SIE 30 Años

GLOBALIoT

UNIÓN EUROPEA
FONDO EUROPEO DE DESARROLLO REGIONAL
"Una manera de hacer Europa"

Desarrollo de infraestructuras IoT de altas prestaciones contra el cambio climático basadas en inteligencia artificial

Retos-colaboración (RTC2019-007159-5)

José María Cecilia
Universitat Politècnica de València
(UPV)

GOBIERNO DE ESPAÑA

MINISTERIO DE CIENCIA, INNOVACIÓN Y UNIVERSIDADES

AGENCIA ESTATAL DE INVESTIGACIÓN

Agenda

El proyecto GLOBALoT [2]

El contexto científico [25]

Aplicación 1: Intelligent UAVs development [10]

Aplicación 2: Smart Greenhouses [8]

Colaboración Universidad-Empresa

El proyecto GLOBALoT.

GLOBALoT. Consorcio Universidad-Empresa

GLOBALoT call Retos Colaboración 2019.

(911.556,05€ = Préstamo: 512.201,21€ + Subvención 292.458,35€)

3

OBJETIVOS DEL PROYECTO

<u>Objetivos científicos del proyecto (OC):</u>

OC1.- Optimización de sistemas de computación y comunicaciones en infraestructuras IoT estáticas (***INVERNADEROS***) y móviles (***DRONES***).

OC2.- Optimización de recursos naturales y energéticos en invernaderos.

OC3.- Optimización del procesamiento de imágenes de drones para la prevención y resolución de catástrofes naturales.

<u>Objetivos tecnológicos (OT):</u>

OT1.- Desarrollo de un sistema avanzado de virtualización remota de GPUs de última generación en entornos IoT estáticos y móviles.

OT2.- Desarrollo de un prototipo de invernadero inteligente para la optimización del uso de recursos naturales.

OT3.- Desarrollo de un sistema de control para la captura y procesamiento eficiente de imágenes de drones.

El contexto científico

~~IoT~~ *enables industrial and socioeconmic process transformation*

AI+IoT

AIoT brings sensors, machines, cloud computing, analytics and people toguether to improve productivity and efficiency

Manufacturing

Agriculture

Policy-makers

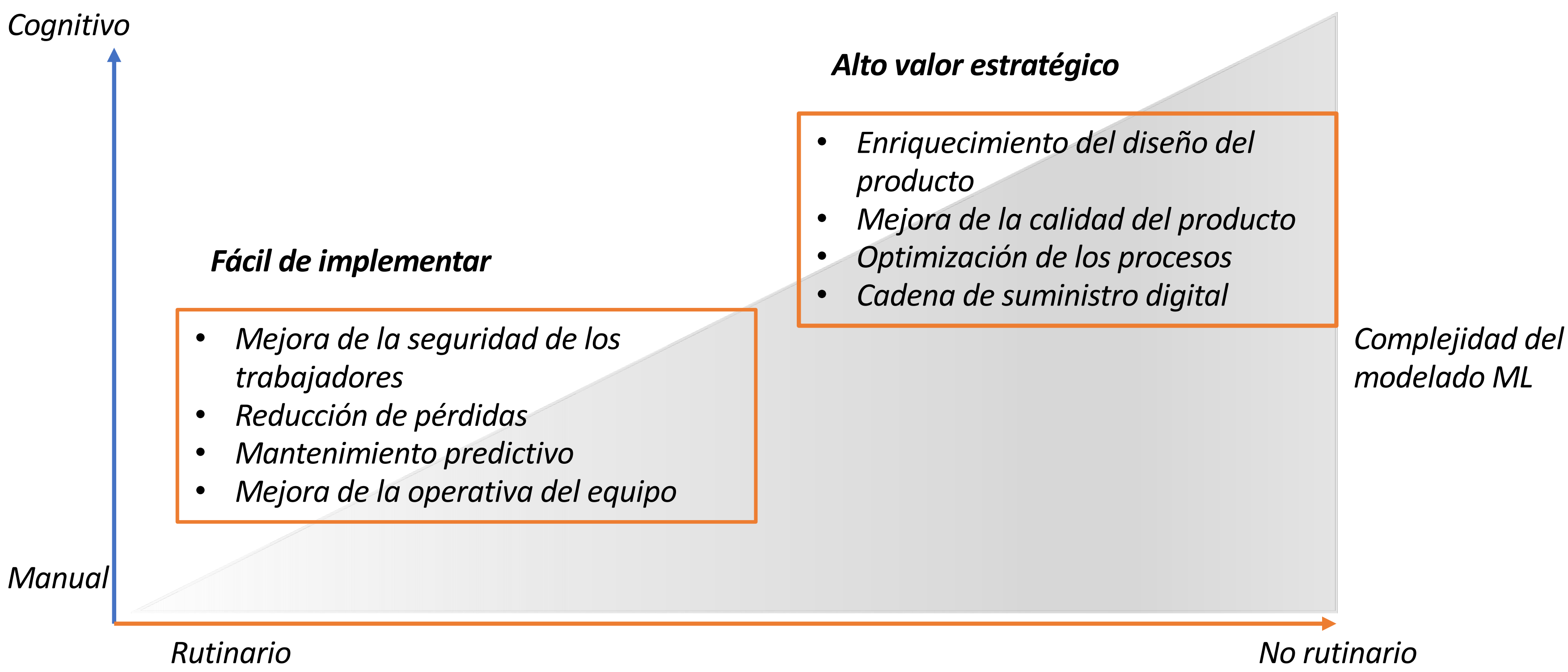
Natural disaster Management & prevention

No body just buys “AI”… or “IoT”

…. they seek bussiness outcomes

AI Implications

Progresión hacia la concienciación industrial de la IA



ML en datos generados por el IoT es complicado

ML is all about data
Los datos generados por las máquinas muchas veces son liosos, no representan la realidad por no hablar de los fallos….

IoT devices deben reportar los datos en formatos Simples para ser flexibles para la abstracción de funciones

- ***Falta de etiquetas, contexto y relaciones:*** Los sensores sólo mandan datos simples (float, int). Se debe construir el contexto de esos datos para añadirlo.

- ***Datos de mala calidad e integridad:*** El IoT tiene recursos limitados y en algunos cosos de mala calidad o en situaciones muy desafiantes tecnológicamente.

- ***Distinguir desviación de variación:*** Se necesita un análisis de datos avanzado para distinguir causalidad de casualidad.

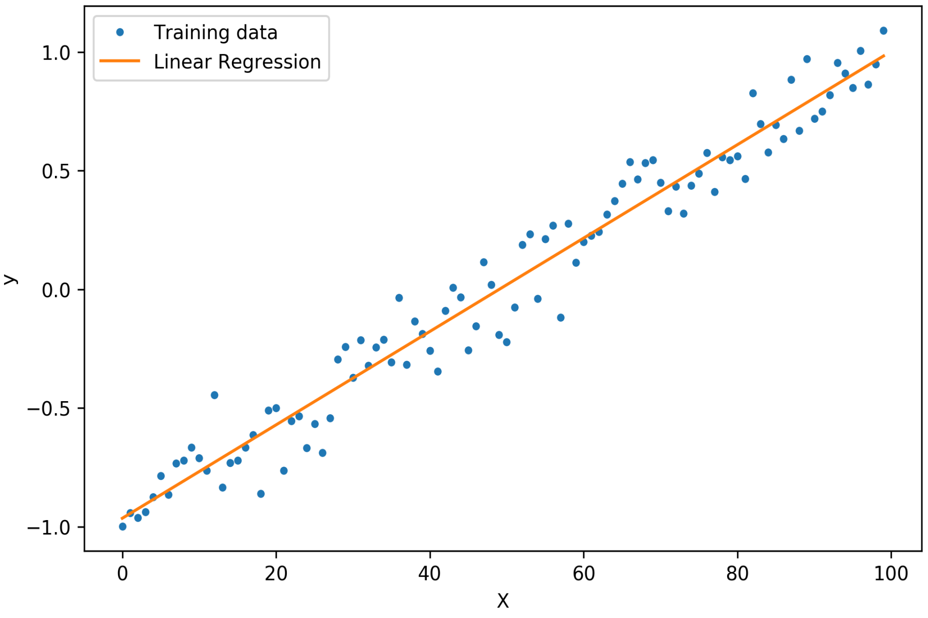
- ***Datos con muchas dimensiones:*** Se refiere al número de parámetros independientes en el análisis. Cuantas más variables los métodos de ML son computacionalmente más costosos

Alinear las técnicas ML con un propósito/función

Supervised Machine learning

- Usado comúnmente para alcanzar un objetivo predefinido basado en funciones algorítmicas que conectan la entrada con la salida.
- Dadas las entradas adecuadas , se puede predecir las salidas usando nuevos datos.
- Entre los modelos destaca método de regresión lineal, auto-regresivos (AR)
- Aplicable a series temporales mantenimiento predictivo, temperatura, vibraciones …

Regresion

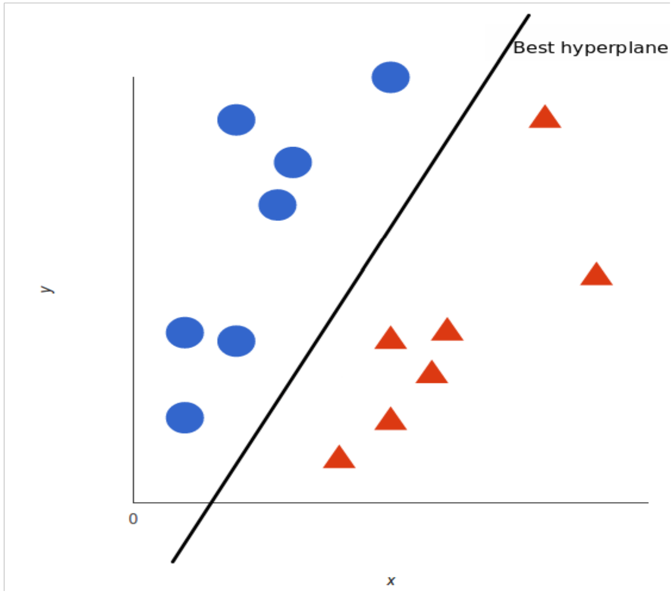


Alinear las técnicas ML con un propósito/función

Supervised Machine learning

- Asume que los datos pertenecen a unas categorías bien defindas.
- Aplicable a modelos basados en contextos donde se quiere determinar la causa (e.g. meteorología, hábitos de consumo)
- Entre los modelos destaca K-NN, SVM, RF, ANNs
- Aplicable a conjuntos de datos con múltiples rangos.

Classificación

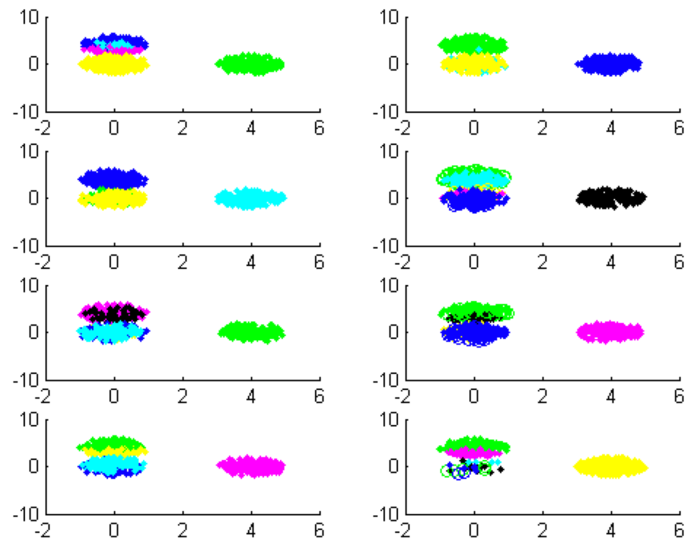


Alinear las técnicas ML con un propósito/función

Unsupervised Machine learning

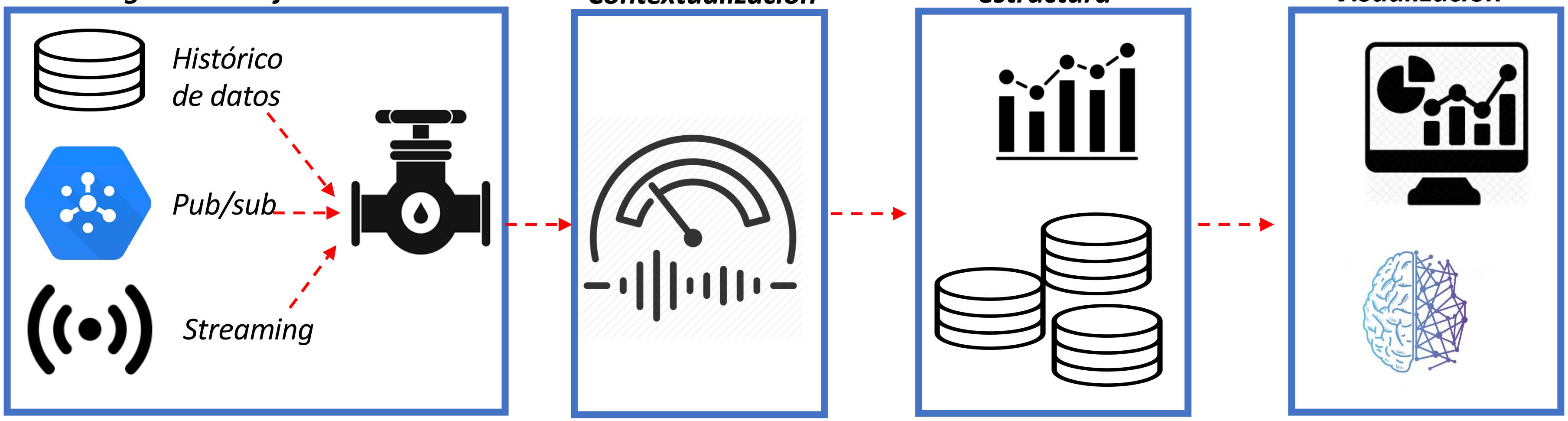
- Asume que no se conocen el resultado final.
- Entre los modelos destaca [Clustering] K-mens, FCM, FM; [Visualización y reducción de la dimensionalidad] PCA, etc.
- Muy efectivo para la mejora de procesos y calidad de producto.
- Complejo de implementar en data sets de alta dimensionalidad.

