How sensors with machine learning core bring power-efficient AI applications to the edge

Outline

Artificial intelligence and machine learning

Adding intelligence to make sensorization sustainable

Smart sensors process the data at the edge

↓

Only a reduced, meaningful set of data is transferred to the network

↓

The required cloud infrastructure is limited

↓

Edge processing makes the sensorization of the world sustainable

Smart sensors are becoming even smarter

- Edge processing
- Ultralow power consumption
- Advanced security
- Small form factor

life.augmented

What is AI?

The evolution of AI

Artificial intelligence

Early artificial intelligence stirs excitement

Machine learning

Machine learning begins to flourish

Deep learning

Deep learning breakthroughs drive AI boom

1950’ 1960’ 1970’ 1980’ 1990’ 2000’ 2010’ 2020’

Any technique that enables a computer to mimic **human behavior**

Subset of AI. Algorithms and methodologies that improve over time through **learning from data**

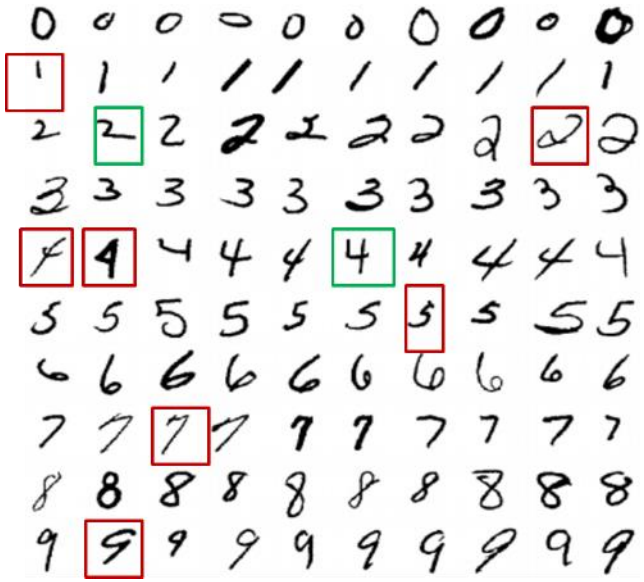
Subset of ML. Learning algorithms that derive meaning out of data, by using a hierarchy of multiple layers **that mimic the neural networks of the human brain**

Machine learning: why do we need it?

When a complex task or problem involves a large amount of data and lots of variables, but no existing formula or equation can solve it

An example of difficult program

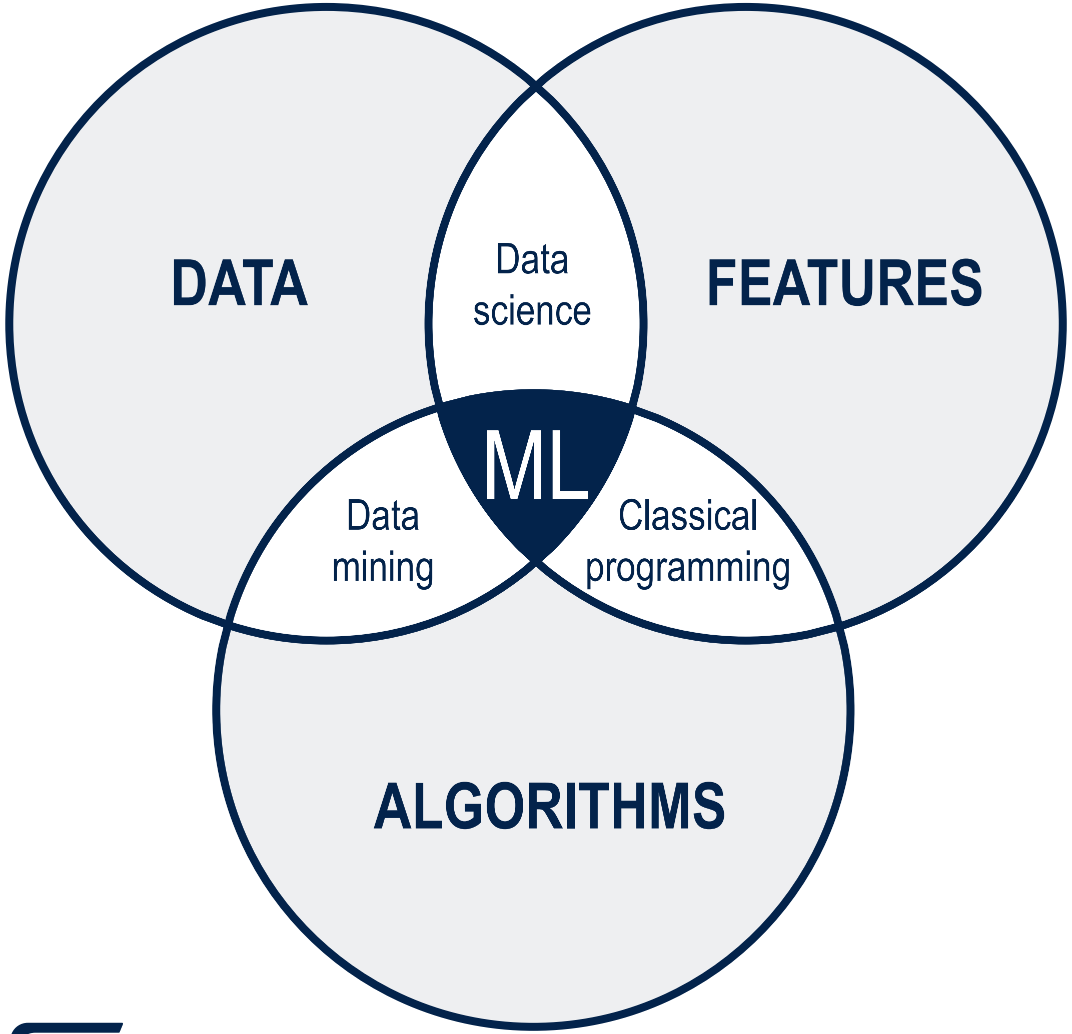
- How to recognize the handwritten digits?
- Very difficult to define the rules!
- What makes all these numbers to be identifiable?
- Is there a pattern?
- What is it that makes a 2 to be identified as a 2?



