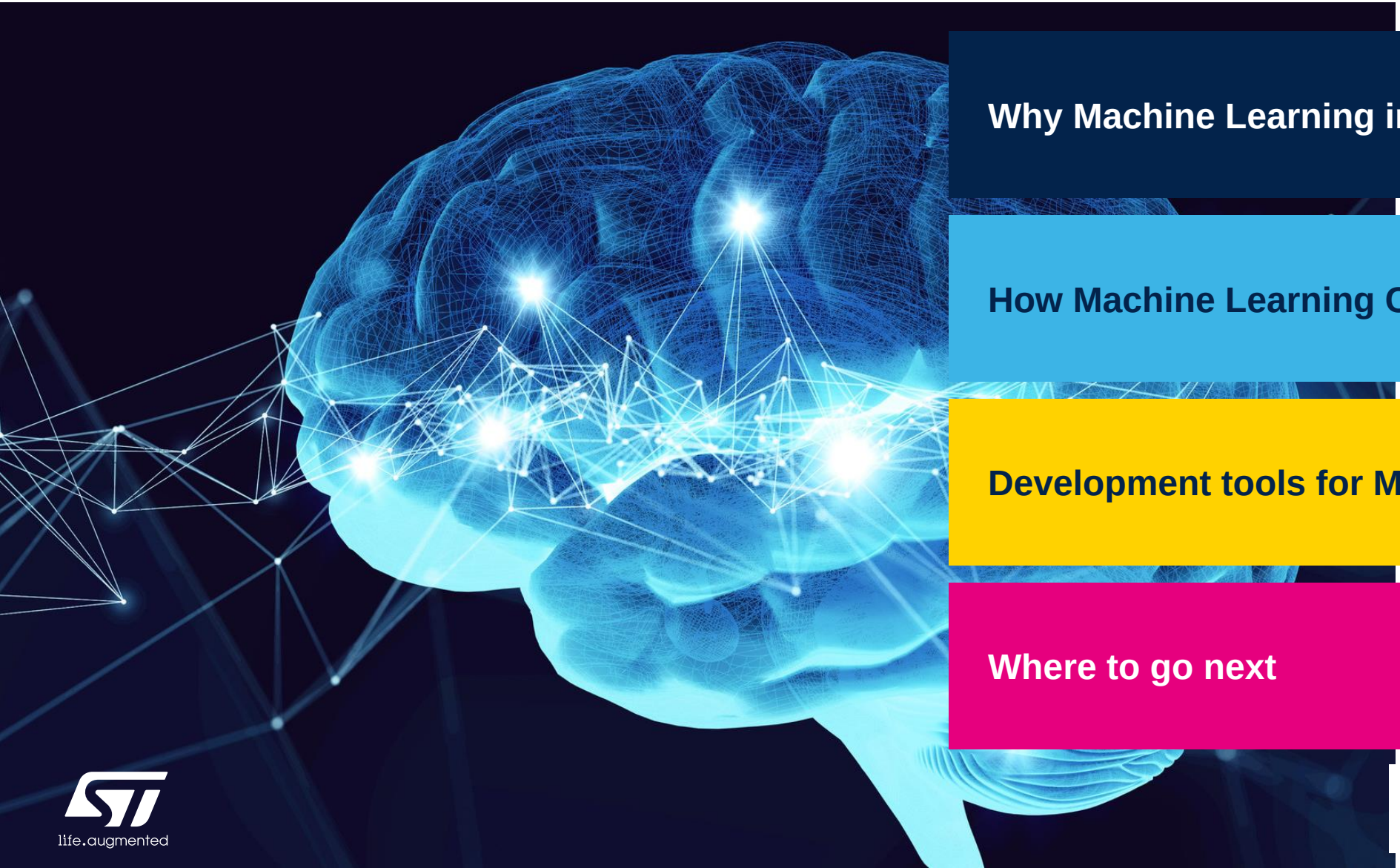
Focus on IMU with Machine Learning Core



Why Machine Learning in a sensor

How Machine Learning Core works

Development tools for MLC

Where to go next

life.augmented

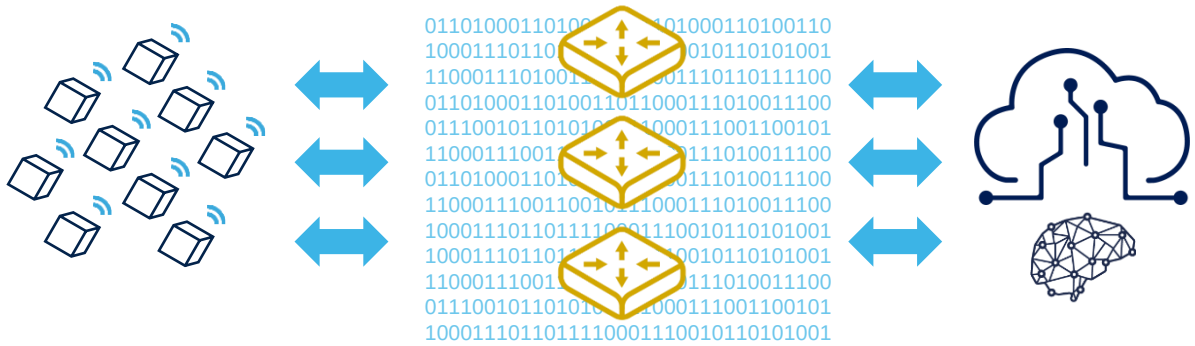
Moving to edge computing

CLOUD COMPUTING

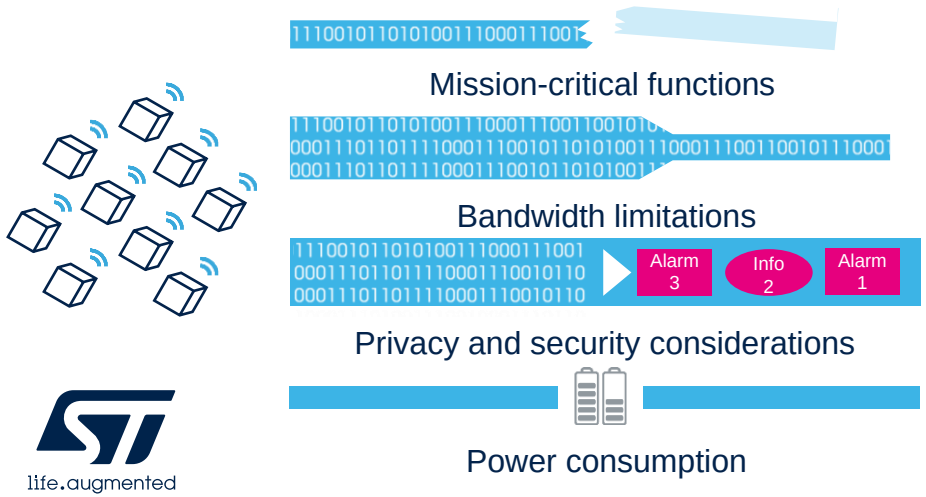
Collect and send data

Protocol translation and device management

Big Data and heavy computation

