Label</td>
<td>Not disclosed</td>
</tr>
</tbody>
</table>

- **Privacy.** Provides clear and transparent information regarding data capture, usage, and storage for each data modality.
- **Security.** Safety protocols and security mechanisms associated with the device are outlined.

(Emami-Naeini et al., 2021)
Performance Analysis

- Provides an indication of **demographic biases** as well as performance changes under **varying environmental conditions**.

- **Experimental study conducted** under different lighting and distances to assess sensor performance in **real-world conditions**.
Experimental Study Details

- Data collected from 39 participants
- Each participant tested at three different distances and three lighting levels (nine total scenarios)
- Ten measurements (~10 s) taken at each location and averaged
- Six sensors utilized: three open-source ML sensors and three commercially available (Useful Sensors)
- Neutral background environment with no ambient light exposure; 32 homogeneously distributed overhead lights
ML sensors should be tested by 3rd party certification agencies or bodies that specialize in AI/ML technologies to:

1. Ensure adherence to industry standards and regulations
2. Support data integrity and accuracy
3. Foster trust in the product’s performance and reliability
# Hardware Details

## Diagram and Form Factor

![Diagram and Form Factor Image]

## Communication Specification

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>I2C/Qwiic mode</td>
<td>Conforms with SparkFun Qwiic electrical/mechanical specifications. <a href="https://www.sparkfun.com/qwiic">https://www.sparkfun.com/qwiic</a></td>
</tr>
<tr>
<td>Max cable length</td>
<td>1 m</td>
</tr>
<tr>
<td>Max data rate</td>
<td>100 kb/s</td>
</tr>
<tr>
<td>Module Orientation</td>
<td>Red arrow on sticker points up.</td>
</tr>
<tr>
<td>GPIO mode</td>
<td>SCL/SDA lines can be customized to make programmable flag lines (I_{out max} = 12 mA)</td>
</tr>
<tr>
<td>Diagnostic LED</td>
<td>Default behavior of green LED on board: illuminates for one second on power-up, then illuminates when person detected.</td>
</tr>
<tr>
<td>Data Transfer and Format</td>
<td>Single byte: number from 0-255 representing confidence score</td>
</tr>
<tr>
<td>I2C Address</td>
<td>0x22</td>
</tr>
</tbody>
</table>
- Need to consider **environmentally-relevant metrics** such as carbon emissions, water usage, and eutrophication potential.

- Carbon emissions have two relevant sources: **operational energy consumption** and **hardware manufacturing and infrastructure**.

- Majority of emissions from our person detection sensor are associated with the **embodied footprint**. Data estimated using our **TinyML sustainability calculator**.

(Gupta et al., 2022; Prakash et al., 2022)
Datasheet Overview
ML Sensors - Guiding Set of Principles

1. We need to **raise the level of abstraction** to enable ease of use for scalable deployment of ML sensors; not everyone should be required to be a developer or an engineer to leverage ML sensors into their ecosystem.

2. The ML sensor’s **design should be inherently data-centric** and defined by its input-output behavior instead of exposing the underlying hardware and software mechanisms that support ML model execution.

1. An ML sensor’s **implementation must be clean and complexity-free**. Features such as reusability, software updates, and networking must be thought through to ensure data privacy and secure execution.

2. ML sensors **must be transparent, indicating in a publicly and freely accessible ML sensor datasheet** all the relevant information to supplement the traditional information available for hardware sensors.

3. We as a community should aim to **foster an open ML sensors ecosystem by maximizing data, model, and hardware transparency** where possible, without necessarily relinquishing any claim to intellectual property.
Machine Learning Sensors

An ML sensor is a self-contained system that utilizes on-device machine learning to extract useful information by observing some complex set of phenomena in the physical world and reports it through a simple interface to a wider system.

Machine learning sensors represent a paradigm shift for the future of embedded machine learning applications. Current instantiations of embedded ML suffer from complex integration, lack of modularity, and privacy and security concerns from data movement. ML sensors provide a more data-centric paradigm for embedding sensor intelligence on edge devices to combat these challenges.

Our vision for “sensor 2.0” entails segregating sensor input data and ML processing from the wider system at the hardware level and providing a thin interface that mimics traditional sensors in functionality. This separation leads to a modular and easy-to-use ML sensor device. ML sensors increase privacy and accuracy while making it easier for system builders to integrate ML into their products as a simple component.

To learn more about our approach, check out our ML sensor whitepaper, as well as our recent work on datasheets for machine learning sensors.

Challenges

Interface
What universal interface is needed for ML

Standards
What standards need to be in place for ML

Ethics
What ethical considerations are needed for
Recap of ML Sensors

1. We need to **raise the level of abstraction** to enable ease of use for scalable deployment of ML sensors; not everyone should be required to be a developer or an engineer to leverage ML sensors into their ecosystem.

2. The ML sensor’s **design should be inherently data-centric** and defined by its input-output behavior instead of exposing the underlying hardware and software mechanisms that support ML model execution.