Some examples from MNIST database

(Mixed standard institute for standard and technology)

Three components of machine learning



The only goal of machine learning is to predict results based on incoming data

DATA: the more diverse the data, the better the result. It's extremely tough to collect a good collection of data (usually called a dataset)

FEATURES: also known as parameters or variables.

ALGORITHMS: The most obvious part. The method you choose affects the precision, performance, and size of the final model.

Standard vs machine learning algorithm approach

Standard programming
"A priori" approach

Design algorithm specific for the given problem

Machine learning
"Empirical" approach

Log of input data to the system + Desired output from the system

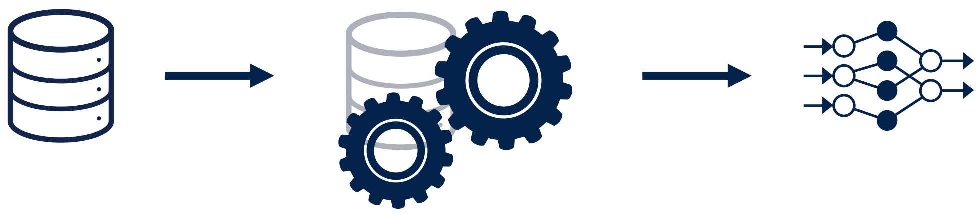
General ML model trained for the specific problem

Using AI algorithms implies two phases

Training

AI models are produced using historical **datasets** and a training engine / framework

Dataset | Training | Model

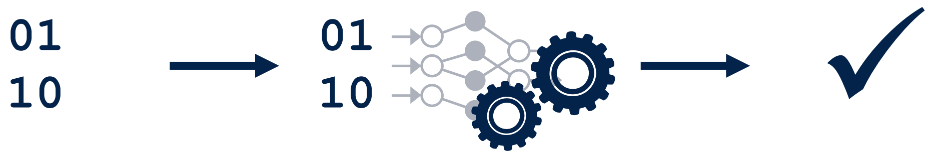


Inference

Model deployment

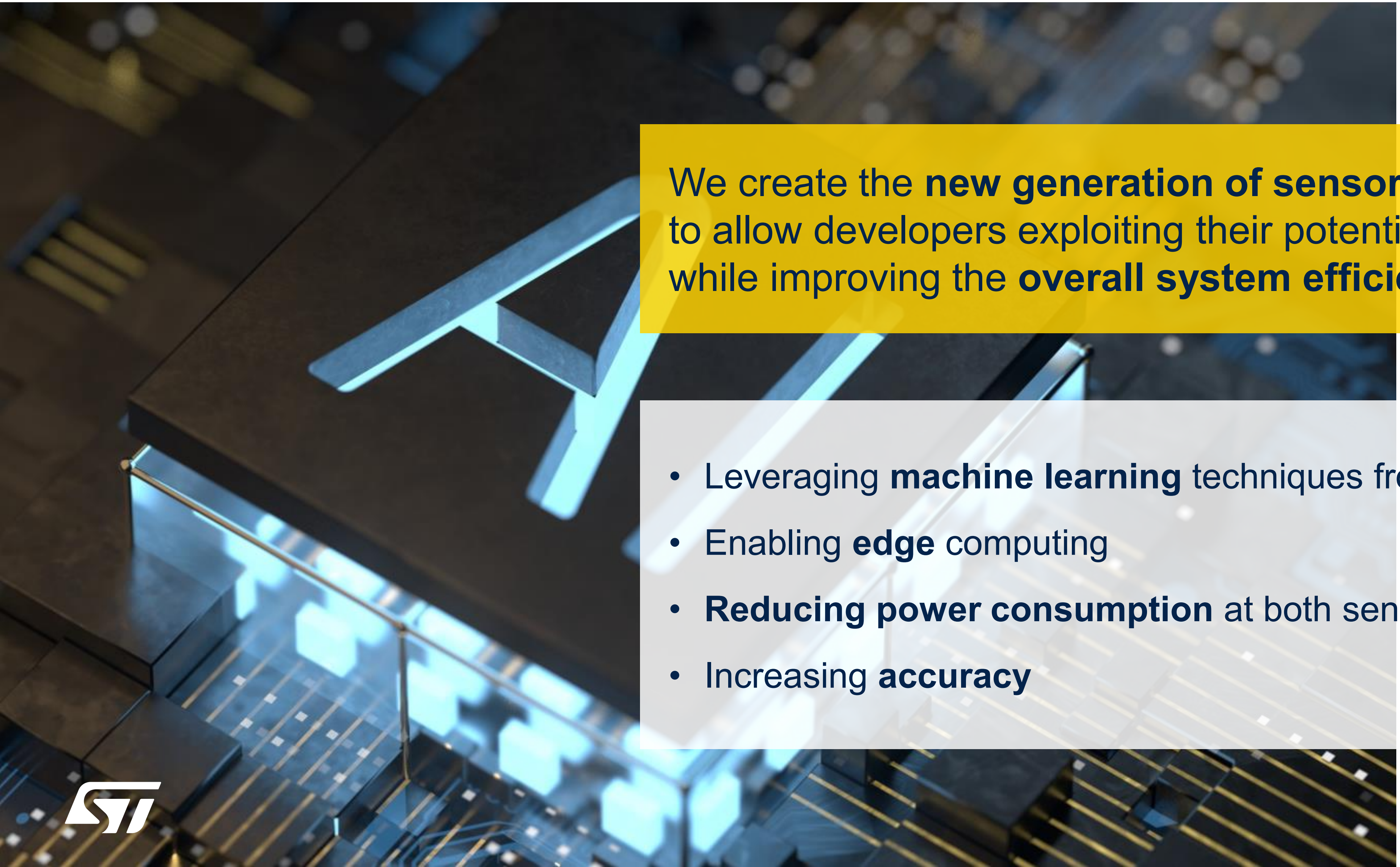
Inference is the process of using a trained machine learning model to make predictions or decisions based on new, unseen data.