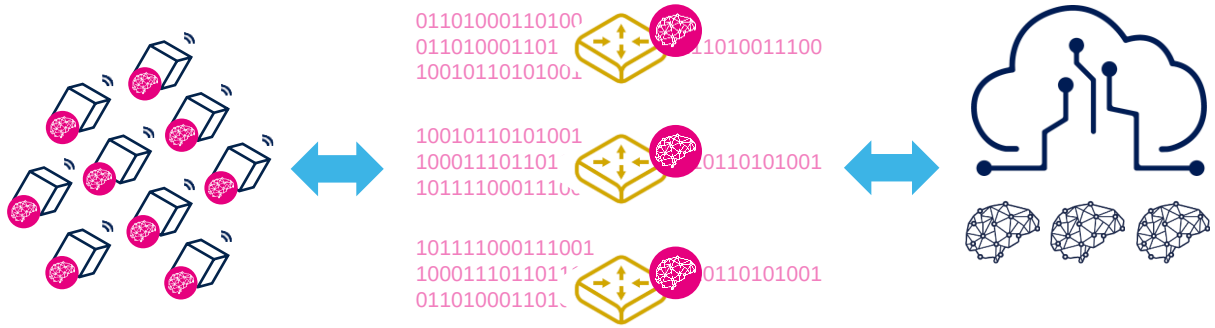


Time-sensitive applications are limited by remote cloud



EDGE COMPUTING

Time-sensitive applications should be locally processed



Collect, Process And Send Data

Local Processing of Data

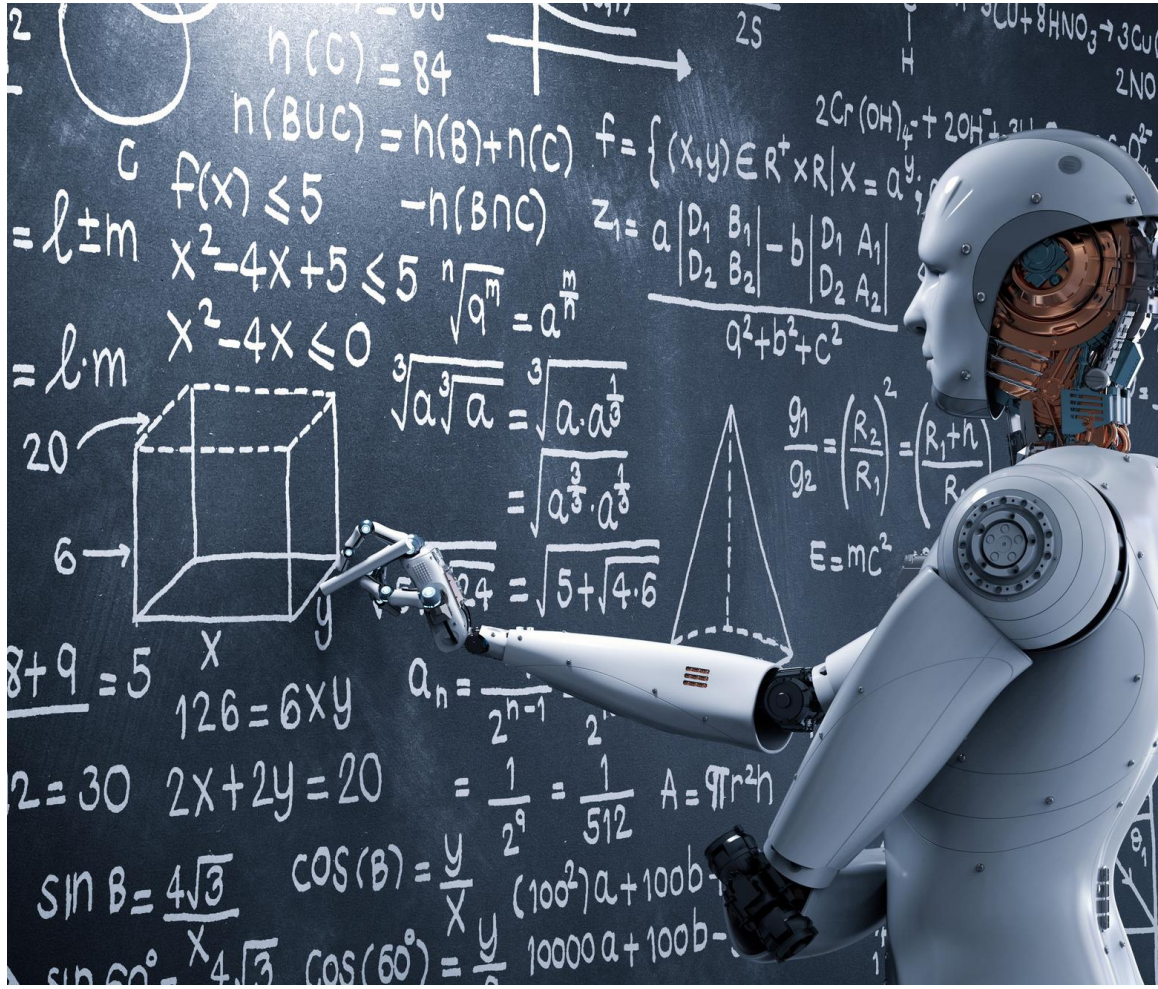
Optimized computation and Advanced Analysis

Opportunity: move computation to sensor nodes with local processing for real-time elaboration and best power efficiency