Clustering



AIoT building blocks → Traditional approach

Recogida & Cotejo | ***Limpieza & Contextualización*** | ***Optimización estructura*** | ***Análisis & Visualización***



Agregados de diferentes fuentes, estructuración y cotejo de datos según ventana temporal. http, MQTT, API rest.

Separar señales de ruido, limpieza, enriquecer y preparar datos de IoT. API notification

Almacenamiento de los datos procesados, análisis de series temporales y reutilización de datos en crudo. Elastic Search. In-memoria database

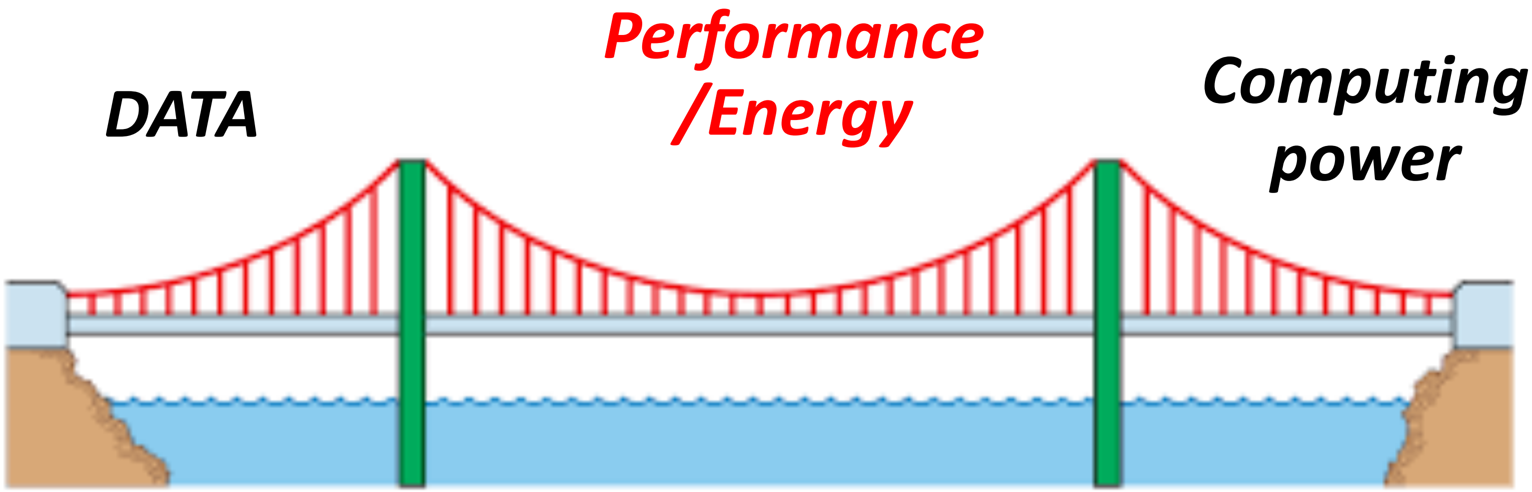
ML models training & inference. Sklearn, RAPIDS, Tensor Flow ….

GLOBALIoT

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

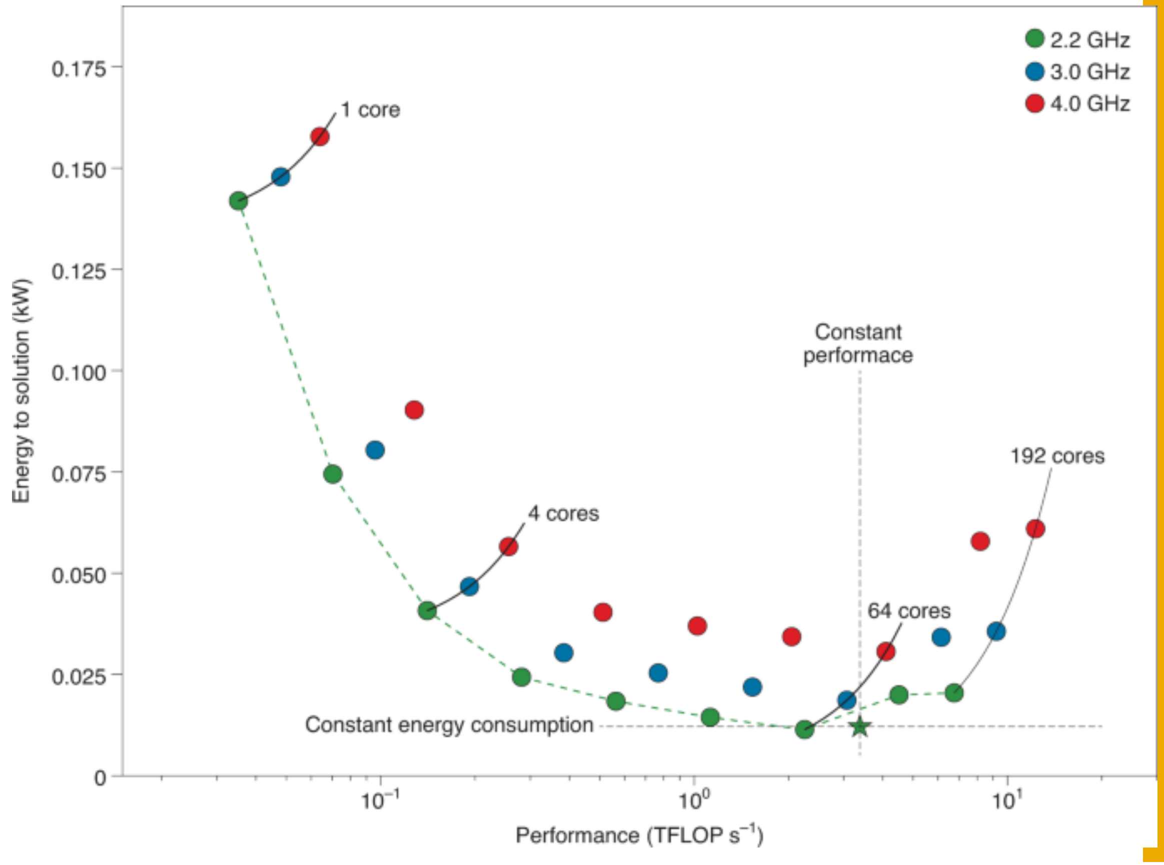
All that glitters is not gold: “*Performance and energy tradeoff* “

Data, Performance and Energy tradeoff



Ecological impact of HPC

Source: Zwart, S. P. (2020). The ecological impact of high-performance computing in astrophysics. *Nature Astronomy*, *4*(9), 819-822.



Bridging the gaps between AI and IoT

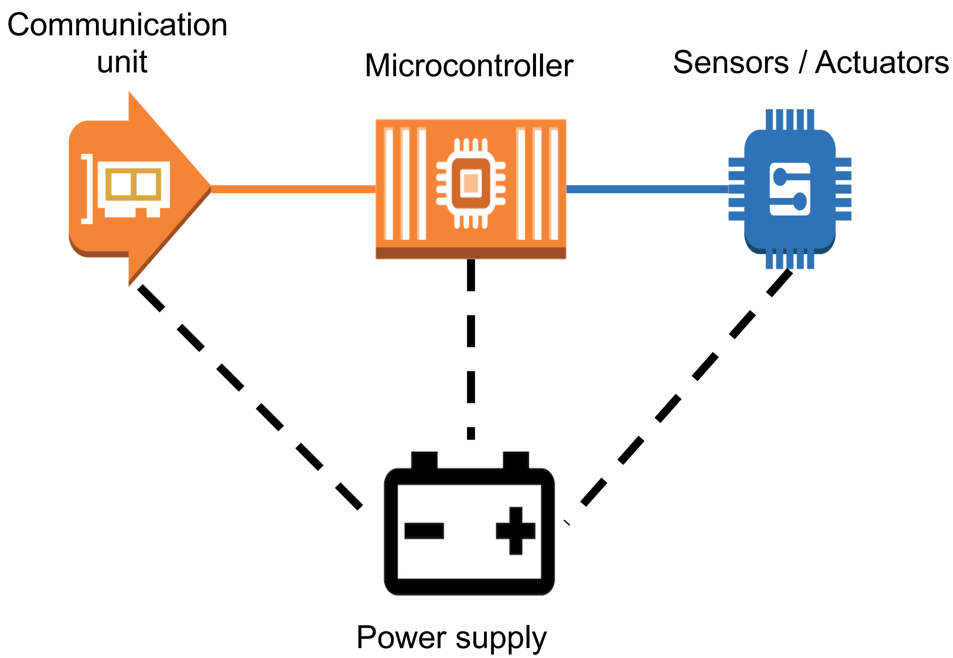
TinyML: Machine Learning meets the Internet of Things → *An energy efficient approach for IoT*

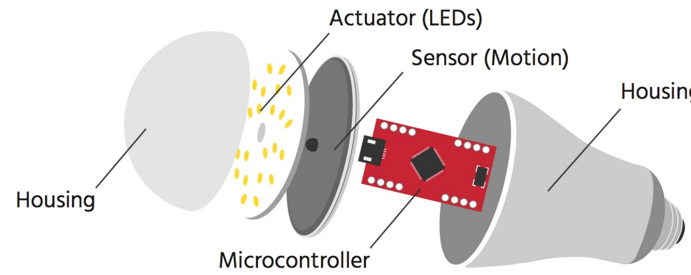
- ✓The ***Internet of Things*** (IoT) refers to a system of interrelated, Internet-connected objects ("*things*"), that can collect and transfer data.

- ✓***Machine Learning*** is typically associated with computationally heavy cloud-based solutions with relatively high latencies, high power consumption, and the need for high bandwidths links.

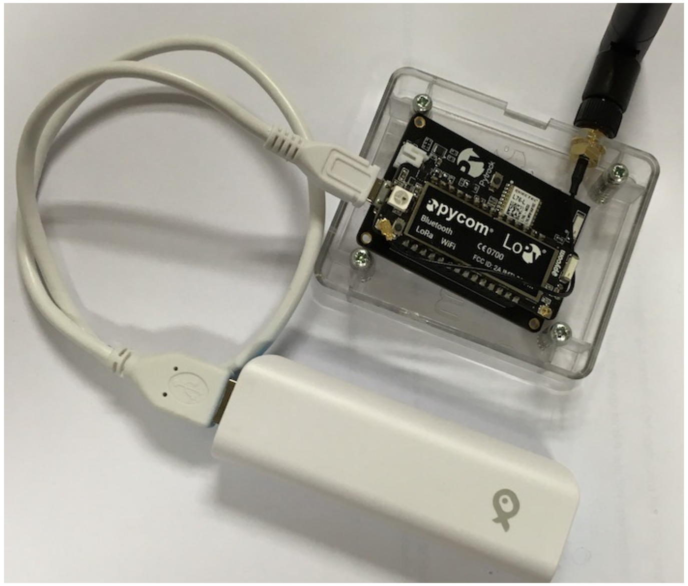
- ✓**With TinyML, IoT and ML come together by shrinking deep learning networks to fit on tiny hardware, thus requiring low energy, low bandwidth links, and reduced latencies.**