Input data | Model & engine | Result



Products with machine learning core

Smart sensors



We create the **new generation of sensors** to allow developers exploiting their potential while improving the **overall system efficiency**

- Leveraging **machine learning** techniques from the world of A.I.
- Enabling **edge** computing
- **Reducing power consumption** at both sensor and **system** levels
- Increasing **accuracy**

Decision-making capabilities closer to data source: sensors with machine learning core (MLC)

MLC is a **hard-coded engine** inside the sensor based on **decision tree logic:**

- **optimal system energy efficiency (few µA)**
- **granting maximum data privacy**

Sensor data	Computation block	Decision tree
Accelerometer	Filters	Metaclassifier
Gyroscope	Predefined features	Results

Machine learning core (MLC) What it is

MLC is an in-sensor classification engine based on decision tree logic

Motion sensing

External sensors

Sensor with machine learning core

Sensor data	**Computation block**	**Decision tree**
Accelerometer	Filters	Meta-classifier
Gyroscope	Digital Pre-defined Features	Results

Main processor

MLC is able to **increase accuracy** with a **better context detectability**, **offloading the main processor** while the built-in sensors identify motion data

Machine learning core (MLC)
The blocks

Sensor data

A wide set of inputs is available:

- **Accelerometer**
  → $[a_x\ a_y\ a_z\ ], [a_v], [a^2{}_v]$
- **Gyroscope**
  → $[g_x\ g_y\ g_z\ ], [g_v]\ , [g^2{}_v]$
- **External sensor** (e.g. magnetometer)
  → $[m_x\ m_y\ m_z\ ], [m_v], [m^2{}_v]$
- **Magnitude**
  → $V = \sqrt{X^2 + Y^2 + Z^2}$

Computation block

- Sensor data can be filtered with a 2nd order **IIR filter**
- Features are statistical parameters calculated from:
  - Input Data
  - Filtered Input Data
- Examples of features are: Mean, Variance, Energy, Peak to Peak, …

Decision tree

- The **decision tree** is a predictive model built from training data. Outputs of the computation block are inputs for the decision tree
- Each **node** is characterized by one «if-then-else» condition. Some examples of conditions:
  - Mean on Acc_X < 0.5 g
  - Variance on Gyro_Z < 200 dps
- A decision tree can either generate a result at every sample or filter the results with a **meta-classifier**, to have a more robust output

Machine learning core
Features & performance

	Generation 1	Generation 2	New features User advantages
Performance			
MLC rates	ODR = 13, 26, 53, 104 Hz	ODR = 15, 30, 60, 120, **240 Hz**	MLC rate increase to enable addressing new apps (spike detection)
MLC with external sensor acquisition	2 bytes	2 bytes, **3 bytes**	Full processing of high-resolution sensor (e.g. ST pressure sensor)
System optimization			
MLC features / filters exportable	n.a.	**Yes (available in FIFO)**	AI data directly stored in FIFO and exportable
MLC recursive features	n.a.	**Yes**	Short-time events perfectly captured by recursive sliding windows
Improved meta classifier	n.a.	**Yes**	
MLC execution before FSM	n.a.	**Yes**	

Machine learning solutions in accelerometers and IMUs

Automotive

Inertial measurement units (IMUs)

- ASM330LHHX / ASM330LHHXG1
- ASM330LHB / ASM330LHBG1

MLC Gen. 2.0

Industrial

Inertial measurement units (IMUs)

- ISM330BX
- ISM330DHCX

Accelerometers

- IIS2DULPX
- IIS2ICLX

MLC Gen. 2.0

Personal electronics

Inertial measurement units (IMUs)

