

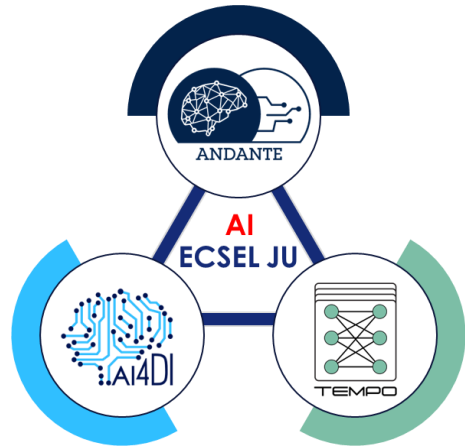


International Workshop on Embedded Artificial Intelligence Devices, Systems, and Industrial Applications (EAI)



Milan, Italy 19 September 2022

International Workshop on Embedded Artificial Intelligence Devices, Systems, and Industrial Applications (EAI)



Architecting Edge AI Workflows for Predictive Maintenance in Industrial Applications

Ovidiu Vermesan, SINTEF, Norway



19 September 2022 Milan, Italy

Presentation Outline



- Introduction
- Background
- Sensor-driven PdM
- Experimental Architecture
- Flexible Model Development Workflow
- Automation of Various Parts of the E2E Workflow
- Pre-analysis in Time and Frequency Domain
- Benchmarking and Testing
- Summary and Future Work

Introduction



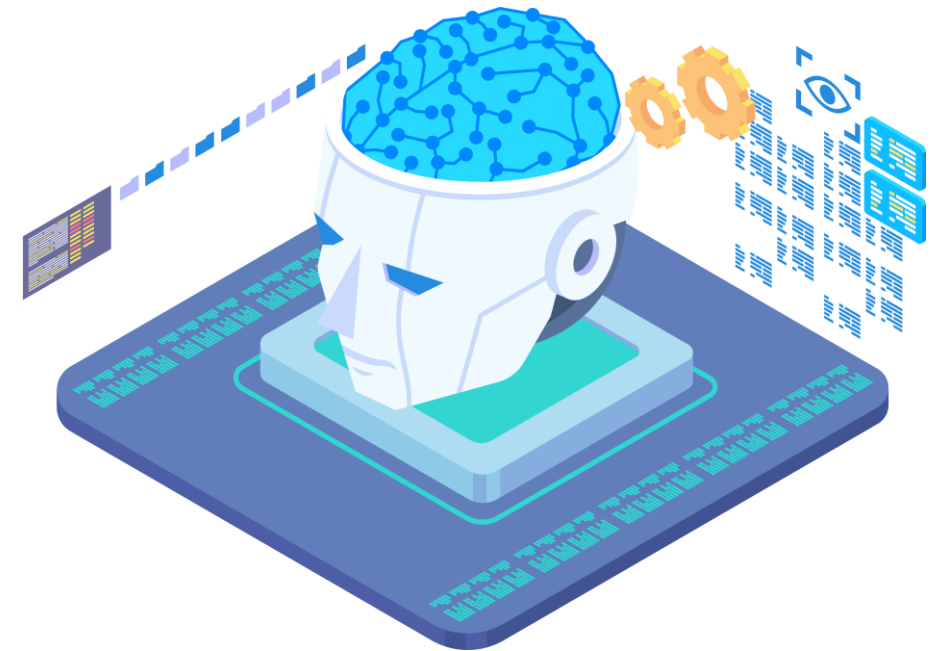
- ❖ Predictive maintenance (PdM) is the procedure industrial enterprises use to predict future failure points and monitor an asset's condition in real-time.
- ❖ The PdM technique leverages ML algorithms that take critical historical data, such as vibration, temperature, and sound, as an input, thus providing anomaly detection, classification and prediction related to the condition of an asset in real-time.

- ❖ PdM enables enterprises to significantly reduce unplanned machine/motor downtime and decide whether any respective motor/equipment needs maintenance.
- ❖ PdM ensures the machine is taken for maintenance before it fails, ensuring minimal losses in production.

- ❖ PdM solutions leverage technologies such as Artificial Intelligence (AI), the Internet of Things (IoT), and edge processing to gather meaningful insights from all the data received from the industrial equipment/motors, thus helping take necessary actions before the asset breakdown.