Things



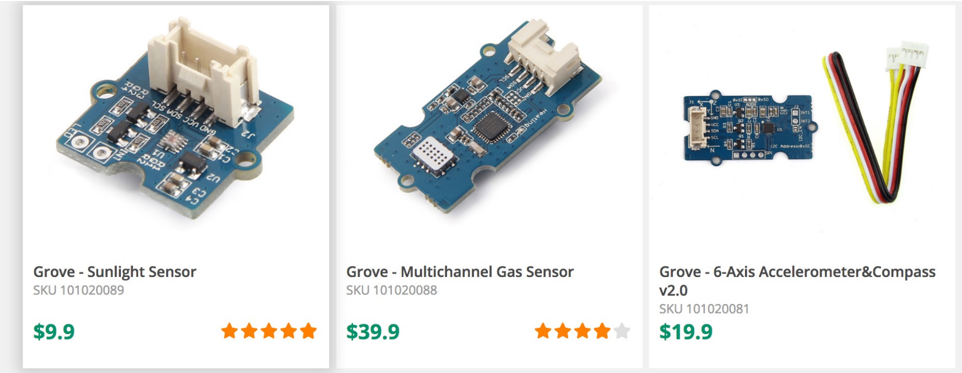


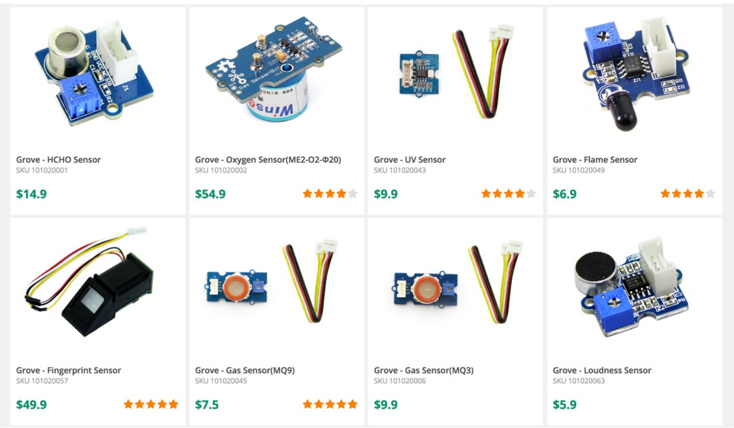
A Reference Guide to the Internet of Things Copyright © 2017 Bridgera LLC, RIoT



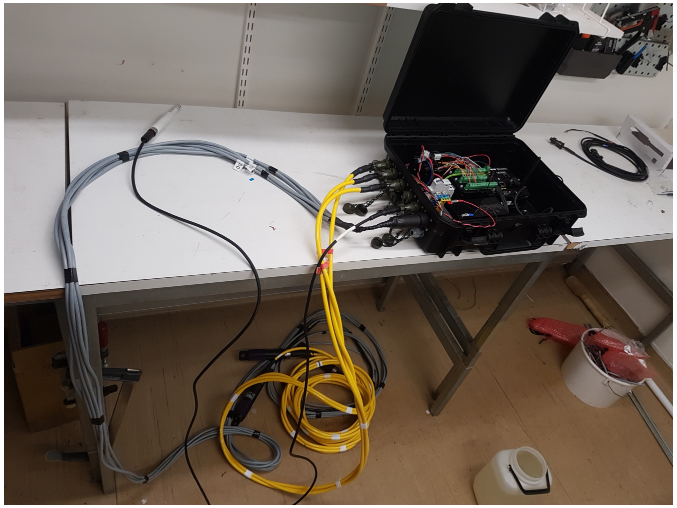
Sensors

cheap...

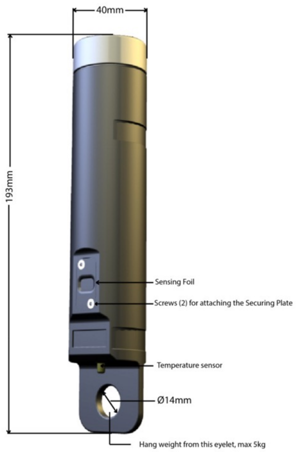




expensive...



1.2 **Oxygen Optode 4531 dimensions**



Parameter	Output	Default range[2)]	Calibrated range	Accuracy	Resolution
Oxygen Concentration	0 - 5V	0 to 800μM	0 to 500μM	<8μM or 5% whichever is greater	< 1μM
	4 - 20mA	0 to 800μM	0 to 500μM	<9μM or 5.2% whichever is greater	< 1μM
Oxygen Saturation	0 - 5V	0 – 200%	0 - 120%	<5 %	<0.4%
	4 - 20mA	0 – 200%	0 - 120%	<5.2 %	<0.4%
Temperature	0 - 5V	-5 to + 35°C	0 - 36°C	±0.1°C	±0.01°C
	4 - 20mA	-5 to + 35°C	0 - 36°C	±0.15°C	±0.02°C

Power sources

- Today, the ***most common power source is a battery***, but there are several other possibilities for power, such as **solar cells, piezoelectricity, radio-transmitted energy, and other forms of power scavenging.**
- Rechargeable batteries are not particularly well-suited to smart objects.
    - Instead of using rechargeable batteries, battery-equipped smart objects are typically designed so a single battery should last the entire lifetime of the smart object.

LiPo

LiPo

Li-ion

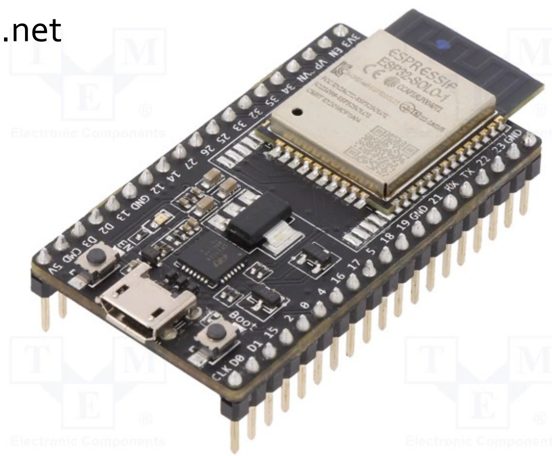
Lithium Ion (Li-ion)
Lithium-Ion Polymer (Li-Po)

Communication devices

- ✓ **Of the hardware components of a "thing", the radio is usually the most power-consuming component.**
  - ✓ 10X more expensive than…
- ✓ For low-power radios, only a small portion of the power consumption is used to send the radio signal into the air.
  - ✓ listening is as power consuming as sending.

https://www.espressif.com/en/products/socs/esp32

http://esp32.net



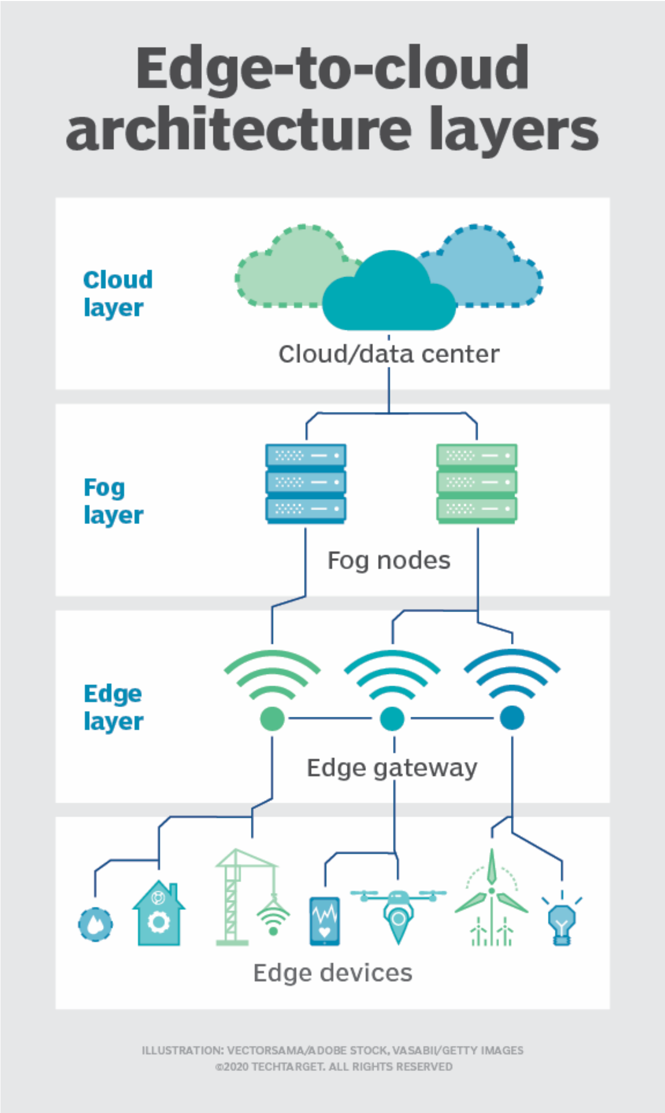
Mode	Min	Typ	Max	Unit
Transmit 802.11b, DSSS 1 Mbps, POUT = +19.5 dBm	-	240	-	mA
Transmit 802.11g, OFDM 54 Mbps, POUT = +16 dBm	-	190	-	mA
Transmit 802.11n, OFDM MCS7, POUT = +14 dBm	-	180	-	mA
Receive 802.11b/g/n	-	95 ~ 100	-	mA
Transmit BT/BLE, POUT = 0 dBm	-	130	-	mA
Receive BT/BLE	-	95 ~ 100	-	mA

Power mode	Description			Power consumption
Modem-sleep	The CPU is powered on.	240 MHz *	Dual-core chip(s)	30 mA ~ 68 mA
			Single-core chip(s)	N/A
		160 MHz *	Dual-core chip(s)	27 mA ~ 44 mA
			Single-core chip(s)	27 mA ~ 34 mA
		Normal speed: 80 MHz	Dual-core chip(s)	20 mA ~ 31 mA
			Single-core chip(s)	20 mA ~ 25 mA
Light-sleep	-			0.8 mA
Deep-sleep	The ULP coprocessor is powered on.			150 μA
	ULP sensor-monitored pattern			100 μA @1% duty
	RTC timer + RTC memory			10 μA
Hibernation	RTC timer only			5 μA
Power off	CHIP_PU is set to low level, the chip is powered off.			1 μA

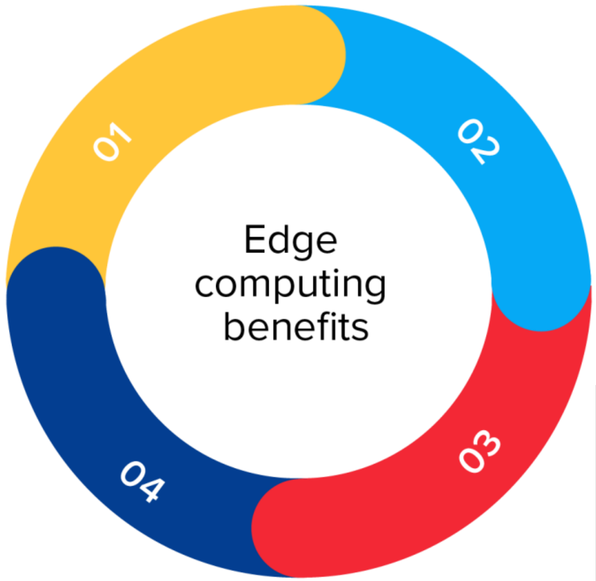
What does that all mean?

So … Edge computing

✓A possible definition of ***edge computing*** is: “a part of a distributed computing topology in which information processing is located close to the edge — where things and people produce or consume that information.”



Advantages of Edge Computing... for IoT



Latency
Reduction of latency by processing the data closer to the customer

Bandwidth
Sending data from edge to the cloud takes up spectral resources; there's just not enough bandwidth for data transportation

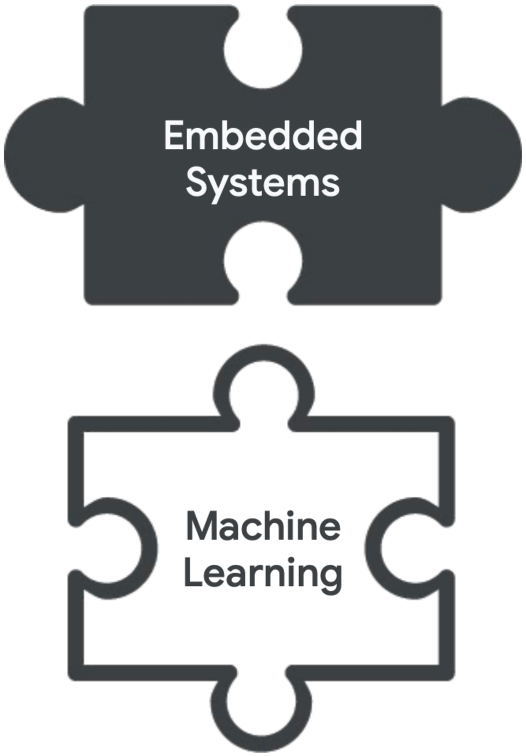
Reliability
By processing data at the edge, you eliminate network reliability problems

Security → Privacy
Computing at the edge provides more security than computing at the cloud because it is less vulnerable to numerous variety of threats due to its scope

https://www.wwt.com/article/show-me-the-money-drive-new-revenue-streams-with-edge-computing

So, what is TinyML?

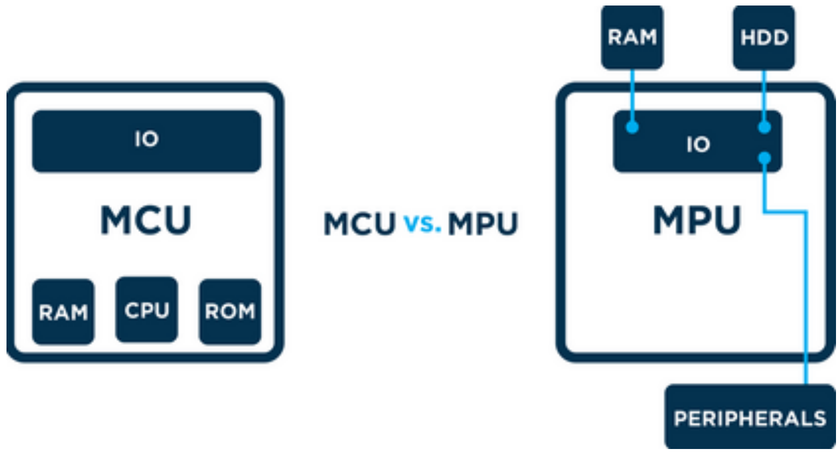
Original definition: "…a neural network model that runs at an energy cost of below 1 mW."
© "TinyML" by Pete Warden, Daniel Situnayake



"Tiny machine learning (TinyML) is a fast-growing field of machine learning technologies and applications including algorithms, hardware, and software capable of performing on-device sensor data analytics at extremely low power consumption, **typically in the mW range and below**, enabling a variety of always-on ML use-cases on battery-operated devices."