life.augmented

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**

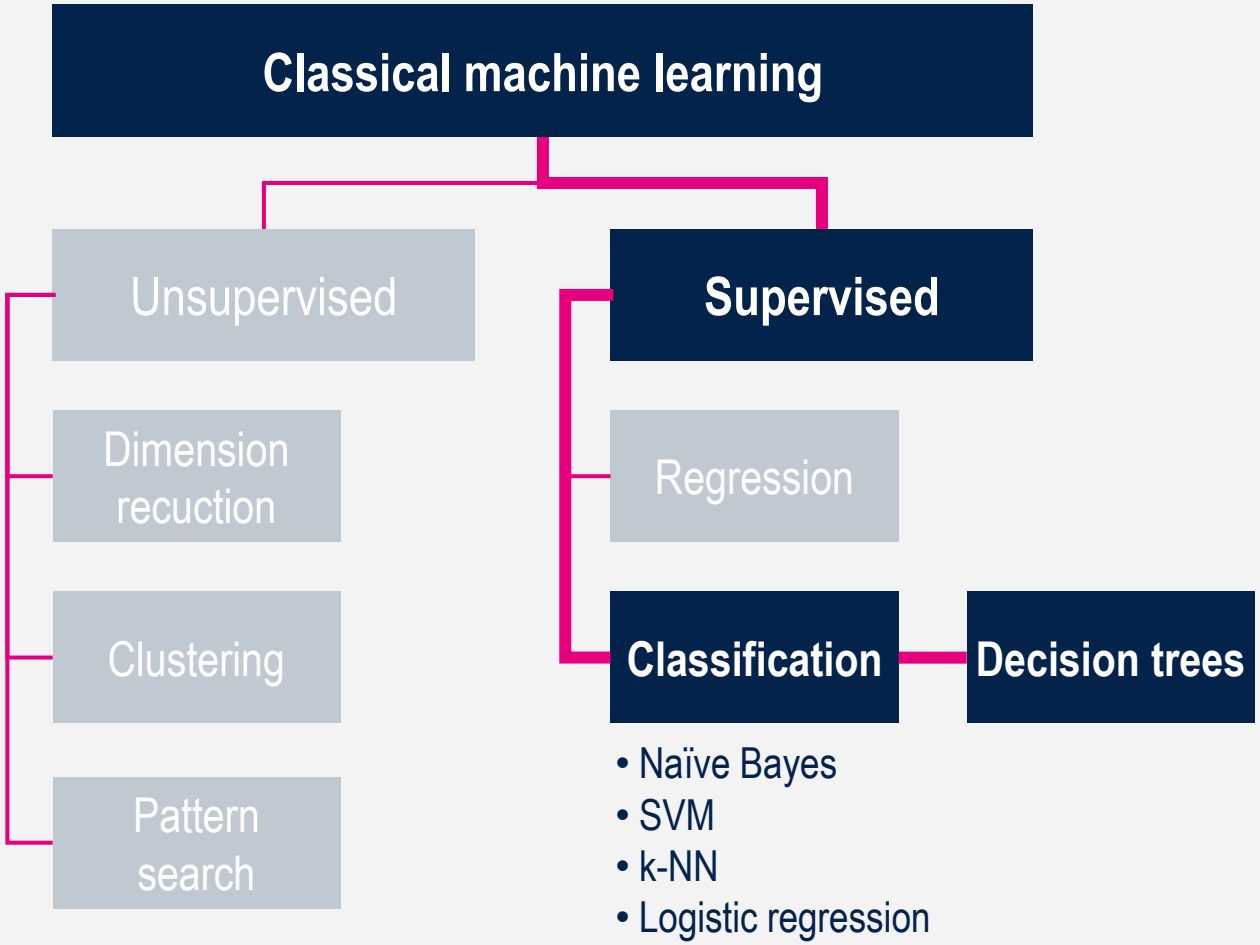


An example of difficult program:

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?