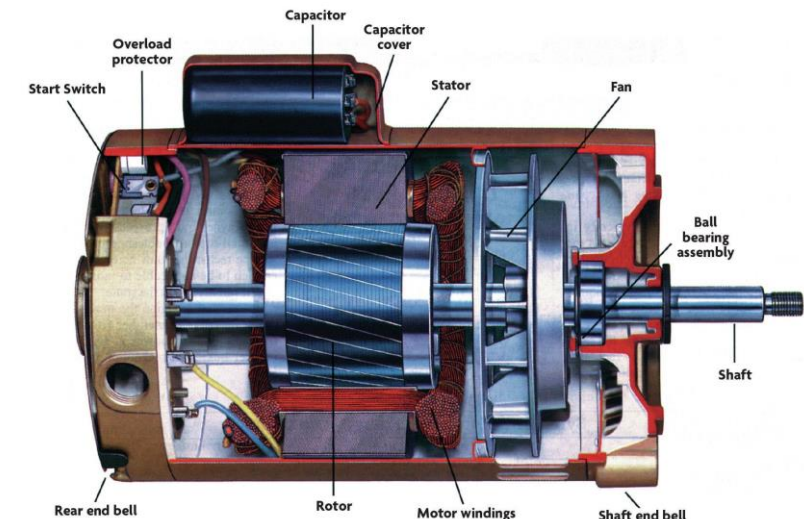
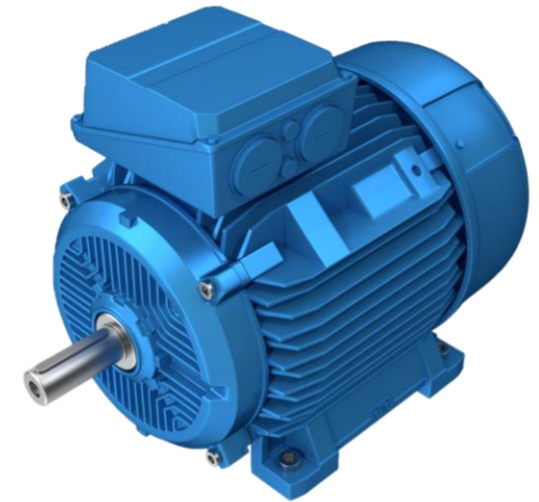
Background

- When processed and analysed intelligently, the data from edge devices provides valuable information/knowledge about manufacturing process, production system and equipment/motor.
- AI/ML methods are tools in PdM applications to develop solutions to prevent failures in equipment/motors operating in the industrial production lines.
- The performance of PdM applications depends on many factors, such as the appropriate choice of AI/ML platforms.
- The selection of the AI frameworks/platforms employed for edge AI machine learning/deep learning implementations largely depend on the application, the IIoT devices and their physical operating environments.