Microcontroller

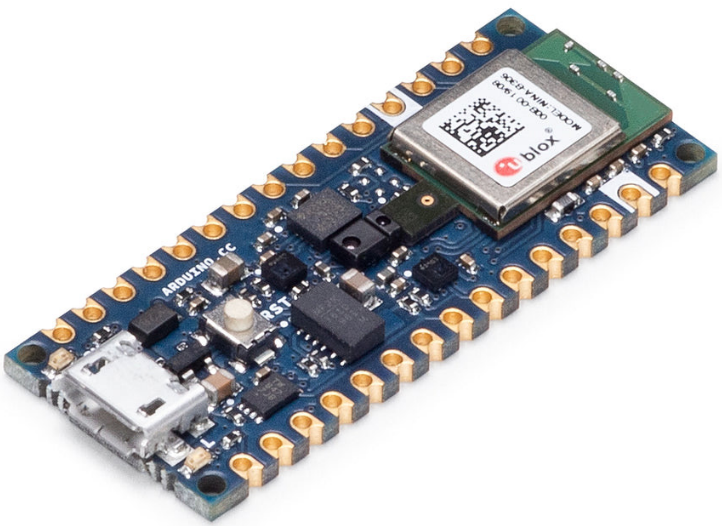
- ✓ A microcontroller is a microprocessor with built-in memory, timers, and hardware for connecting external devices such as sensors, actuators, and radio transceivers.
- ✓ Typically, a smart object microcontroller has a few kilobytes of on-chip memory and is run at a clock speed of a few megahertz.



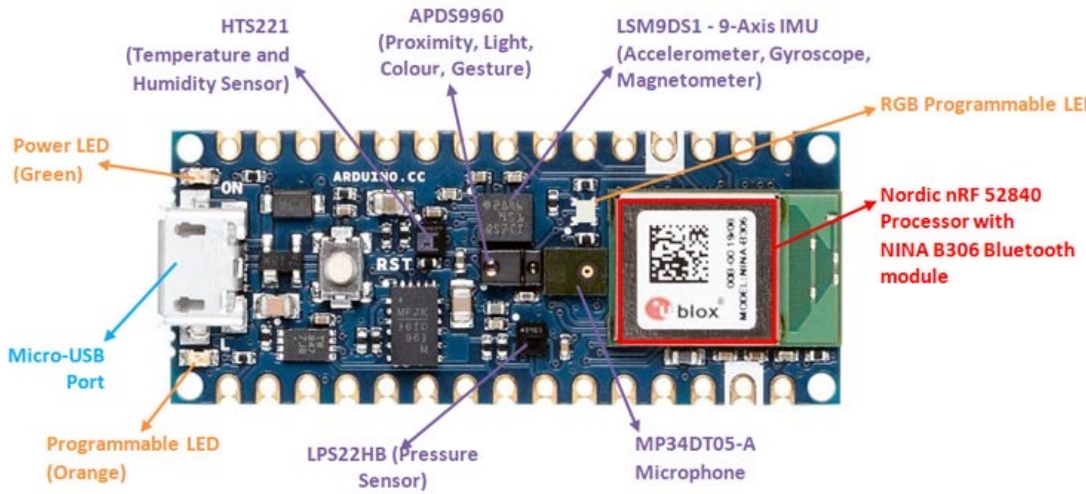
Orders of magnitude

	Microprocessor	>	Microcontroller
Platform			
Compute	1GHz–4GHz	**~10X**	1MHz–400MHz
Memory	512MB–64GB	**~10000X**	2KB–512KB
Storage	64GB–4TB	**~100000X**	32KB–2MB
Power	30W–100W	**~1000X**	150µW–23.5mW

TinyML Hardware



Arduino Nano 33 BLE Sense
64 MHz **Arm® Cortex-M4**F (with FPU)
1 MB Flash + 256 KB RAM
Bluetooth® 5 multiprotocol radio



…but edge is low-power and low-performance

Nvidia Jetson AGX Xavier:
- K$
- 10W-15W
- 32 TOPS
- 136.5GB/sec

Coral Google
- 150 $
- 5-10 W
- 4TOPS

Nvidia Jetson Nano
- 100$
- 5-10W
- 472GFLOPS
- 25.6GB/s

- ✓ Low Power and Low Performance.
- ✓ Is it enough? -> No, at the moment.
- ✓ Virtualization tech. may help.

Arduino
- 10$
- 1W

What about Nvidia Jetson Family?

	Jetson Nano	Jetson TX2	Jetson AGX Xavier	Jetson AGX Orin 32 GB	Jetson AGX Orin 64 GB
Rendimiento IA	472 GFLOPS	1,33 TFLOPS	32 TOPS	200 TOPS	275 TOPS
GPU	Maxwell 128 nucleos	Pascal 256 núcleos	Volta, 512 núcleos + 64 Tensor cores	Ampere 1792 núcleos + 56 Tensor Cores	Ampere 2048 núcleos + 64 Tensor Cores
CPU	4 Cores ARM A57	2 Cores Denver 2 ARM A57	8 cores Carmel ARM	8 cores ARM Cortex A79AE	12 cores ARM Cortex A79AE
Memoria	LPDDR4 4GB (25.6 GB/sec)	LPDDR4 8GB (59.7 GB/sec)	LPDDR4 32GB (136.5 GB/sec)	LPDDR5 32 GB (204.8GB/sec)	LPDDR5 64 GB (204.8GB/sec)
Power	5-10W	7.5W-20W	10W-40W	15W-50W	15W-60W

Source: Nvidia.com

A real-time UAV surveillance system for natural disaster management

The climate is changing.

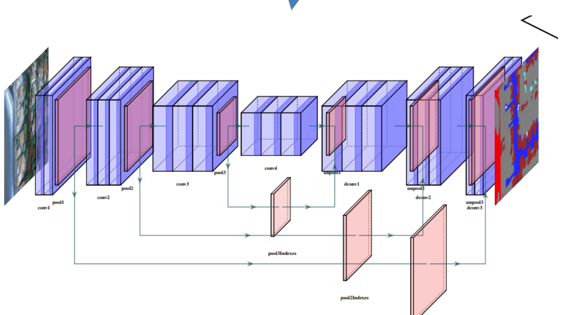
Imagenes cedidas por el ayuntamiento de San Javier (Murcia)

Phases of disaster management

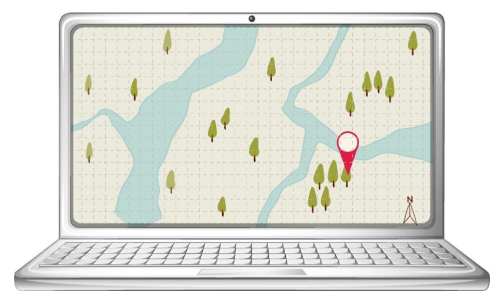
Autonomous AI swarm drone to help decision makers

Disaster management: an overview

UAV
swarm



Computer vision
on the edge
device



Management platform

Aerial Surveillance of unknown area

HERNÁNDEZ, Daniel, et al. AI-enabled autonomous drones for fast climate change crisis assessment. IEEE Internet of Things Journal, 2021.

1. AI based pipeline to reduce the amount of information to be monitored
2. Image clustering with unknown number of centroids
3. End-to-end operation on Jetson Nano

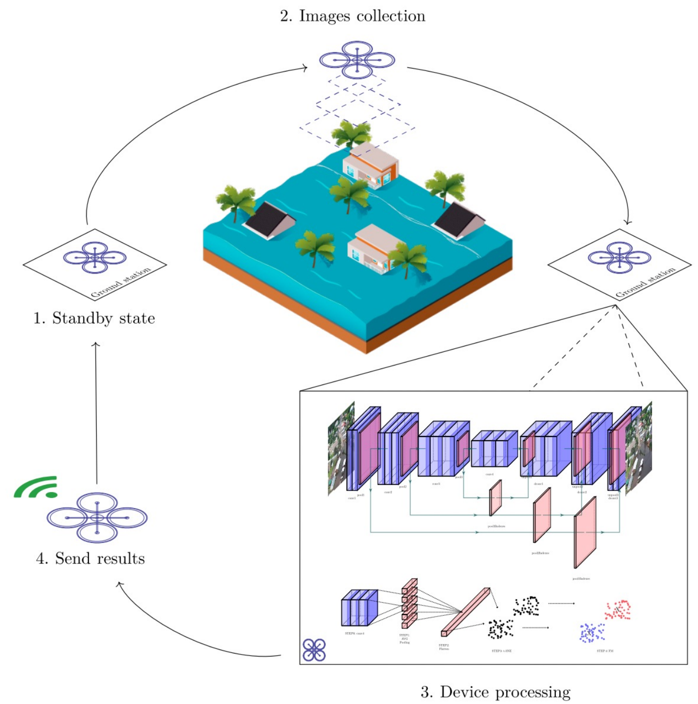


Fig. 3: Natural disasters management overview.

Reducing the number of images…

- … To be processed by humans → Early response and decision making
- ... to decrease the communication bandwidth requirments

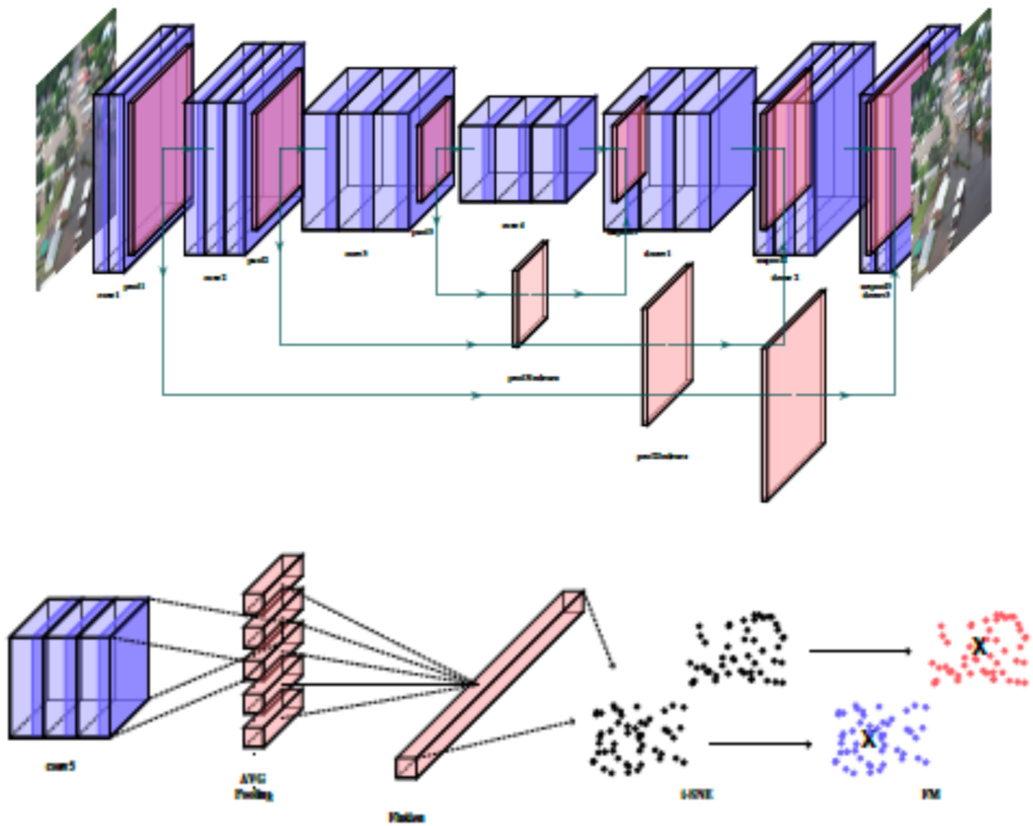


Fig. 2: AI-pipeline for the latent space clustering extracted from the autoencoder.