Machine Learning embedded in ST Sensors



ST sensors embed **Decision trees**

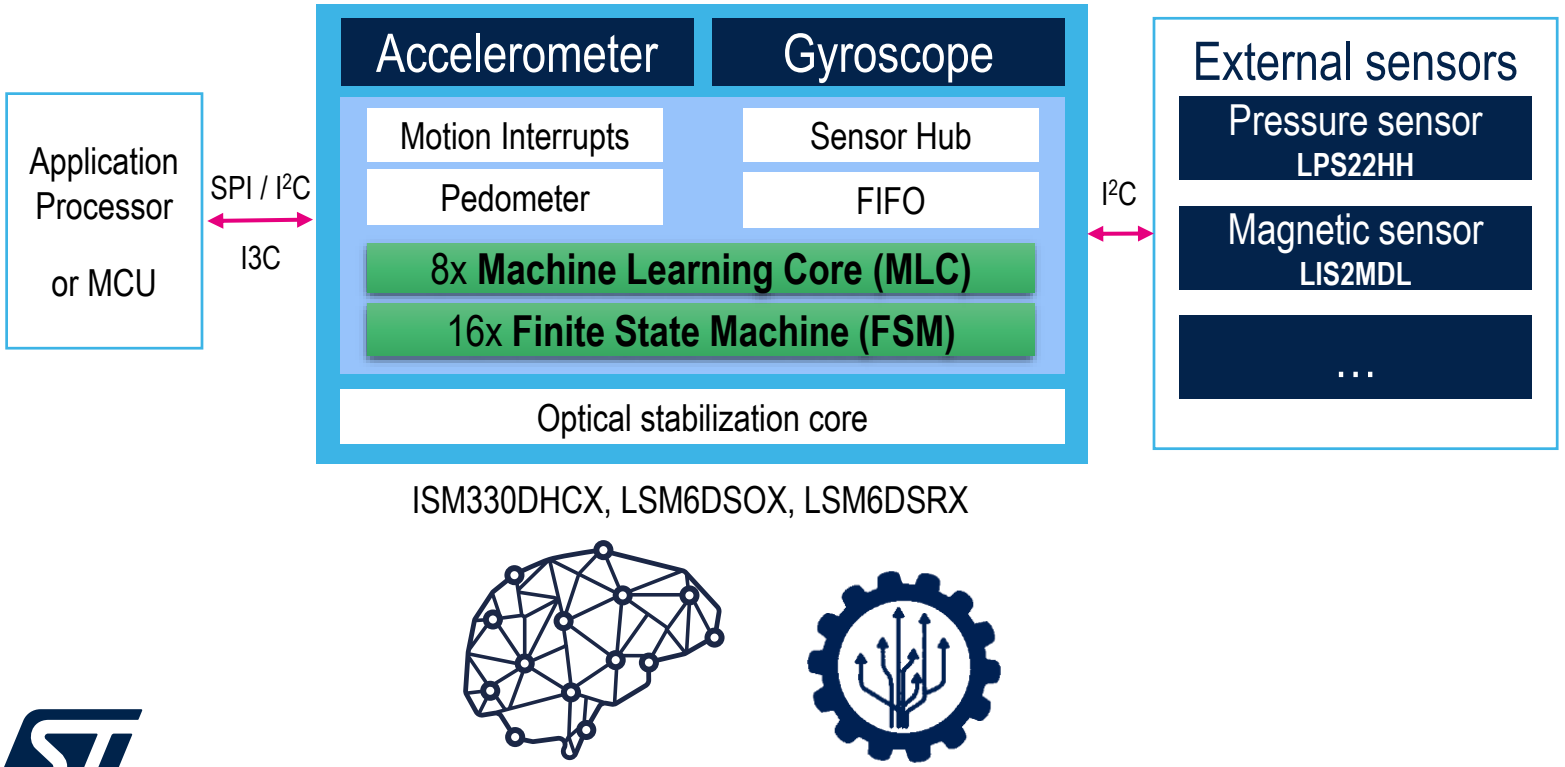
Decision tree runs **Classification**, i.e algorithm that splits objects into classes based on attributes known beforehand