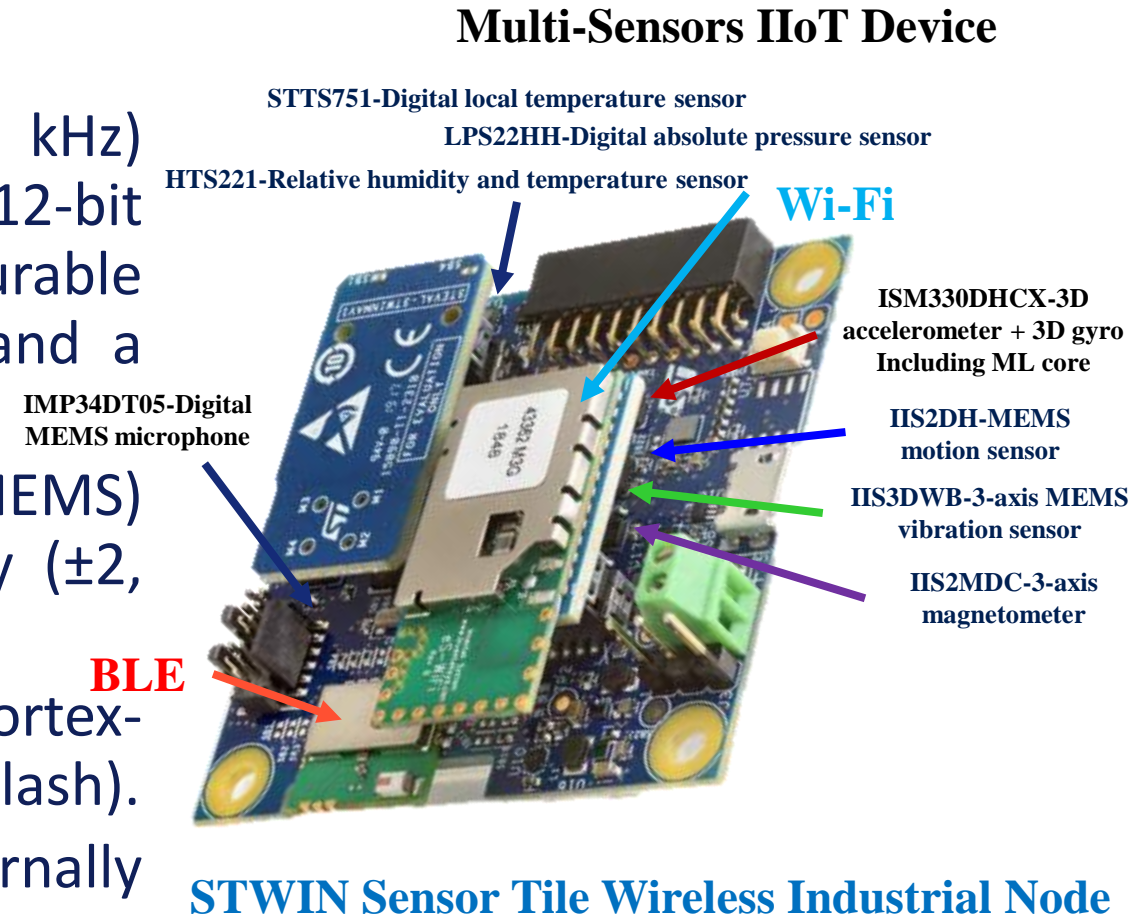
Sensor-driven PdM

- Sensor-driven PdM involves leveraging sensor data to predict mechanical machine failures before they happen.
- Rotating machine failures can be diagnosed and predicted by analyzing the vibration signal derived from accelerometers connected on industrial equipment.
- Sensor-driven PdM presents many challenges. Transforming raw sensor data into actionable insights is complex, time consuming, and costly, requiring a systematic engineering approach to building, deploying, and monitoring ML solutions.
- Many aspects need to be considered such as:
 - What type of data will capture the differences between classes
 - What signal length will capture the differences between classes
 - What range of sensor values will fully capture the range of the input information