Evaluation

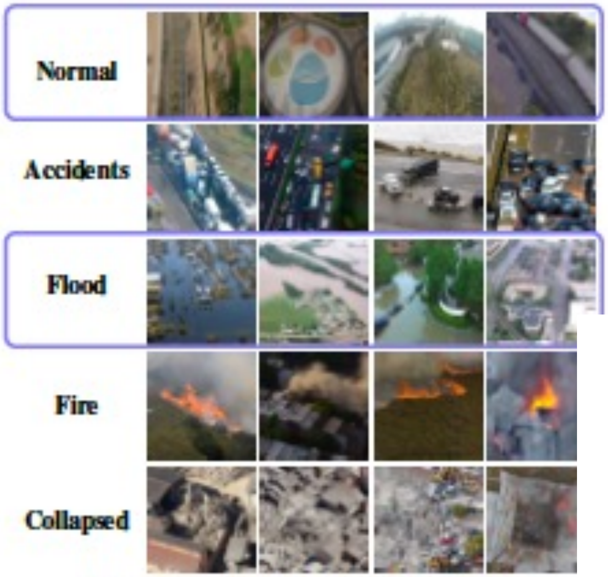
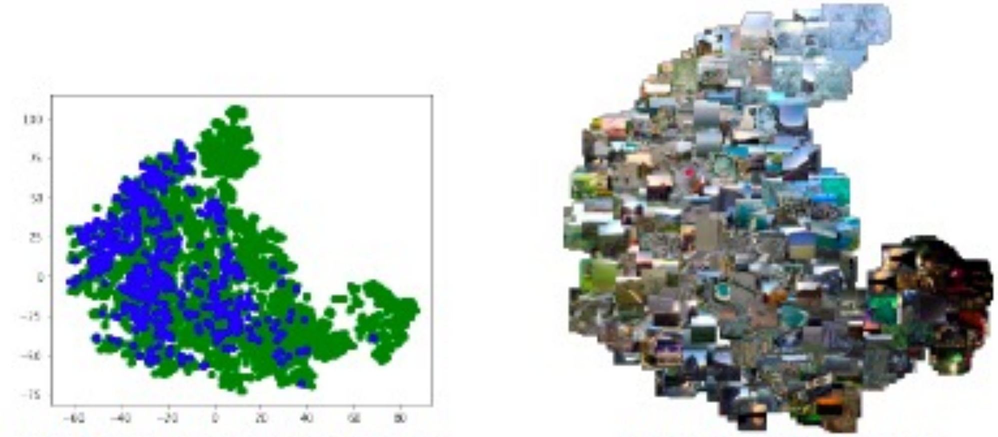


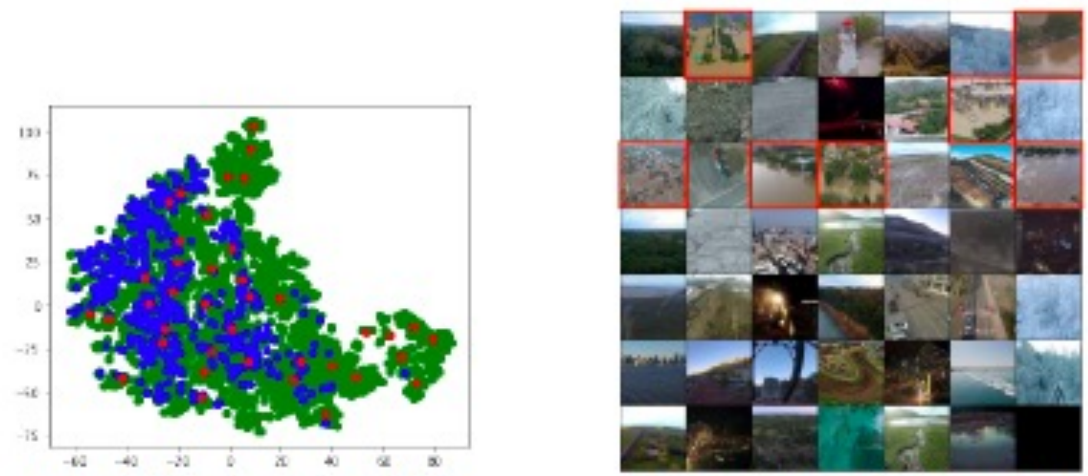
Fig. 3: Classes within the AIDER dataset.



(a) Data points for flooding images (blue), and others (green)

(b) Images corresponding to data points.

Fig. 4: Data points and images after running the t-SNE algorithm.



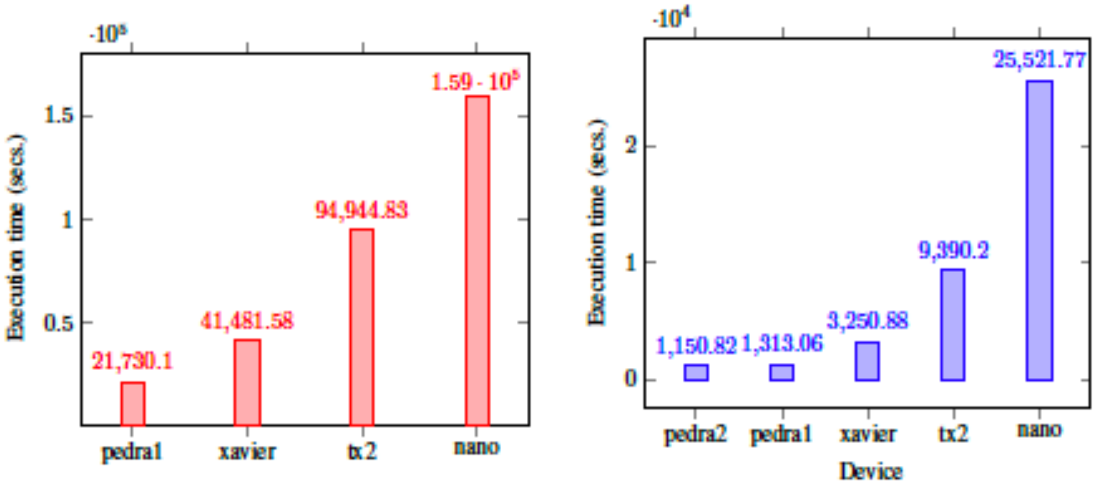
(a) Data points of flooding images (blue), others (green), and prototypes obtained by the FM algorithm (red).

(b) Images closer to the prototype found by FM.

Fig. 5: Data points and images after running the FM algorithm.

TABLE I: Specification of the various GPU platforms used in our experiments.

	Pedra	Jetson AGX Xavier	Jetson TX2	Jetson Nano
CPU	Intel Silver 4216	NVIDIA Carmel ARM v8.2	ARMv8	ARM Cortex-A57 MPcore
2xGPU (NVIDIA)	GeForce RTX 2080 Ti	Volta	Pascal	Maxwell
Memory [Gb]	12 DDR5	32 LPDDR4x	8 LPDDR4	4 LPDDR4
Size [mm]	N/A	105 x 105	50 x 87	70 x 45
Weight [g]	N/A	280	85	61
Energy consumption [W]	N/A	10-30	7.5	3-5



(a) CPU training time (30 epochs).

(b) GPU training time (30 epochs).

Fig. 7: Execution time for the auto-encoder's training stage.

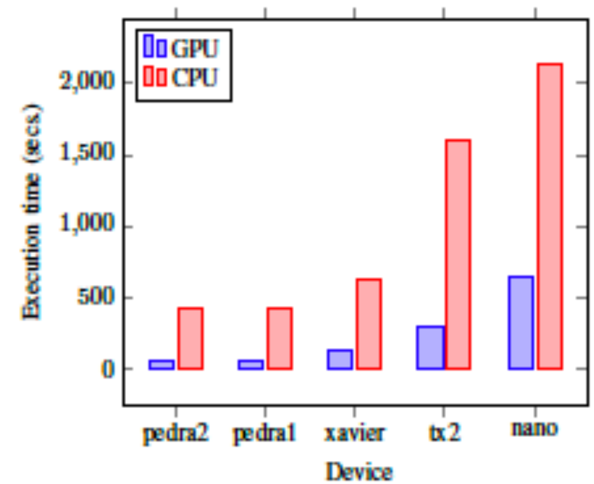


Fig. 8: Inference comparison of GPU/CPU for the entire dataset.

Performance evaluation

Flood Detection Using Real-Time Image Segmentation

HERNÁNDEZ, Daniel, et al. Flood Detection Using Real-Time Image Segmentation from Unmanned Aerial Vehicles on Edge-Computing Platform. Remote Sensing, 2022, vol. 14, no 1, p. 223.

1. Real time image capture and mask generation
2. Send Mask & GPS position to the management platform
3. Show data in a heat map

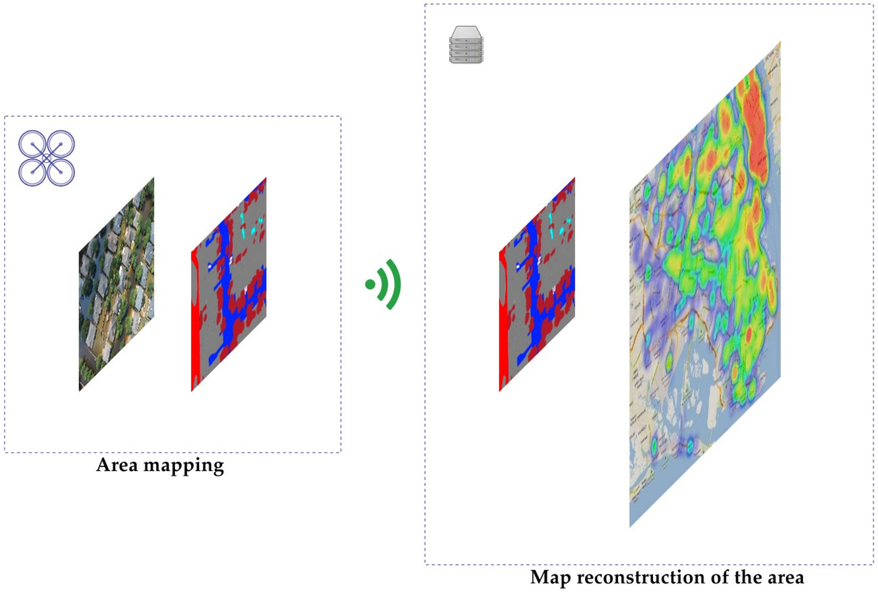


Figure 1. Overall view of our proposal.

Flood Detection Using Real-Time Image Segmentation

1. Analysis of state-of-the-art models for image segmentation in terms of performance, accuracy, and memory footprint. Particularly, PSPNet, DeepLabV3, and U-Net

2. A semi supervised training procedure with a pseudo-labeling strategy is designed to increase the accuracy

3. End-to-end operation on Jetson Nano

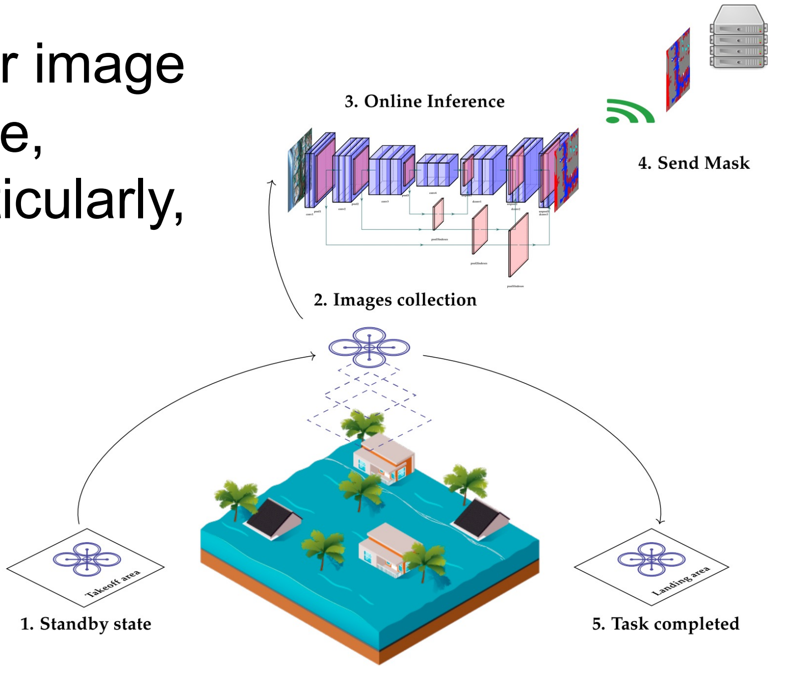


Figure 3. Overview of our image collection and processing solution.

Flood Detection Using Real-Time Image Segmentation

Experiment on FloodNet dataset

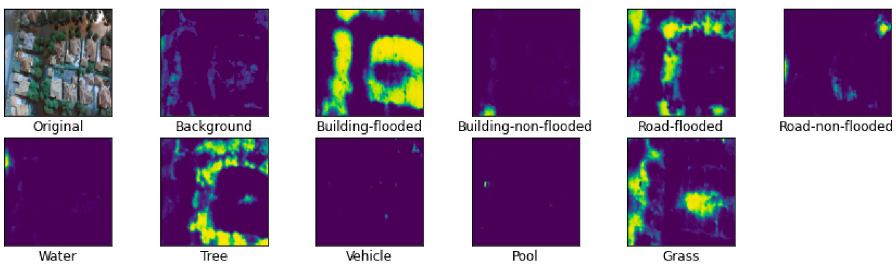


Figure 5. Segmentation mapping of each of the classes on the original photo.