1. An ML sensor’s **implementation must be clean and complexity-free**. Features such as reusability, software updates, and networking must be thought through to ensure data privacy and secure execution.

2. ML sensors **must be transparent, indicating in a publicly and freely accessible ML sensor datasheet** all the relevant information to supplement the traditional information available for hardware sensors.

3. We as a community should aim to **foster an open ML sensors ecosystem by maximizing data, model, and hardware transparency** where possible, without necessarily relinquishing any claim to intellectual property.
Call to Action

Radcliffe exploratory seminar to determine:

What ethical considerations are necessary when developing ML sensors?

What compliance standards must be met by ML sensor developer and manufacturers?

How should ML sensors interface with existing systems?
Machine learning sensors represent a paradigm shift for the future of embedded machine learning applications. Current incarnations of embedded machine learning (ML) suffer from complex integration, lack of modularity, and privacy and security concerns from data movement. This paper proposes a more data-centric paradigm for embedding sensor intelligence on edge devices to combat these challenges. Our vision for “sensor 2.0” entails segregating sensor input data and ML processing from the wider system at the hardware level and providing a thin interface that mimics traditional sensors in functionality. This separation leads to a modular and easy-to-use ML sensor device. We discuss challenges presented by the standard approach of building ML processing into the software stack of the existing microprocessor on an embedded system and how the modularity of ML sensors alleviates these problems. ML sensors require privacy and accuracy while making it easier for system builders to integrate ML into their products as a simple component. We provide examples of prospective ML sensors as an illustrative database as a demonstration and hope that this will build a dialogue to progress us towards sensor 2.0.

Datasheets for Machine Learning Sensors

Figure 1. The Sensor 1.0 paradigm tightly couples the ML model with the application processor and logic, making it difficult to provide hard guarantees about the ML sensor’s ultimate behavior.

Figure 2. Our proposed Sensor 2.0 paradigm. The ML model is tightly coupled with the physical sensor component of the application processor, and comes with an ML sensor descriptor that makes it Behavior transparent to the system integrator and developer.

Abstract

Machine learning (ML) sensors offer a new paradigm for sensing that enables intelligence of the edge while empowering end-users with greater control of their data. As these ML sensors play a crucial role in the development of intelligent devices, clear documentation of their specifications, functionalities, and limitations is needed. In this paper, we introduce a standard datasheet template for ML sensors and discuss various essential components making up the system’s hardware, ML model, and data transfer, highlighting the need for end-to-end performance metrics, and environmental impact. We propose a datasheet template that could be integrated into software and hardware development to aid in the standardization of these kinds of sensors.

1 Introduction

The recent emergence of tiny machine learning (TinyML), a branch of ML dedicated to ultra-low power devices, has opened the door to a myriad of new possibilities for intelligent sensing at the edge by leveraging embedded systems [1, 2]. TinyML enables resource-constrained devices to perform complex computations with low latency and minimal energy consumption, making it particularly suitable for applications such as the Internet of Things (IoT), wearables, and smart sensors. However, integrating TinyML models into physical sensor systems can be complex, often requiring a deep understanding of ML algorithms and embedded systems. This knowledge barrier can hinder the widespread adoption of AI-driven intelligence. To address these challenges, the “ML sensor” project has been proposed as an innovative solution that tightly couples the TinyML model with the physical sensor, effectively offloading the computational burden from the application processor [3]. This ML sensor architecture introduces useful layers of abstraction both at the hardware level and at the level of the full integrated device, creating a fully self-contained intelligent sensor module.

ML sensors, however, also present a new challenge: the lack of transparency (4, 5). Unlike traditional sensors that come with datasheets providing hardware and operating characteristics, ML sensors lack such documentation. This absence hampers developers’ ability to assess sensor suitability and independently evaluative performance. To address this gap, ML sensors require a datasheet that not only includes traditional sensor specifications but also captures ML model characteristics, dataset details, and other important considerations such as environmental impact and end-to-end performance. With such a datasheet, users can easily determine whether an ML sensor is suitable for their application.
Copyright Notice

This multimedia file is copyright © 2023 by tinyML Foundation. All rights reserved. It may not be duplicated or distributed in any form without prior written approval.

tinyML® is a registered trademark of the tinyML Foundation.

www.tinymce.org
Copyright Notice

This presentation in this publication was presented as a tinyML® Talks webcast. The content reflects the opinion of the author(s) and their respective companies. The inclusion of presentations in this publication does not constitute an endorsement by tinyML Foundation or the sponsors.

There is no copyright protection claimed by this publication. However, each presentation is the work of the authors and their respective companies and may contain copyrighted material. As such, it is strongly encouraged that any use reflect proper acknowledgement to the appropriate source. Any questions regarding the use of any materials presented should be directed to the author(s) or their companies.

tinyML is a registered trademark of the tinyML Foundation.

www.tinyml.org