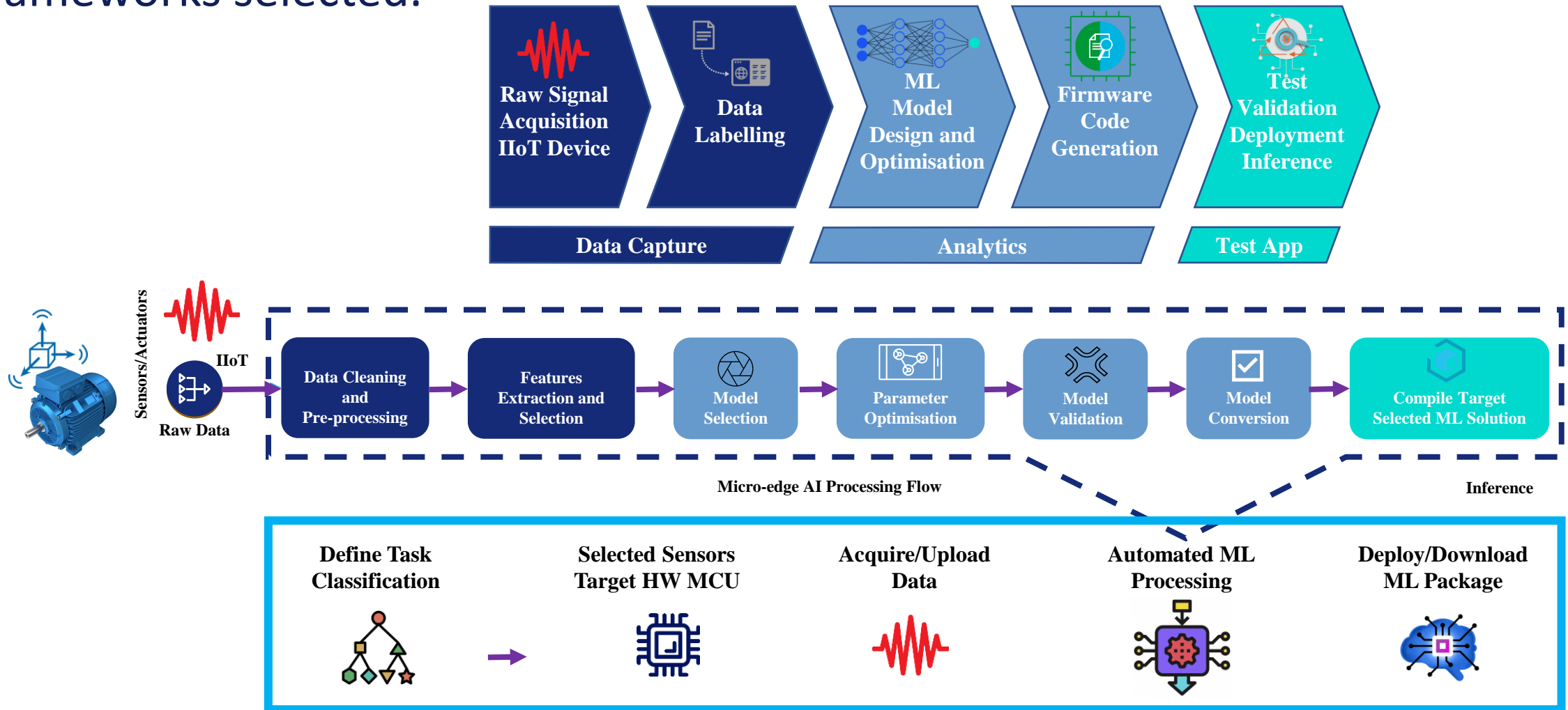
Experimental Architecture

- This micro-edge IIoT device used for the experiments comprises of:
 - Three axis ultrawide bandwidth (DC to 6 kHz) acceleration sensor (ISM330DHCX), a 12-bit analog-to-digital converter, a user-configurable digital filter chain, a temperature sensor, and a serial peripheral interface.
 - The micro electromechanical systems (MEMS) vibration sensor has a selectable sensitivity (± 2 , ± 4 , ± 8 , or ± 16 g)
 - Processing capabilities ensured by an Arm Cortex-M4 processor (120 MHz, 640 KB RAM, 2 MB Flash).
 - The micro-edge device can be powered externally or by an internal lithium-ion battery
 - BLE and Wi-Fi connectivity.

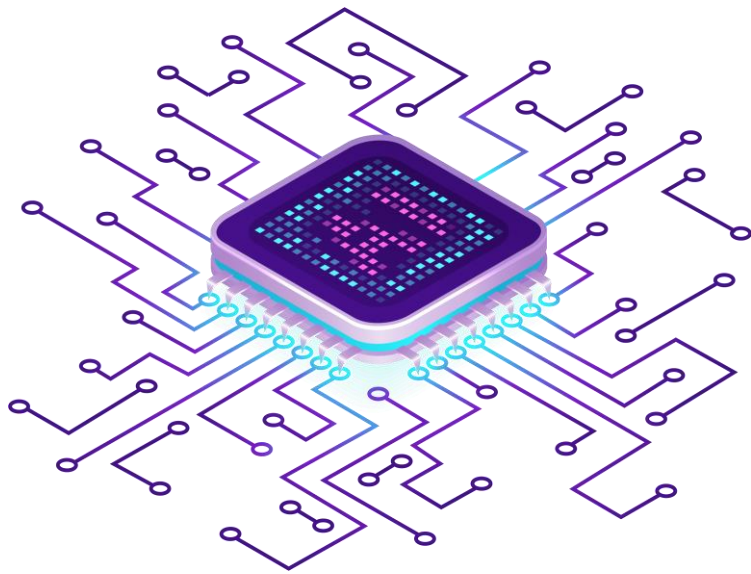


Flexible Model Development Workflow

- The micro-edge AI processing flow has been implemented for each of the frameworks selected.