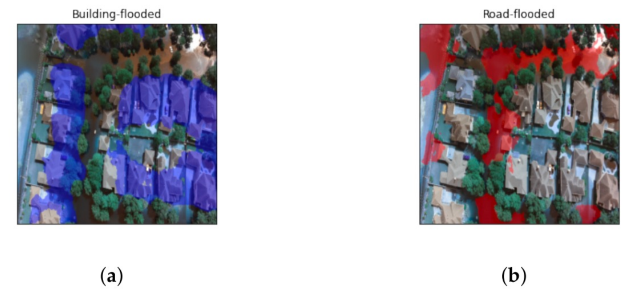
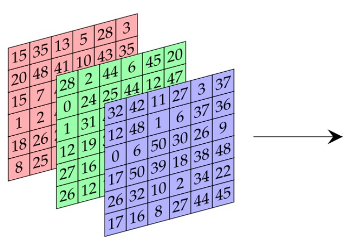


Figure 6. Segmentation mask for the classes of interest. (**a**) Blue mask over flooded buildings; (**b**) Red mask over flooded roads.



1	1	7	7	4	2
6	1	3	3	0	3
1	3	7	7	7	6
3	7	5	7	5	7
1	6	0	5	3	2
5	2	7	7	1	6

Figure 14. Compression comparison between an RGB image and its segmentation mask.

Accuracy/Size

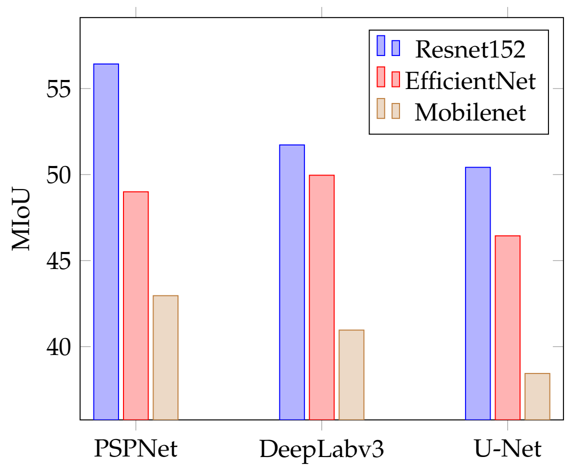


Figure 7. Accuracy (MIoU) of three neural networks and encoders used, having been trained with the baseline dataset.

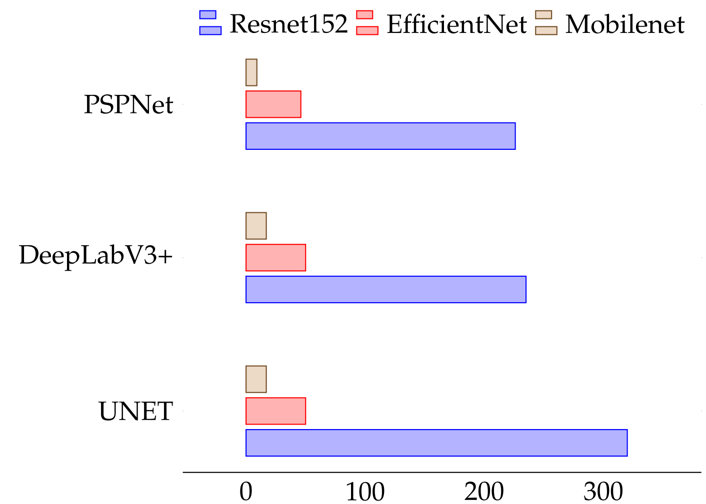


Figure 9. Size (in MB) of the output of every model, with every encoder.

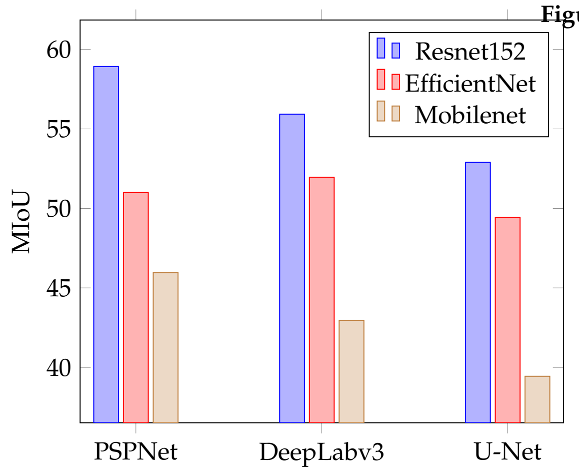


Figure 8. Accuracy (MIoU) of three neural networks and encoders used, having been trained with the pseudolabeling technique described in Section 2.6.

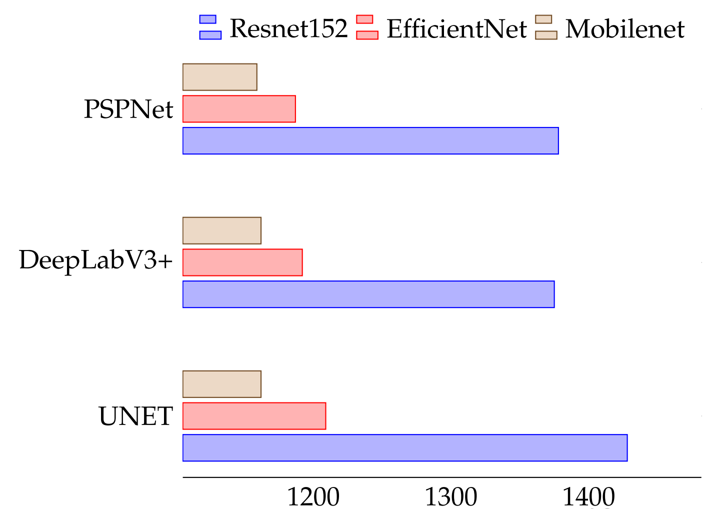


Figure 10. Memory footprint (in MB) for every model, and with every encoder.

Performance

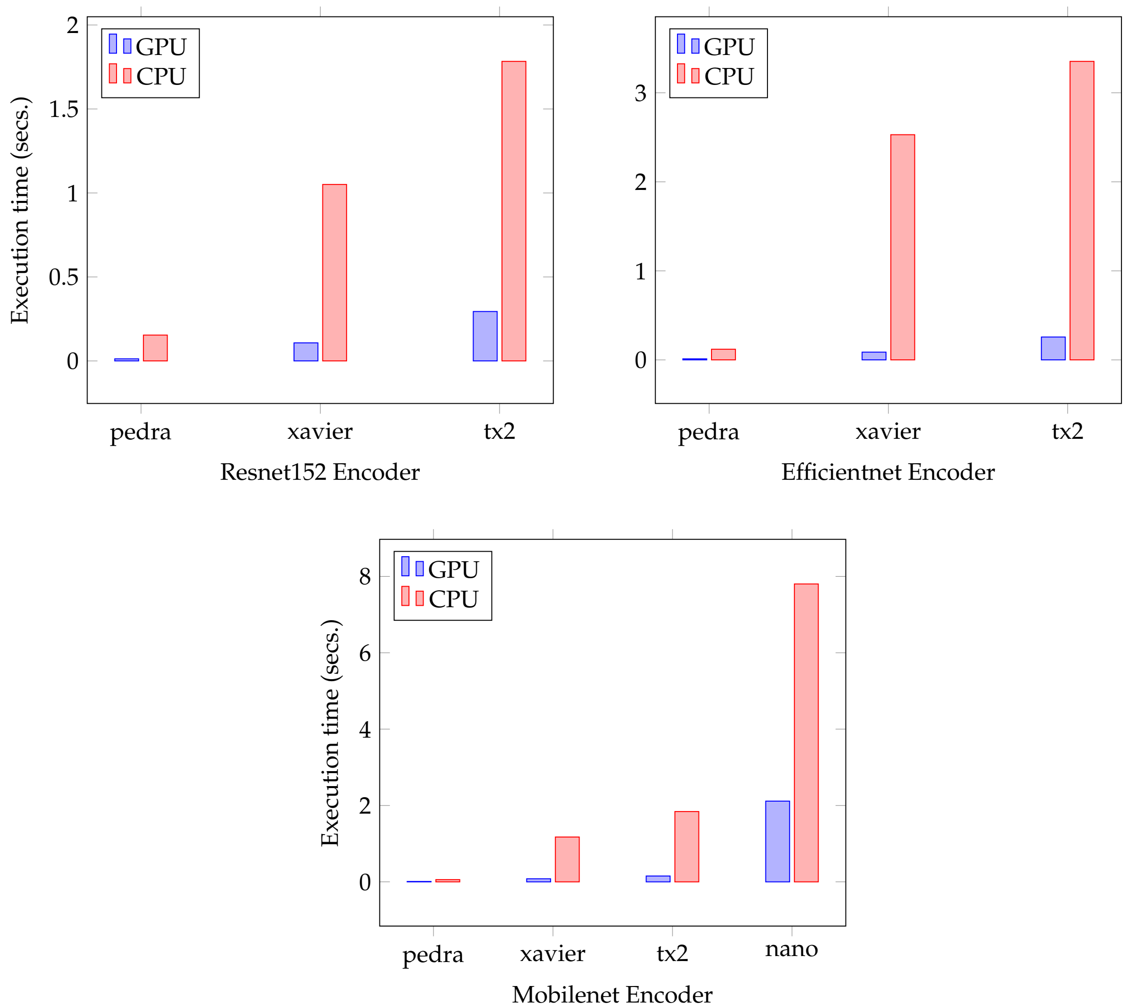


Figure 11. PSPNet architecture inference time comparative with all encoders.

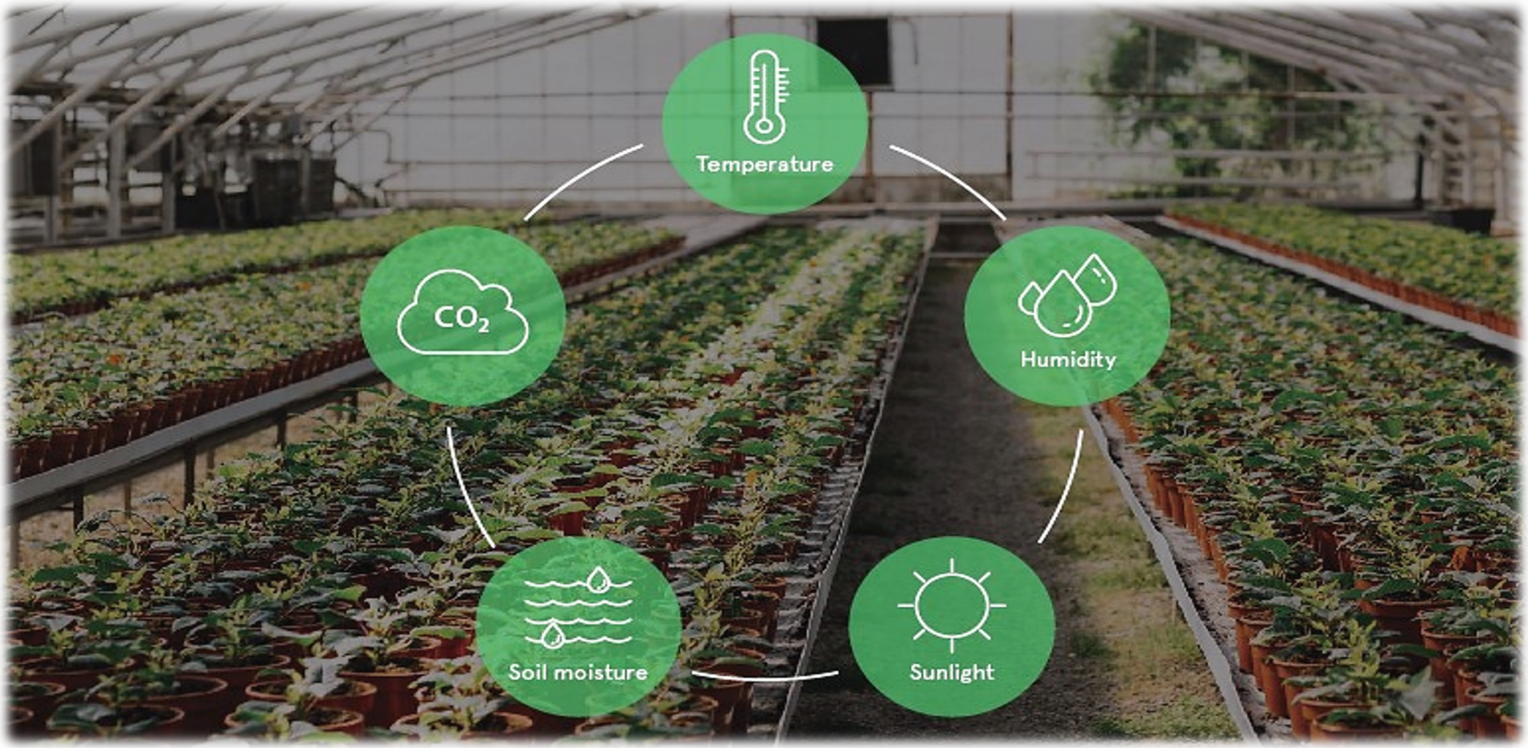
Conclusions

- ✓ UAVs have the potential to play a "key role" at mitigating the consequences of climate change.
- ✓ However, both hardware and software solutions are needed to make these devices truly crucial players in these tasks.
- ✓ AI and edge computing are undoubtedly a winning combination by enabling to transform autonomous drones into useful tools in various emergency situations.
- ✓ Our results reveal that the use of neural networks designed for real-time image segmentation from drones can be a viable solution as long as the drone is equipped with an edge computing device endowed with GPUs.
- ✓ The benefit of merely sending the result mask instead of the raw image, which is made possible by performing image processing at the same location where the image is captured, reduces the required network traffic by several orders of magnitude.
- ✓ It is worth mentioning that the computational load differences between edge and cloud platforms remain large, with speedup factors in the range of 2.8x-22.17x.