- LSM6DSV320X
- LSM6DSV80X
- LSM6DSV16BX
- LSM6DSV16X / LSM6DSV32X

Accelerometers

- LIS2DUXS12
- LIS2DUX12

MLC Gen. 2.0

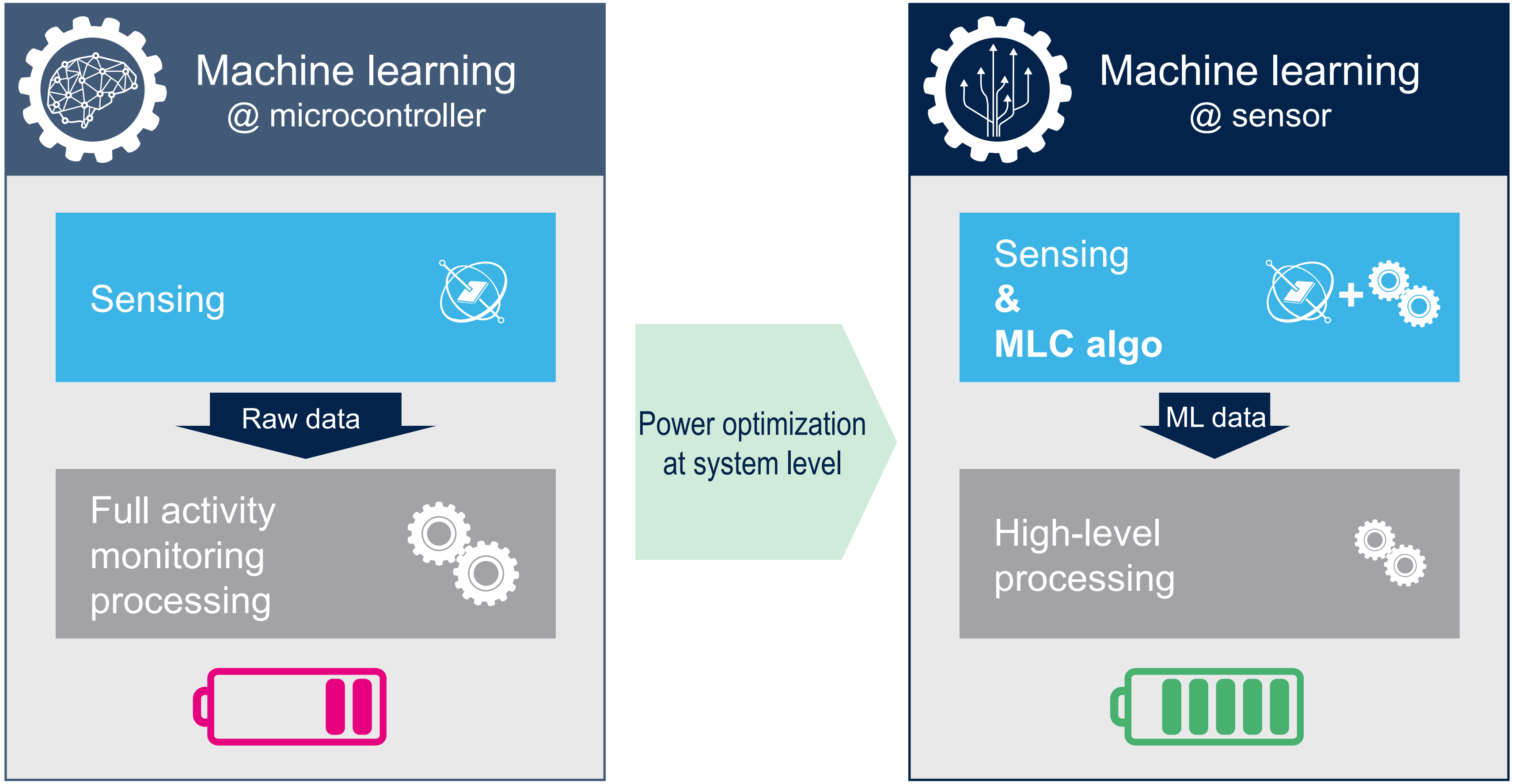
IMUs

- LSM6DSOX / 32X
- LSM6DSRX

MLC Gen. 1.0

From low-power sensor to low-power system

Machine learning core (MLC) for real edge computing enables high system flexibility



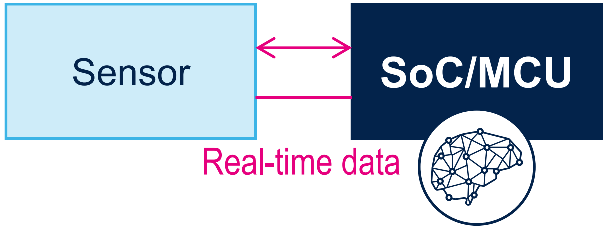
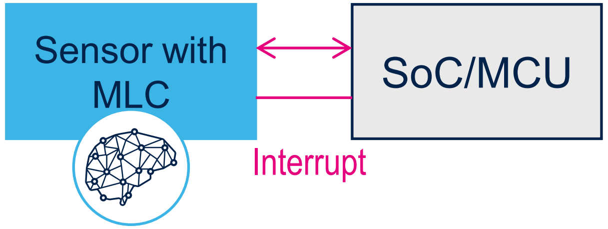
FSM
up to 16

MLC
up to 8

- Higher computation power at sensor level
- Lower power consumption at system level
- Cost-optimized solution

This is added value!

System level benefits using sensors with MLC

Parameters	Activity recognition in **software**	Activity Recognition in **sensors with MLC**	Benefits in using MLC
			
LSM6DSOX current			Similar consumption on LSM6DSOX
MIPS MCU/SoC			Offload SoC / MCU
Traffic			Less traffic on bus
Battery usage			Reduction of system current

Key steps behind machine learning core

Machine learning solutions in sensors
New developer model approach

Shorter development time and better accuracy with machine learning techniques (decision trees)

Machine learning core configuration → Operating mode

WHAT

Capture data
- Accelerometer
- Gyroscope
- External sensors

Label data
- Filters
- Features

Build decision tree
- Classification
- Results

Embed decision tree
- DT implementation

Process new data
- Real time test

Example of sensors MLC programming

10 to 1000-time energy saving by running MLC on Sensor vs. MCU/AP
Improved detecting accuracy

How it works in 5 simple steps and with an intuitive use case:

1

User defines **classes** to be recognized

Capture data

2

Label data and select **filters and features**

Label data

3

Build the decision tree based on a wide range of software tools.

Build decision tree

4

Program the decision tree into the MLC-enabled sensor

Embed decision tree

5

Run the MLC model and process incoming data in real time

Process new data

Current consumption: AI in MCU vs. MLC in sensor

Only 4µA additional current consumption to run Activity Recognition with MLC

Activity recognition library (MotionAR) running **in software in MCU**

In this scenario, the MCU wakes up to read all new sensor data

LSM6DSOX sensor	Current consumption
Sensor core	15 µA
MLC (not used)	0 µA

MCU	Wake-up rate	Current consumption
MCU Core	1/16 = 63 ms	51 µA

Total: 66 µA

Activity recognition library (MotionAR) running **inside the LSM6DSOX sensor**

In this scenario, the MCU wakes up only when a new class is detected

LSM6DSOX sensor	Current consumption
Sensor core	15 µA
MLC	4 µA

MCU	Wake-up rate	Current consumption
MCU Core	30 s	0.65 µA

Total: 20 µA

3 x power saving

In both scenarios the ODR of the sensor is set in the same condition (ODR 26 Hz, LP mode) and the same sampling time window.