Supervised machine learning technique is used to create decision trees

6-axis IMUs with Machine Learning Core

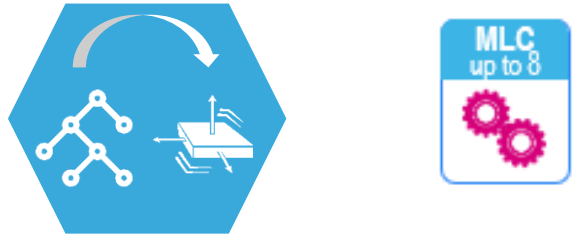
From low power sensor to low power system

Advanced Features

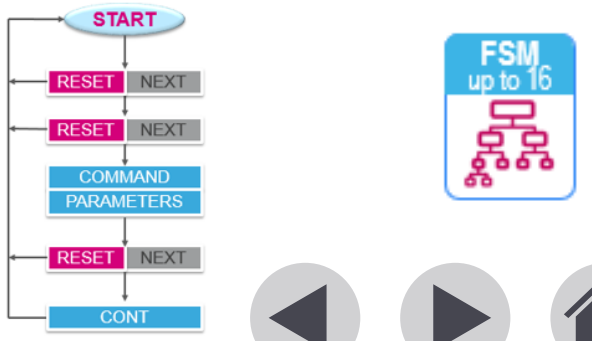


Embedded AI

MLC with embedded Decision Trees



Finite State Machine



life.augmented

More intelligence with an embedded Machine Learning Core

Get inspired by MLC examples!

Personal Electronics

- Activity recognition
- Gym activity recognition
- Head gestures

Industrial IoT

- Motion intensity
- Orientation detection
- Vibration monitoring

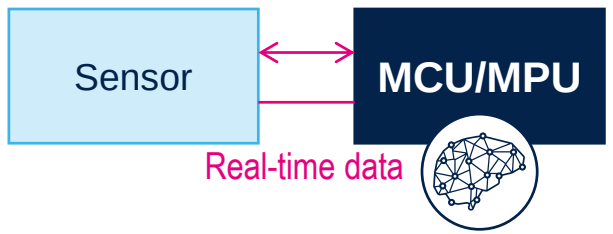
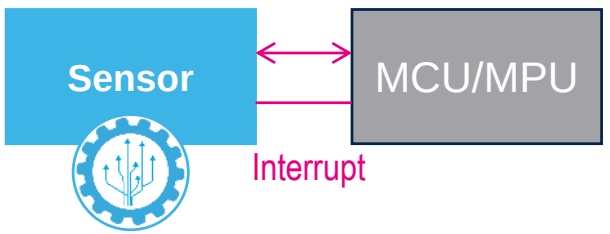
Automotive