Time series analysis for temperature forecasting using TinyML

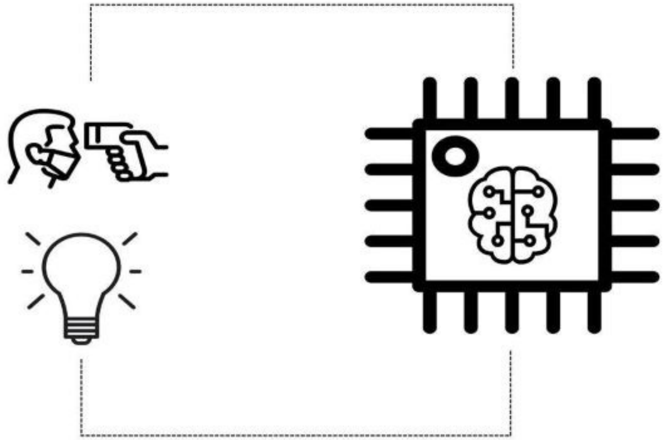
Introduction

- **Climate control system:** automation of climate inside greenhouses
- **Optimisation:** to improve temperature management



Objetives

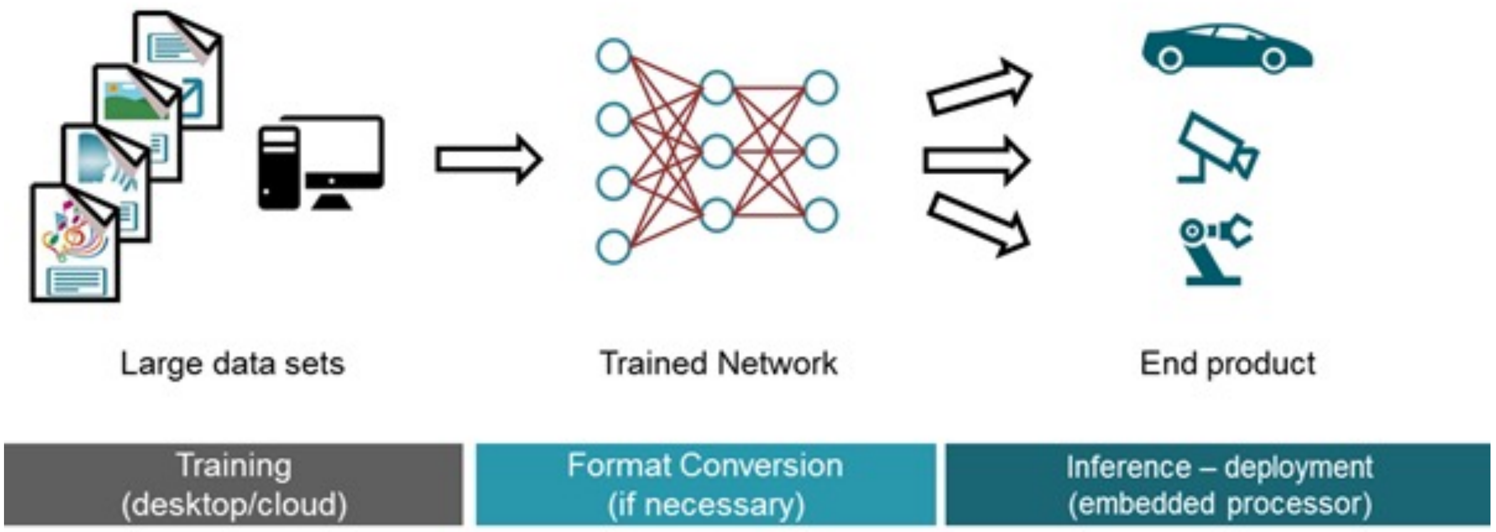
- **Task**: define a neural network for the prediction of time series.
- **Execution**: run the neural network on a microcontroller.
- **Results**: Evaluate the performance of the network and compare the results in terms of execution time and energy consumption



IoT Device – AI in the Microcontroller

TinyML – Workflow

1. **Step 1:** Train the model locally/remote
2. **Step 2:** Convert the model
3. **Step 3:** Transfer the model to the microcontroller



TinyML - Tools

1. **Step 1: Google Colab**

Platform for running code on the Cloud, in the form of Jupyter Notebook

1. **Step 2: Tensorflow Lite**

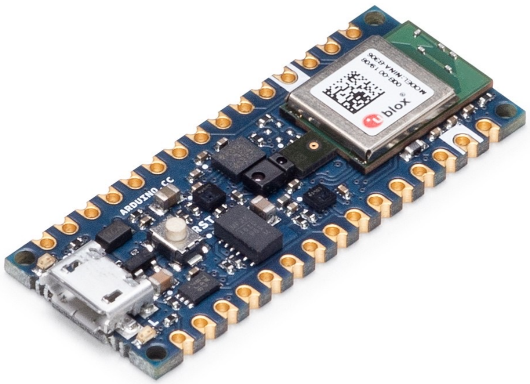
Library enabling the conversion of Machine Learning models

1. **Step 3: Arduino IDE**

Arduino development environment

TinyML - Hardware

Arduino Nano 33 BLE Sense



- CPU 32-bit ARM® Cortex™-m4
- Clock Speed 64mhz
- Cpu Flash Memory 1mb
- Sram 256kb
- Sensors: Gesture, Light, Proximity, Barometric Pressure Ecc..
- Bluetooth BLE

Raspberry Pi 4 Model B

- CPU Quad Core Broadcom BCM2711B0 (Cortex A-72)
- Clock Speed 1.5 Ghz
- Ram 8GB
- Uscita Video HDMI, 2x USB 3.0 / 2x USB 2.0
- Bluetooth 5.0

Implementation - Dataset

- **Api Endpoint:**
  - Metodo: GET
  - Parametri: sensor, fromDate, toDate

Figura 1 I GLOBALoT prototype greenhouse project

Figure 2 Optimus climate control and fertigation system of the project

Implementation - Dataset

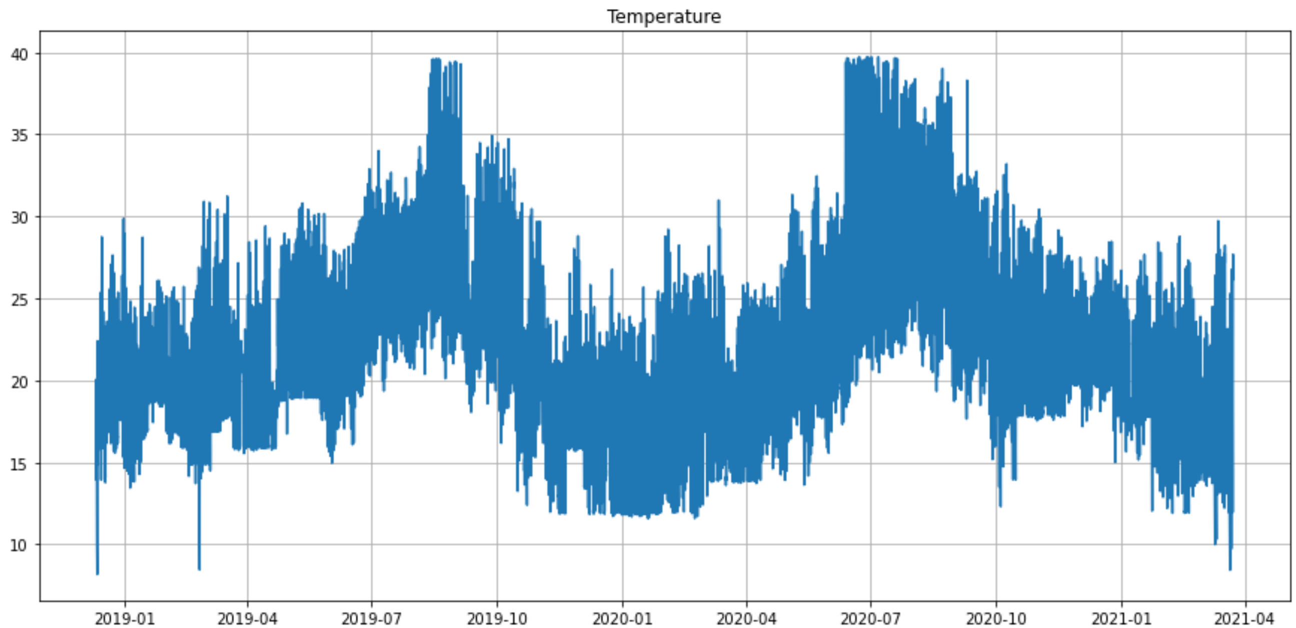
- **Preprocessing:**
  - Removing incorrect data
  - Removal of outliers
  - Timestamp fixed

Dataset 1

timestamp	value
2018-12-11 10:30:00	13.58
2018-12-11 10:45:00	14.35
2018-12-11 11:00:00	15.64
2018-12-11 11:15:00	16.58
2018-12-11 11:30:00	15.36

Dataset 2

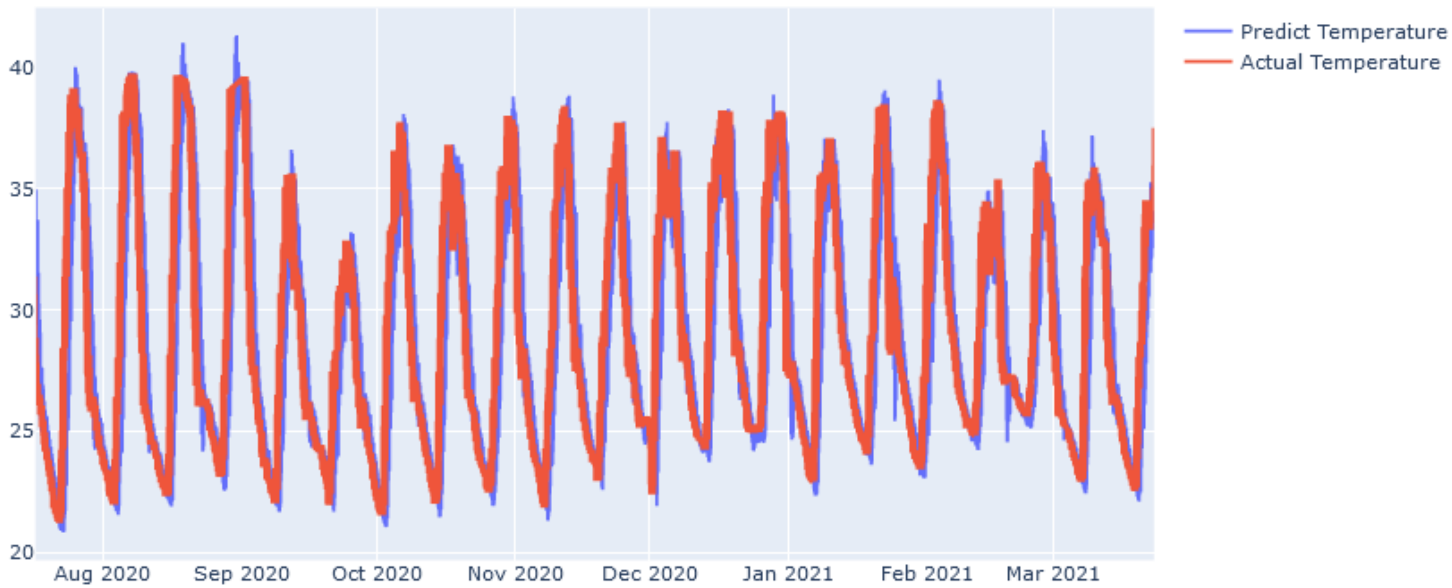
timestamp	value
2018-12-11 10:00:00	13.96
2018-12-11 11:00:00	15.92
2018-12-11 12:00:00	18.05
2018-12-11 13:00:00	20.07
2018-12-11 14:00:00	19.56



Implementation - Multilayer perceptron

Parameters:

- Optimizer: adam
- Loss: mean squared error
- Learning rate: 10^{-6}
- Epoche: 100/200