Benchmarking Based on Three Different Frameworks



**NANOEDGE AI
STUDIO** 



 **EDGE IMPULSE**

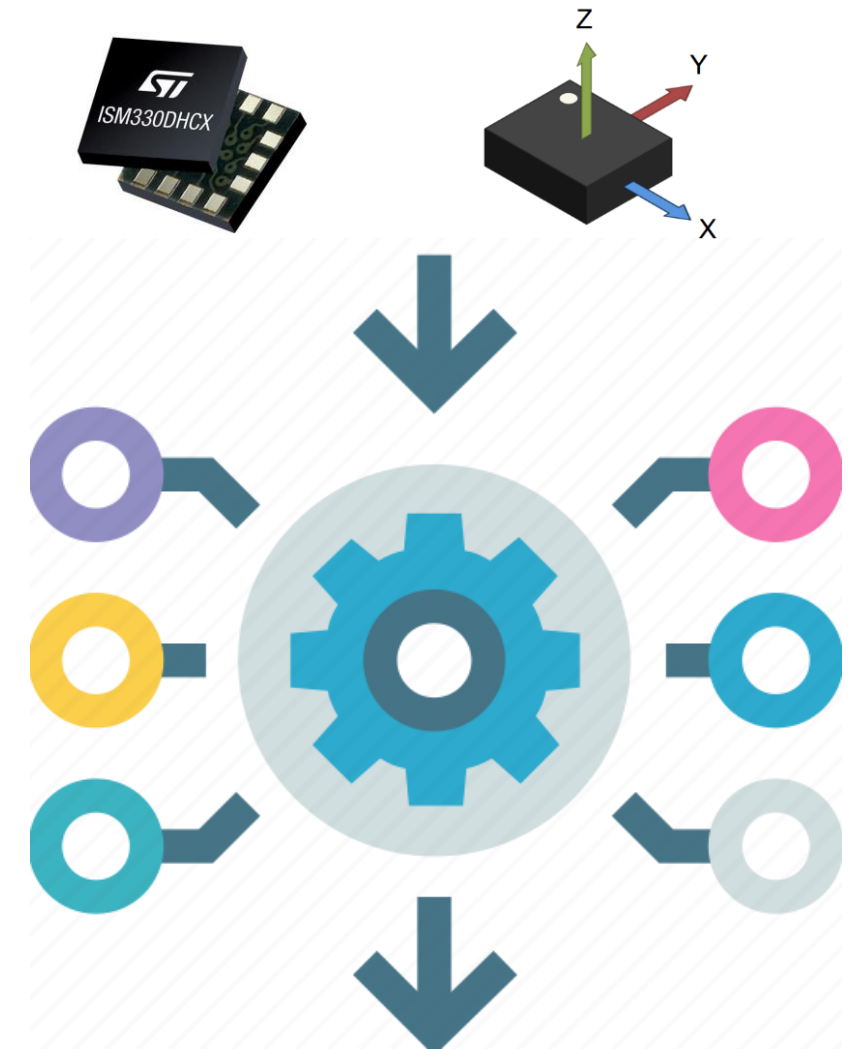
- In ML, benchmarking is the practice of comparing the performance of different model architectures within the same framework.
- In this presentation, benchmarking is about comparing different AI/ML platforms, which poses challenges due to the large number of factors involved.
- The aim has been to identify the most critical factors that impact on performance (key differentiating indicators - KDIs) and define consistent AI workflows.
- Three existing frameworks and inference engines for integrating AI mechanisms within MCUs have been employed:
 - Qeexo AutoML - automated ML platform for Arm Cortex-M0-to-M4-class processors
 - NanoEdge™ AI (NEAI) Studio,
 - Edge Impulse (EI)

Use Case Design

- Classification of the state of a motor based on vibration measurements
- A built-in three-axis accelerometer (ISM330DHCX) measures the accelerations of three orthogonal directions
- Classes defined based on conditions (motor speeds) and sub-conditions (malfunctions):
 - MIN: the motor is running at minimum speed
 - MED: the motor is running at half of the speed
 - MAX: motor is running at maximum speed
 - MIN_W: the motor is running at minimum speed with an excess load producing a centrifugal force
 - MED_W: the motor is running at half of the speed with an excess load producing a centrifugal force
 - MAX_W: the motor is running at maximum speed with an excess load producing a centrifugal force