Form factor tools and GUI

Machine learning solutions in sensors
The ecosystem

A complete suite to create ML applications in sensors

st.com/mlc

Programming with **ST tools** and **ST partners**

MEMS Studio · ST AIoT Craft · ARDUINO

Getting start with **ST development kit**

Examples for motion recognition and context recognition

GitHub · ST Edge AI Suite

Videos, training material, in products campaign available

MEMS & Sensor community: **MEMS Machine learning & AI**

ST toolbox for machine learning core (MLC) in the sensor

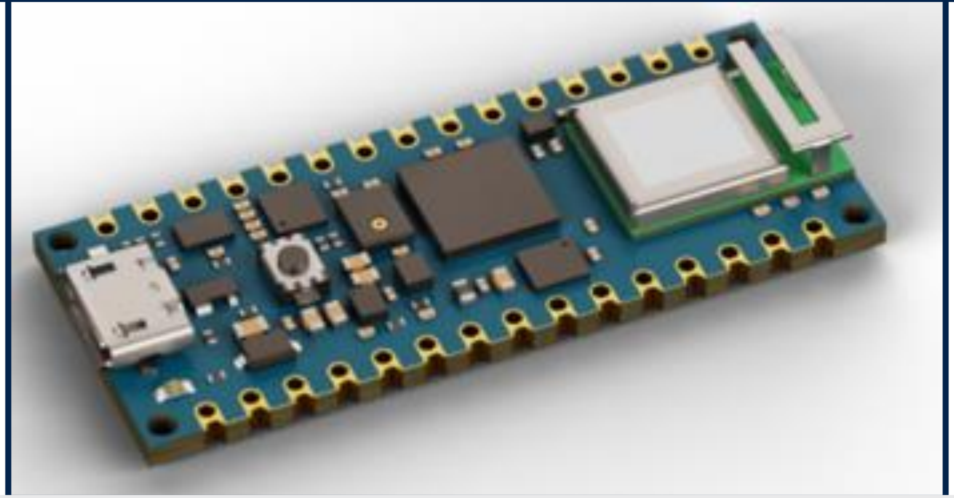
Machine learning core configuration → **Operating mode**

	Capture data	Label data	Build decision tree	Embed decision tree	Process new data
WHAT	• Accelerometer • Gyroscope • External sensors	• Filters • Features	• Classification • Results	• DT implementation	• Real time test
HOW	MEMS Studio ST AIoT Craft STBLESensor	MEMS Studio ST AIoT Craft	MEMS Studio ST AIoT Craft	MEMS Studio ST AIoT Craft	MEMS Studio ST AIoT Craft STBLESensor

ST Partner: Arduino® with machine learning core

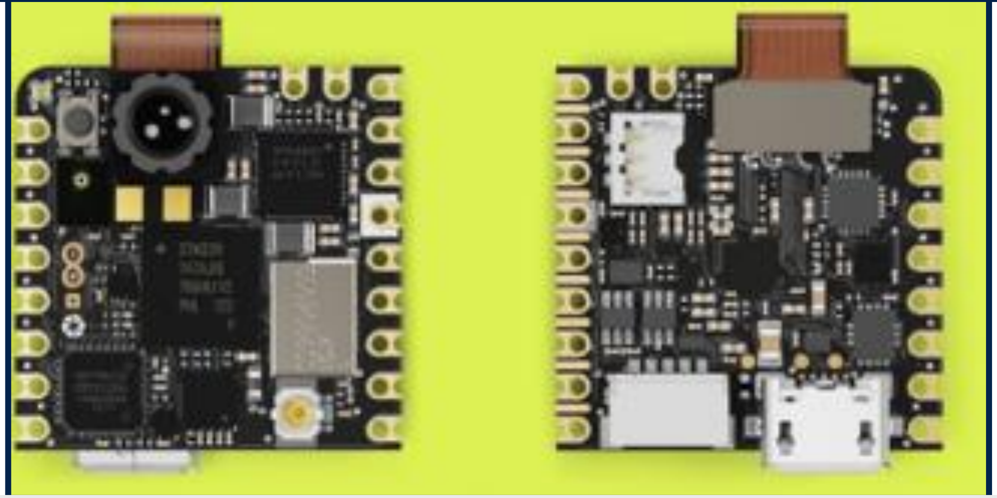
Arduino ecosystem for MLC

Arduino® RP2040



- Arduino® RP2040 mounts the LSM6DSOX 6 axis sensor with MLC
- Build the MLC with the Arduino® environment:
  - https://www.youtube.com/watch?v=hHVsLHqIN9g

Arduino® Nicla Vision



- Arduino® Nicla Vision mounts the LSM6DSOX 6 axis sensor with MLC
- Build the MLC with the Arduino® environment:
  - Webinar: Intelligent sensors for smart access monitoring – STMicroelectronics

MLC applications

Explore MLC examples and resources and get inspired

MLC examples are available online on **GitHub** and through the case studies section of the **ST Edge AI Suite**

ST Edge AI Suite

Consumer
6D position recognition, activity recognition, gym activity recognition, head gestures

Industrial
6D position recognition, motion intensity, vibration monitoring

Automotive
Vehicle stationary detection

… and more to come !