- Vehicle stationary detection

MLC examples are available online at dedicated GitHub project for Machine Learning Core

life.augmented

System level benefits using MLC

Parameters	Activity recognition in **SW**	Activity recognition in **MLC**	Benefits in using MLC
			
Sensor consumption			Similar consumption of the sensor
MIPS MCU/MPU			Offload MCU/MPU
Traffic on serial bus			Less traffic on bus
Battery usage			Reduction of system current

>1 to 100 times better current consumption with MLC!

life.augmented

Current consumption: AI in MCU vs. MLC in sensor

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

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

LSM6DSOX Sensor		Sensor Current consumption
Sensor Core		15 µA
MLC – not used		0 µA
MCU	**Wake-up rate**	**MCU Current consumption**
STM32L476RG	1/16 = 63ms	51 µA

Total: **66µA**

Activity recognition algorithm running **inside LSM6DSOX**

LSM6DSOX Sensor		Sensor Current consumption
Sensor Core		15 µA
MLC		4 µA
MCU	**Wake-up rate**	**MCU Current consumption**
	1 s	2.8 µA
STM32L476RG	30 s	0.65 µA
	100 s	0.59 µA

Total: **20µA**
3x power saving

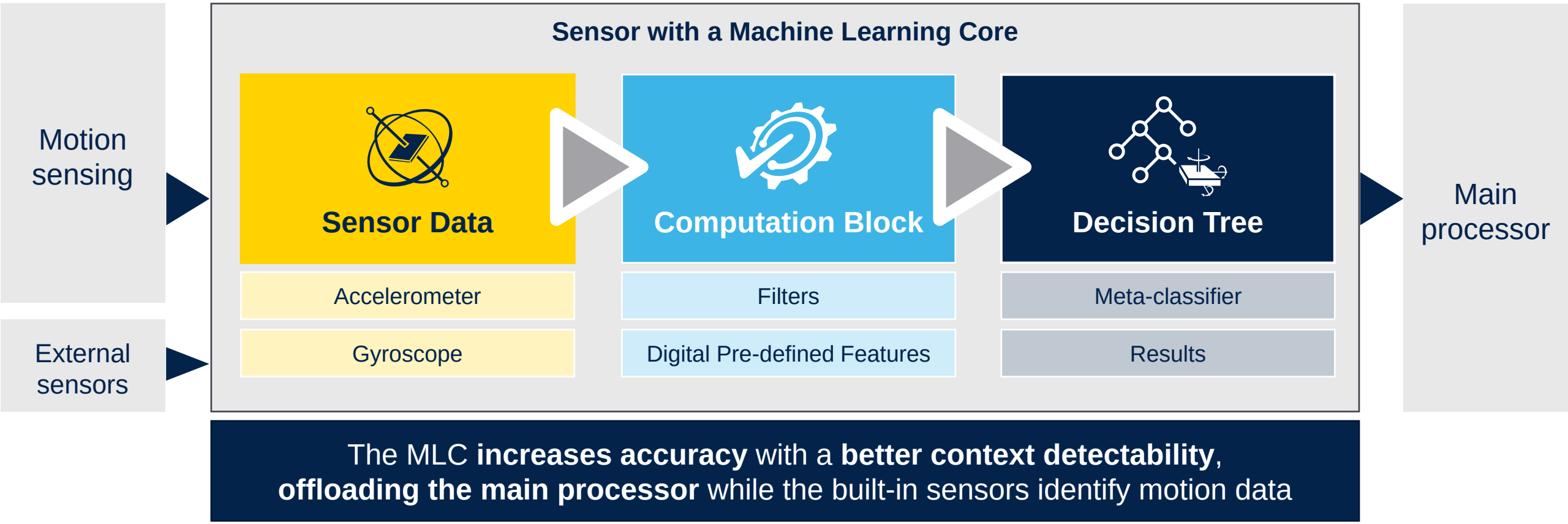
In both scenario the ODR of the sensor is set in the same condition (ODR 26Hz, LP mode) and same sampling time window.

In the first scenario the microcontroller wakes up to read all new sensor data, in the second scenario the microcontroller wake ups only when a new class is detected.
From system power consumption point of view, the second scenario brings significant power consumption improvement since the microcontroller is less active.

life.augmented

Machine Learning Core (MLC) Definition

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



The MLC **increases accuracy** with a **better context detectability**, **offloading the main processor** while the built-in sensors identify motion data

life.augmented

MLC inputs

MLC works with multiple inputs

Sensor Data

- Accelerometer
- Gyroscope
- External sensor

A wide set of inputs to be chosen

1. Accelerometer $[a_x\ a_y\ a_z\ ], [a_v], [a^2_v]$
2. Gyroscope $[g_x\ g_y\ g_z\ ], [g_v]\ , [g^2_v]$
3. External sensor $[m_x\ m_y\ m_z\ ], [m_v]\ , [m^2_v]$