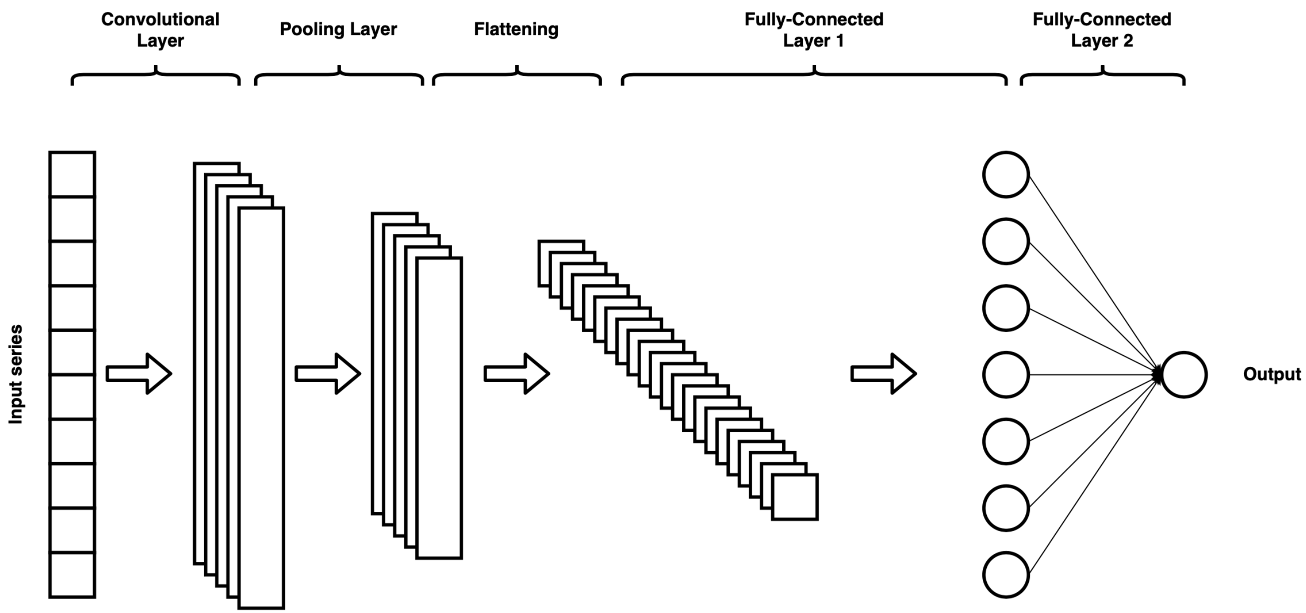
Implementation – Convolutional Neural Network

Levels of the convolutional neural network:

- Two convolution levels of 16 and 32 filters
- MaxPooling level with pool size 2
- Flatten level
- Two dense levels with 100 and 3 or 12 nodes

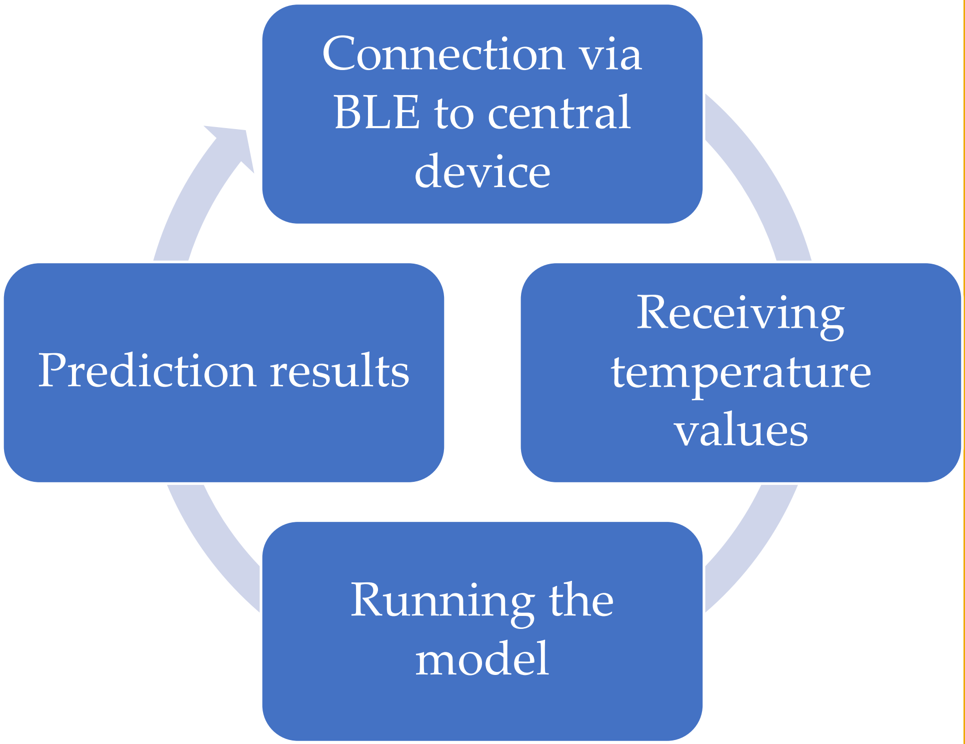
Parameters

- Optimizer: adam
- Loss: mean squared error
- Epoche: 20/50



Implementation - Model execution on Arduino

- Importing libraries
- Declare variables
- Metodo setup()
- Metodo loop()



Results Analysis - Performance Models

Root Mean Squared Error

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}$$

Mean Absolute Error

$$MAE = \frac{1}{N} \sum_{i=1}^{N} |y_i - \hat{y}_i|$$

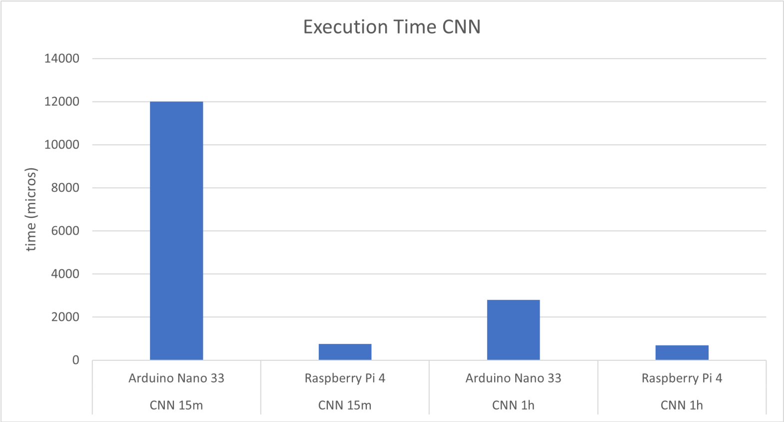
R squared

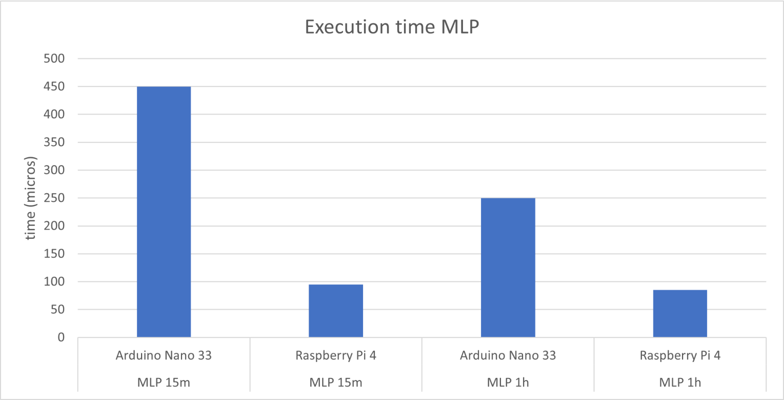
$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

Modello	RMSE	MAE	R2
CNN 15m	2,120	1,298	0,842
CNN 1h	2,148	1,355	0,837
MLP 15m	2,349	1,463	0,806
MLP 1h	2,851	1,993	0,713

Analysis of results - Execution time

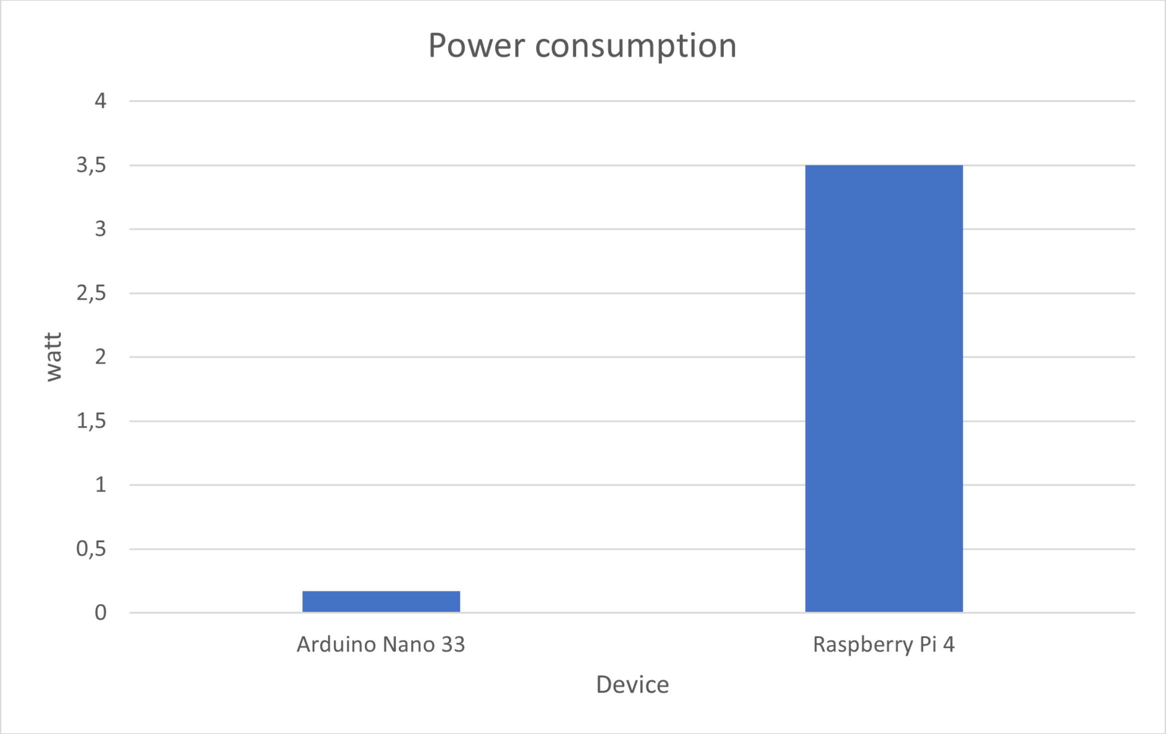
- Execution time to to carry out a single forecast (microseconds)





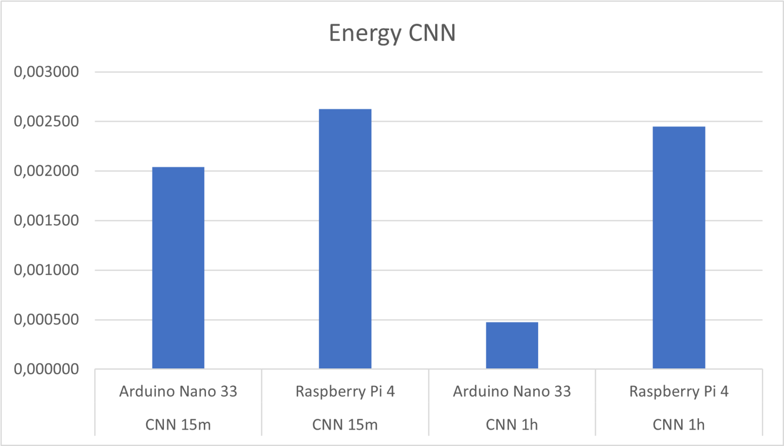
Analysis of results - Power

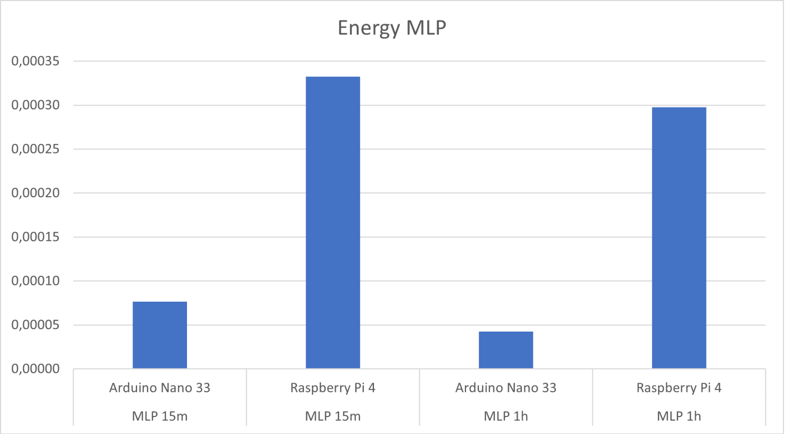
- The difference in terms of consumption between the two devices is very high.
- The Raspberry board consumes 3.5 W while the Arduino consumes only 0.17 W.



Analysis of results - Energy

- Energy was calculated by multiplying the run time by the power output.
- It is used to understand the best model-device combination in the trade-off between runtime and power.





Conclusions and Future work

The study carried out allowed the implementation of Machine Learning techniques on IoT devices in the context of temperature automation inside greenhouses.

The solution obtained using the Arduino device was very economical and efficient.

TinyML is a very promising ML technique that is applicable in multiple contexts.

The findings highlighted the benefits, but also the challenges of carrying out such projects.

Future developments: use of higher performance devices to further compare energy consumption between different solutions.

GOBIERNO DE ESPAÑA

MINISTERIO DE CIENCIA, INNOVACIÓN Y UNIVERSIDADES

AGENCIA ESTATAL DE INVESTIGACIÓN

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

GLOBALIOT

http://www.grc.upv.es/

Departamento DISCA. Edificio 1G.

Camí de Vera S/N C.P. 46022.

Valencia

José María Cecilia

jmcecilia@disca.upv.es

SIE 30 Años