github.com/STMicroelectronics/STMems_Machine_Learning_Core

https://www.st.com/content/st_com/en/st-edge-ai-suite/case-studies.html

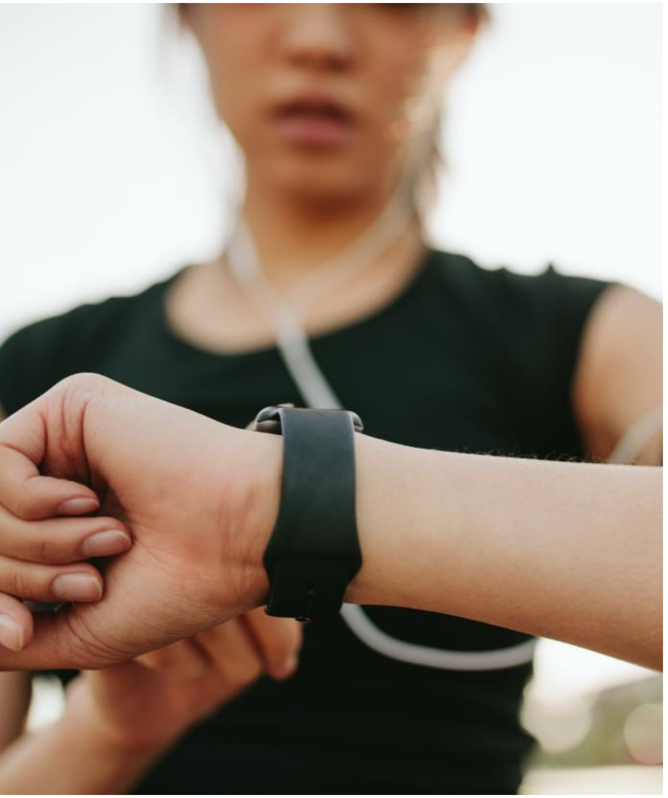
Get inspired by MLC applications

Industrial	Drilling machine status detection	Static inclinometer
	Fan rack monitoring	Cobot movement and anomaly detection
Personal electronics	Human activity recognition	Gym activity recognition
	Smart kettle	Smart access
	Gesture recognition	Yoga position recognition
Automotive	Baby crying detection	Vehicle monitoring

LSM6DSV32X
Activity recognition

Activity recognition for wrist applications

LSM6DSV32X



Use case

MLC embedded in LSM6DSV32X inertial sensor is programmed to detect 3 different classes using the MLC:

- **Stationary**
- **Walking**
- **Jogging**

Demo overview

- MLC with 3 classes
- MLC running on MEMS ASIC
- No pre-processing required from MCU

Demo setup

- Test platform: **SensorTile.box PRO + STEVAL-MKI240KA**
- Result output: **STBLESensor**
- Tutorial with **MEMS Studio**
- LSM6DSV32X: 6-axis IMU with 32 g accelerometer and embedded sensor fusion, AI, Qvar for high-end applications

▶ On-demand webinar available

ISM330BX
Movement and anomaly detection

Movement and anomaly detection in industrial cobots

ISM330BX

Use case

MLC embedded in ISM330BX inertial sensor is programmed to detect 3 different classes using the MLC:

- **Stationary**
- **Pick and place**
- **Anomaly**

Demo overview

- MLC with 3 classes
- MLC running on MEMS ASIC
- No pre-processing required from MCU

Demo setup

- Test platform: **STWIN.box + STEVAL-MKI245KA**
- Result output: **FP-SNS-DATALOG2**
- Tutorial with **MEMS Studio**
- ISM330BX: 6-axis IMU with wide bandwidth, low-noise accelerometer, embedded AI and sensor fusion for industrial applications

► On-demand webinar available

Industrial IMUs
Drilling machine status detection

Drilling machine status detection

LONGEVITY 10 YEARS COMMITMENT

ISM330DHCX / ISM330BX

Use case

- MLC embedded in ISM330DHCX / ISM330BX inertial sensor is programmed to detect the **status of the drilling machine**.
- 3 different classes can be identified by the MLC:
  - "**Idle**" state
  - "**Drilling**" state
  - "**Screw turning**" state

Demo overview

- MLC with 3 classes
- MLC running on MEMS ASIC
- No pre-processing required from MCU

Demo setup

- Test platform: **STWIN evaluation kit**
- Result output: **Unicleo / MEMS Studio**
- Training data acquired using **Unicleo / MEMS Studio**
- ISM330DHCX / ISM330BX: inertial sensor with 3-axis accelerometer + gyroscope

► Demo available

ISM330DHCX
Drilling machine status detection

Drilling machine status detection

10 YEARS LONGEVITY COMMITMENT

ISM330DHCX

Use case

- MLC embedded in ISM330DHCX inertial sensor is programmed to detect the **status of the drilling machine**.
- 3 different classes can be identified by the MLC:
  - "**Idle**" state
  - "**Drilling**" state
  - "**Screw turning**" state

Demo overview

- MLC with 3 classes
- MLC running on MEMS ASIC
- No pre-processing required from MCU

Demo setup

- Test platform: **STWIN evaluation kit**
- Result output: **Unicleo**
- Training data acquired using **Unicleo**
- ISM330DHCX: inertial sensor with 3-axis accelerometer + gyroscope

▶ Demo available

IIS2ICLX
Static inclinometer

Static inclinometer

IIS2ICLX

Use case

- MLC embedded in IIS2ICLX inertial sensor is programmed to detect the **truck bed positioning**
- 3 different classes can be identified by the MLC:
  - “**Closed**” truck bed
  - **“Moving”** truck bed
  - **“Fully opened**” truck bed

Demo overview

- MLC with 3 classes
- MLC running on MEMS ASIC
- No pre-processing required from MCU

Demo setup

- Test platform: **IIS2ICLX adapter board** with **ProfiMEMS** tool motherboard for sensor interaction
- Result output: **UNICO-GUI** SW for Win/Mac
- Training data acquired using ProfiMEMS motherboard and adapter board