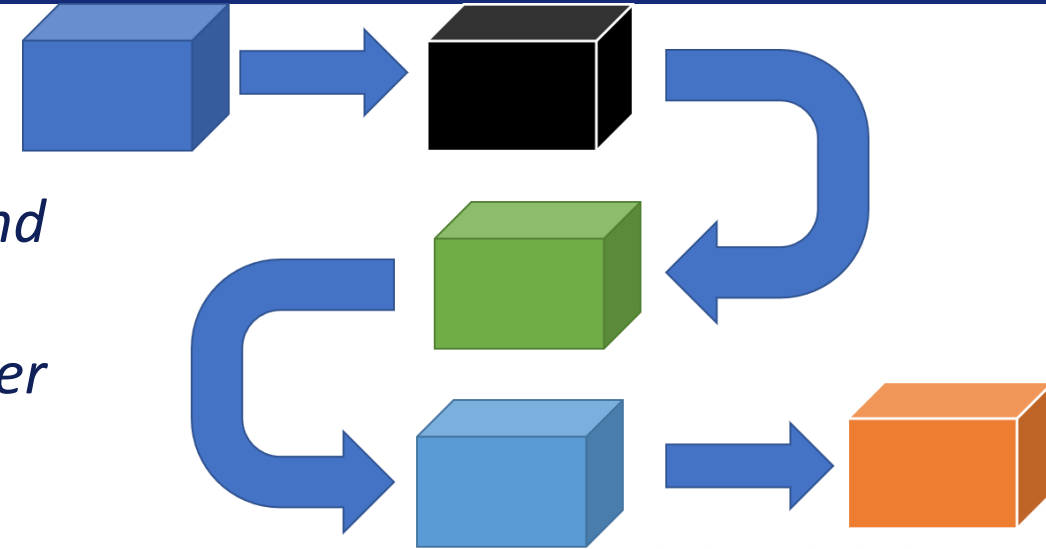
The use case was designed with the following goals in mind:

- The motor behaviour and the classification problem being solved with ML/DL were studied in-depth
- Classes should be distinguishable for easier classification
- Data sets should be class-balanced
- Data sets should be properly split (training, validation, test)



Comparison Factors

- The platforms offer various degrees of:
 - *Automation of various parts of the E2E workflow,*
 - *Transparency into the ML/DL algorithms and model architecture*
 - *Control over model parameters and hyper parameters*
 - *Pre-analysis in Time and Frequency Domain*
 - *Visualization and exploration of features*
 - *Model generation, optimization and selection*
 - *Testing*
 - *Support for neural network architectures*
 - *Customization for applications in PdM*
 - *Deployment facilities*
 - *Validation*



NANOEDGE AI
STUDIO

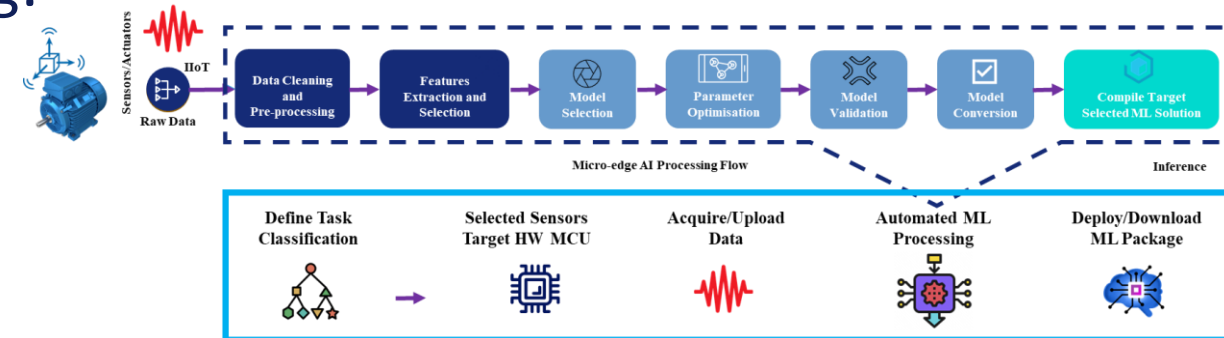
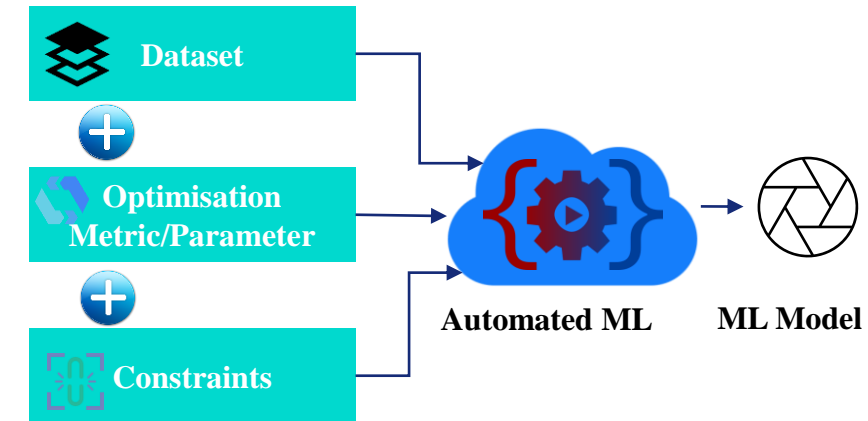
Qeexo