Magnitude Available

$$V = \sqrt{X^2 + Y^2 + Z^2}$$

MLC computation block

Rich set of digital filters and features

Computation Block

Filters

Digital Pre-defined Features

2nd order IIR filters and statistical features

Filters: high-pass, band-pass, user- defined IIR 1st or 2nd order
Features computed on a sequence of input samples:

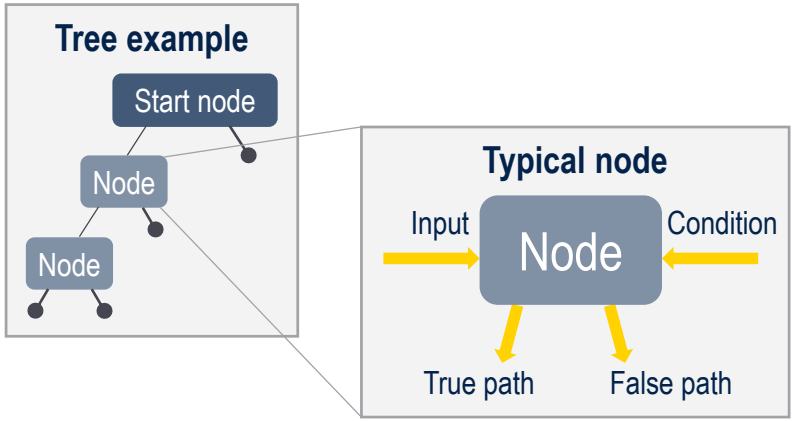
- MEAN
- VARIANCE
- ENERGY
- PEAK TO PEAK
- ZERO CROSSING
- PEAK DETECTOR
- MINIMUM
- MAXIMUM

MLC decision tree

MLC performs classification based on sensor data

Decision Tree: A predictive model built from training data

- Decision tree is composed by nodes, it is a binary tree
- Meta-classifier is a filter on the outputs of the decision tree
- Decision tree is automatically built by dedicated ML tool



life.augmented

What should be done to have decision tree running in the sensor ?

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

1

User defines **classes** to be recognized and **collects data logs**.

2

Clean and **label** logs. Define **features** best characterizing the identified classes.

3

Machine Learning tools **build tree** based on logs and features.

4

ST tool generates sensor configuration with **embedded decision tree.**

5

Configure the sensor and **run** the application.

Capture data

Label data & extract features

Build decision tree

Embed decision tree

Process new data

life.augmented

Machine Learning process with ST tools

Capture data

Label data & extract features

Build decision tree

Embed decision tree

Process new data

* Alternatively other external tools:
Weka, RapidMiner, MATLAB, Python

Unico GUI → PC tool for MLC development

AlgoBuilder → PC tool for sensor algorithms development

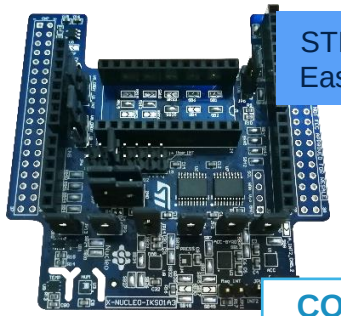
Unicleo-GUI → PC tool for STM32 Nucleo with MEMS expansion board