► Demo available

LSM6DSOX – LSMDSV16X / 32X
Human activity recognition

Human activity recognition

LSM6DSOX

Use case

- MLC embedded in LSM6DSOX inertial sensor is programmed to detect **Human Activity Recognition** in different scenario
- The MLC identifies 5 scenarios:
  - **“Stand up”** scenario
  - **“Walking”** scenario
  - **“Running”** scenario
  - **“Cycling”** scenario
  - **“Car driving”** scenario

Demo overview

- MLC with 5 classes
- MLC running on MEMS ASIC
- No pre-processing required from MCU

Demo setup

- Test platform: **SensorTile.box** evaluation kit
- Result output: **STBLESensor** (Android/iOS)
- Training data acquired using **ST BLE app**
- LSM6DSOX: inertial sensor with 3-axis accelerometer + gyroscope

► Demo available

LSM6DSOX – LSMDSV16X / 32X
Gym activity recognition

Gym activity recognition

LSM6DSOX

Use case

- MLC embedded in LSM6DSOX inertial sensor is programmed to detect gym activity by identifying 4 common actions
- The MLC identifies 4 scenarios:
  - **“No activity”** scenario
  - **“Bicep curls”** scenario
  - **“Lateral raises”** scenario
  - **“Squats”** scenario

Demo overview

- MLC with 4 classes
- MLC running on MEMS ASIC
- No pre-processing required from MCU

Demo setup

- Test platform: **any board** with LSM6DSOX
- Result output: **MLC value**
- Training data acquired using **Sensortile.box**
- LSM6DSOX: inertial sensor with 3-axis accelerometer + gyroscope

► Teaser available, GitHub example available

LSM6DSOX – LSMDSV16X / 32X Smart kettle

Smart kettle

LSM6DSOX

Use case

- MLC embedded in LSM6DSOX inertial sensor is programmed to detect the status of a boiling kettle
- The MLC identifies 2 stats scenarios:
  - **“Idle”** state
  - **“Boiling”** state

Demo overview

- MLC with 3 classes
- MLC running on MEMS ASIC
- No pre-processing required from MCU

Demo setup

- Test platform: **SensorTile.box** evaluation kit
- Result output: **STBLESensor** (Android/iOS)
- Training data acquired using **ST BLE app**
- LSM6DSOX: inertial sensor with 3-axis accelerometer + gyroscope

► Demo available

LSM6DSOX
Smart access

Smart access

LSM6DSOX

Use case

- MLC embedded in LSM6DSOX inertial sensor is programmed to detect the check a safety box status
- The MLC identifies 4 stats scenarios:
  - “**Steady**” state
  - “**Gear**” state
  - “**Door Opening**” state
  - “**Burglar**” state

Demo overview

- MLC with 4 classes
- MLC running on MEMS ASIC
- No pre-processing required from MCU

Demo setup

- Test platform: **Arduino Nicla Vision**
- Result output: **MLC value**
- Training data acquired using **Arduino Nicla Vision**
- LSM6DSOX: inertial sensor with 3-axis accelerometer + gyroscope

▶ Webinar available

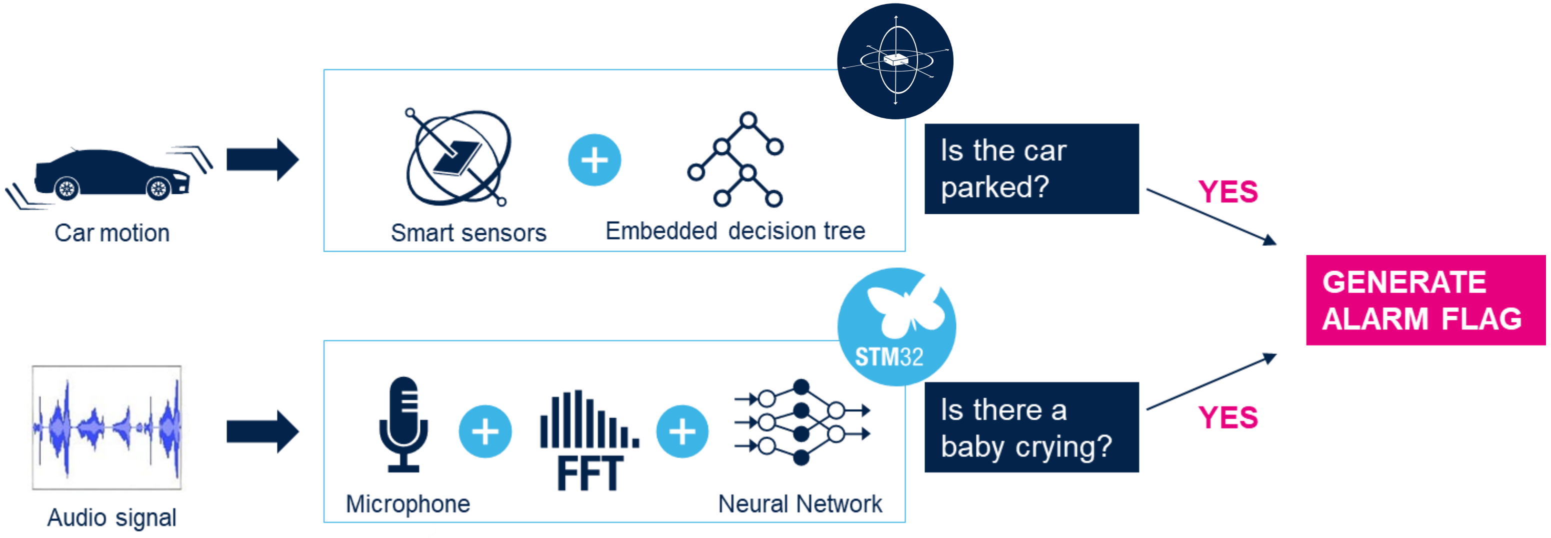
SensorTile.box
Baby crying detection

Baby crying detection

Use case

MLC for vehicle detection combined with fusion of ST MEMS microphones and STM32 AI to create a fully-trained neural network detecting if there is a baby crying in the room.