EDGE IMPULSE



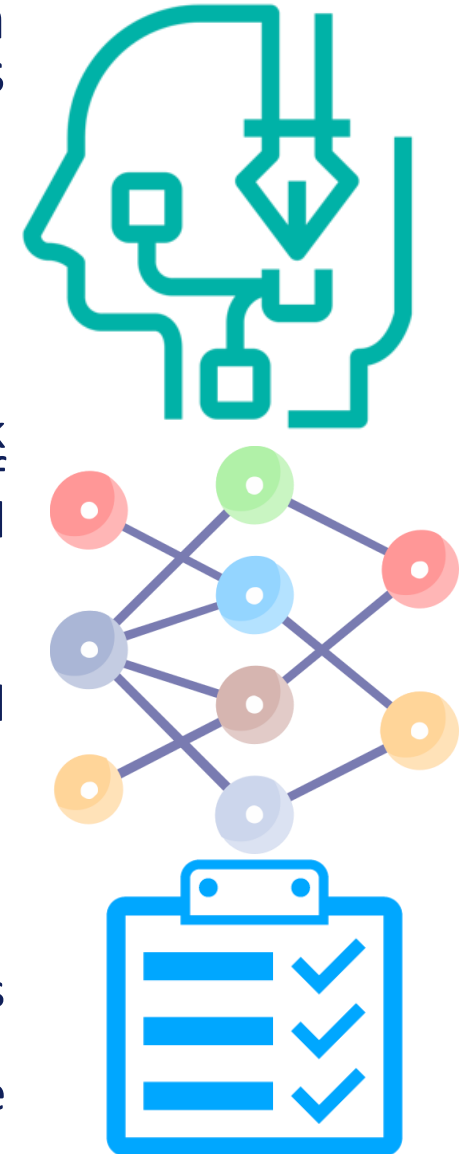
Automation of Various Parts of the E2E Workflow

- Automated machine learning (autoML) aims to make easier and more accessible the use of ML algorithms by removing tedious, iterative, and time-consuming work across the E2E workflow.
- The autoML process comprises different tasks, such as feature selection, feature extraction, model selection, and hyper-parameter tuning. In spite of the proliferation of autoML related technologies, many parts of the E2E are still highly dependent on expert interventions.
- The process of automating machine learning covers a wide range of automation topics, including:
 - Data preprocessing
 - Feature extraction
 - Feature engineering
 - Algorithm selection
 - Parameter and hyperparameter optimization
 - Model and data deployment, monitoring and management.



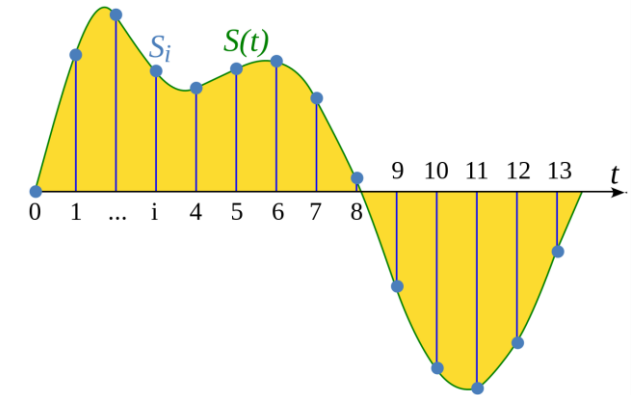
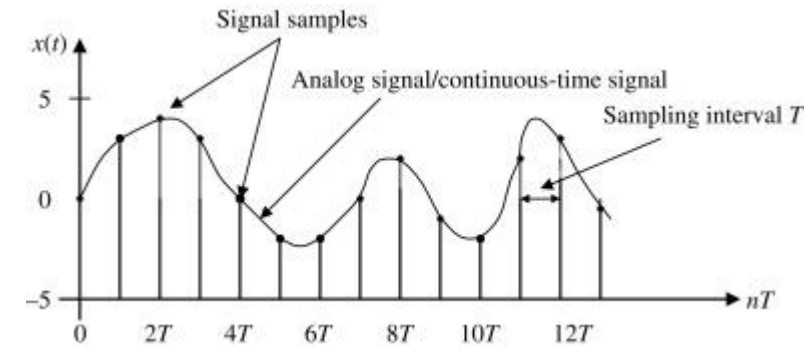
Challenges