ST BLE Sensor → Mobile App for SensorTile.box

life.augmented

Evaluation boards

STM32Nucleo expansion X-NUCLEO-IKS01A3



STM32CubeMX
Easy debugging

CONSUMER

Profi MEMS tool STEVAL-MKI109V3

Evaluate All ST sensors through DIL24 adapter

AUTOMOTIVE

INDUSTRIAL

CONSUMER

BlueNRG-Tile STEVAL-BCN002V1B

Focus on BLE communication

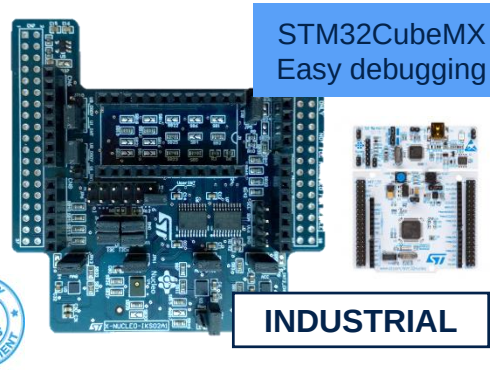
CONSUMER

SensorTile.Box STEVAL-MKSBOX1V1

Play with ST Sensors

CONSUMER

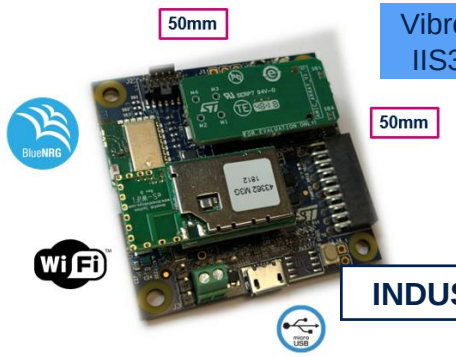
STM32Nucleo expansion X-NUCLEO-IKS02A1



STM32CubeMX
Easy debugging

INDUSTRIAL

STWIN: Wireless Industrial Node STEVAL-STWINKT1



Vibrometer IIS3DWB

INDUSTRIAL

life.augmented

Watch the webinar

Register and watch
the step-by-step video.

Webinar | Program decision trees in sensors with a Machine Learning Core

How to build decision trees in sensors with Machine Learning to create power-efficient AI applications for edge computing solutions

During this one-hour webinar, you will learn how to run a classification engine on the Machine Learning Core embedded in our latest iNEMO™ inertial modules, based on a decision-tree logic. In this webinar we will show you how to quickly and easily design power-efficient decision trees using the AlgoBuilder Graphical User Interface and ensure they provide accurate results in the shortest possible time.

From theory to practice, we will implement the ready-to-go IoT node SensorTile.box, together with AlgoBuilder

Email Address *

Country/Region *

Select one

I want to stay informed about ST's latest news

I consent that ST (as data controller according to the Privacy Policy) will keep a record of my navigation history and use that information as well as the personal data that I have communicated to ST for marketing purposes relevant to my interests. My personal data will be provided to ST affiliates and distributors of ST in countries located in the European Union and outside of the European Union for the same marketing purposes READ MORE

I understand that I can withdraw my consent at any time through opt-out links embedded in communication I receive or by managing my account settings. I can also exercise other user's rights at any time as described in the Privacy Policy.

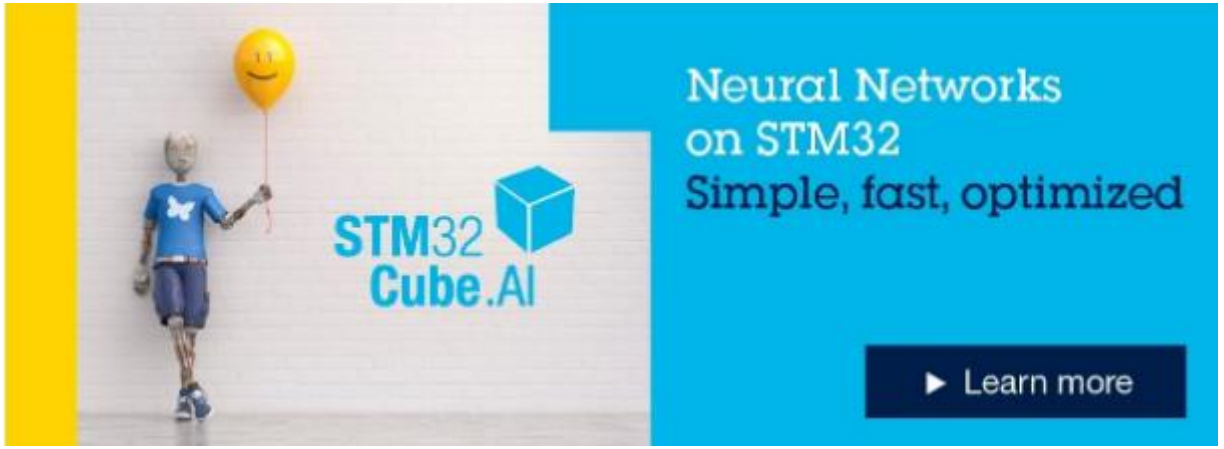
I have read and understood the Terms of Use and Privacy Policy *

Submit

life.augmented

Artificial intelligence at STMicroelectronics

- Thanks to STM32Cube.AI, pre-trained **Artificial Neural Networks** (ANN) can be run on STM32 microcontrollers.



- Advanced sensors contain a **Machine Learning Core (MLC)**, a Finite State Machine (FSM), and advanced digital functions. They run custom algorithms on the IMU and share the workload from the main processor enabling system functionality while significantly saving power.

life.augmented

MLC toolbox - a complete suite

Function packs for quick prototyping

Getting start with **ST development kit and GUI**

Videos, training material, in products campaign available

Examples for motion recognition and context recognition

MEMS & Sensor community:

MEMS and Sensors
MEMS Machine Learning & AI

www.st.com/content/st_com/en/campaigns/machine-learning-core.html

life.augmented