- Detect a baby crying in car left unattended in a parking lot.
- By detecting the baby crying and the idle state of the car, the application can ideally raise an alarm flag by a system to communicate the dangerous situation to the outside world.



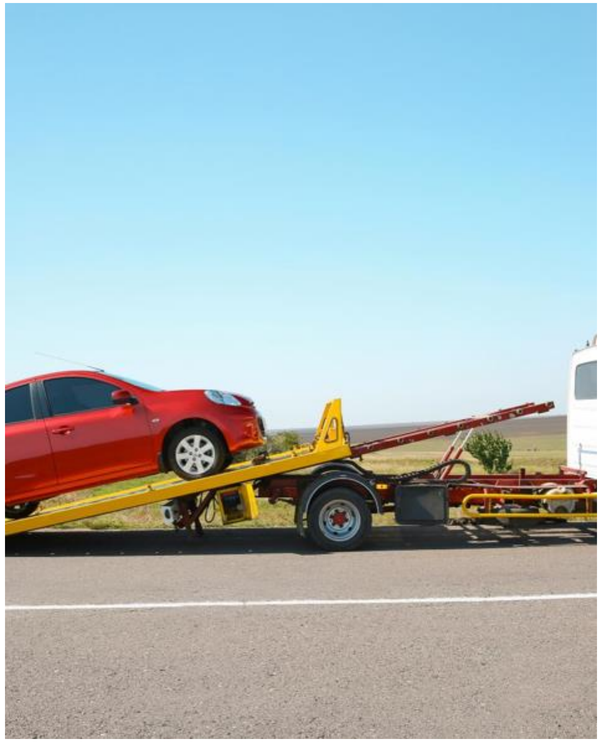
This demonstration running on the **SensorTile.box** can be used as a starting point for more complex solutions involving child safety.

▶ Demo available

ASM330LHHX – ASM330LHHXG1 Vehicle monitoring

Vehicle monitoring

ASM330LHHX



Use case

- MLC embedded in ASM330LHHX inertial sensor is programmed to detect different vehicle scenario
- The MLC identifies 4 stats scenarios:
  - **“Still”** state
  - **“Moving”** state
  - “**Front & lateral jack up**” state
  - **“Crash”** state

Demo overview

- MLC with 4 classes
- MLC running on MEMS ASIC
- No pre-processing required from MCU

Demo setup

- Test platform: **Profi MEMS** with ASM330LHHX
- Result output: **MLC value**
- Training data acquired using Profi MEMS
- LSM6DSOX: inertial sensor with 3-axis accelerometer + gyroscope

▶ Webinar available

LSM6DSOX
Yoga position detection

Yoga position detection

LSM6DSOX

Use case

MLC embedded in LSM6DSOX inertial sensor is programmed to detect 13 different classes using the MLC:

- Boat pose
- Bow pose
- Bridge
- Child's pose
- Cobra pose
- The tree
- Downward-facing dog
- Meditation pose
- Plank
- Seated forward bend
- Upward plank
- Standing in motion
- Standing still
- The extended side angle
- Plank
- Seated forward bend

Demo overview

- MLC with 13 classes
- MLC running on MEMS ASIC
- No pre-processing required from MCU

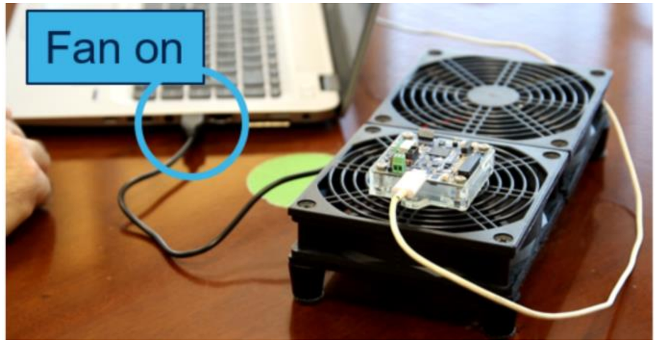
Demo setup

- Test platform: **SensorTile.box**
- Result output: **STBLESensor**
- Decision tree generated with **Unico**
- LSM6DSOX: inertial sensor with 3-axis accelerometer + gyroscope

GitHub Link

Fan rack monitoring

ISM330DHCX



Use case

MLC embedded in ISM330DHCX inertial sensor is programmed detect 3 states:

- **“Fan_on” state**
- **“Fan_off” state**
- **“Fan_fault” state**

Demo overview

- MLC with 3 classes
- MLC running on MEMS ASIC
- No pre-processing required from MCU

Demo setup

- Test platform: **STWIN**
- Result output: **STBLEApp**
- Decision trees generated with **Unico - GUI**
- ISM330DHCX : inertial sensor with 3-axis accelerometer + gyroscope

GitHub Link

Resources

Resources

MEMS sensors ecosystem for machine Learning

MEMS sensors community

Design Tips for Machine Learning algorithms generation

Automatic filters and features selection for Machine Learning Core in MEMS Studio

Deploy in-sensor AI in your IoT infrastructure with a cloud-based tool

Takeaways

We create the **new generation of sensors** to allow developers exploiting their potential by improving the overall system **efficiency leveraging Machine Learning techniques**

ST has **all the building blocks** for the **machine learning core** for tools and software

Our constantly expanding development ecosystem makes **design fast** and help you **reduce the time-to-market**

life.augmented

Our technology starts with You

Find out more at www.st.com