- Building high-quality machine learning models through autoML is an iterative, resource-intensive, and time-consuming process that involves many different components.
- The three platforms does this at various degree of automation
- Transparency and control over parameters and hyperparameters
- Deployment and inference
 - The final step in the E2E workflow is to flash the compiled binary to STWIN and check that the classifier is producing the expected output. As shown in the video version of this presentation, the final model is able run inference on the embedded device and accurately recognize a variety of anomalous states, in real time
- Support neural network architectures
 - AI/ML platforms that do not support NN architectures, use feature extraction and then select from a wide range of traditional ML models.
 - NanoEdge – do not support NNs
 - Edge Impulse and Qeexo – do support NNs
- Validation
 - Although autoML compensate for many of the drawbacks of manual processes, it is still important to verify that the E2E workflow is easily repeatable and reproducible.
 - The particularities of verification and validation when deploying AI at the edge require at least one complementary workflow implemented with another framework



Sampling Methodology

- Main parameters are the same: frequency, range; the buffering method differ.
- Sampling frequency 1667 Hz; tested out different rates to figure out what will be the best option
- The higher the frequency, the higher the chances to get important features in the signal snapshot; however, look for memory, latency, and power consumption constraints
- Collection of signals (of approx. 30 seconds).
- NEAI: The length of the signal snapshot is approximately 300 milliseconds ($= 512/1667$) for a buffer size of 512 samples on each axis, in total 1536 values per signal.
- Qeexo: The length of the signal snapshot is 50 milliseconds. The buffer size is approximately 83 samples: $50/(1000/1667)$. The buffer size can vary (due to sample rate tolerance).



Three-step Signal Data Acquisition

NANOEDGE AI
STUDIO

EDGE IMPULSE

Qeexo

Collected



Format conversion

$$\text{Signal_length (ms)} = \frac{\text{Buffer_size}}{\text{ODR}} \times 1000$$

Uploaded

timestamp	Xacc	Yacc	Zacc
0	1102	212	165
0.6	1107	6	172
1.2	1038	-246	98
1.8	976	-347	2
2.4	958	-260	-62
2.999	1043	-9	-67
3.599	1148	284	4
4.199	1142	375	49

Format conversion

$$\text{Signals length (50 ms)} = \frac{\text{Buffer_size}}{1000} \times \text{ODR}$$

Uploaded

timestamp	accel	label
50	[[32764, 21836, 22449 MAX	
100	[[[-16063, -12445, -146 MAX	
150	[[[24899, 6033, 6153], [MAX	
200	[[[-10170, -1676, -1974 MAX	
250	[[[654, -17090, -17916] MAX	
300	[[[-2464, 7910, 5814], [MAX	
350	[[[11335, -5959, -2154] MAX	
400	[[[18886, 22490, 19167 MAX	

Uploaded

line 1	X ₀	y ₀	z ₀	x ₁	y ₁	z ₁	(...)	x ₂₅₅	y ₂₅₅	z ₂₅₅
line 2	X ₀	y ₀	z ₀	x ₁	y ₁	z ₁	(...)	x ₂₅₅	y ₂₅₅	z ₂₅₅
(...)										
line n	X ₀	y ₀	z ₀	x ₁	y ₁	z ₁	(...)	x ₂₅₅	y ₂₅₅	z ₂₅₅

Format conversion

Uploaded

timestamp	Xacc	Yacc	Zacc
0	1102	212	165
0.6	1107	6	172
1.2	1038	-246	98
1.8	976	-347	2
2.4	958	-260	-62
2.999	1043	-9	-67
3.599	1148	284	4
4.199	1142	375	49

Format conversion

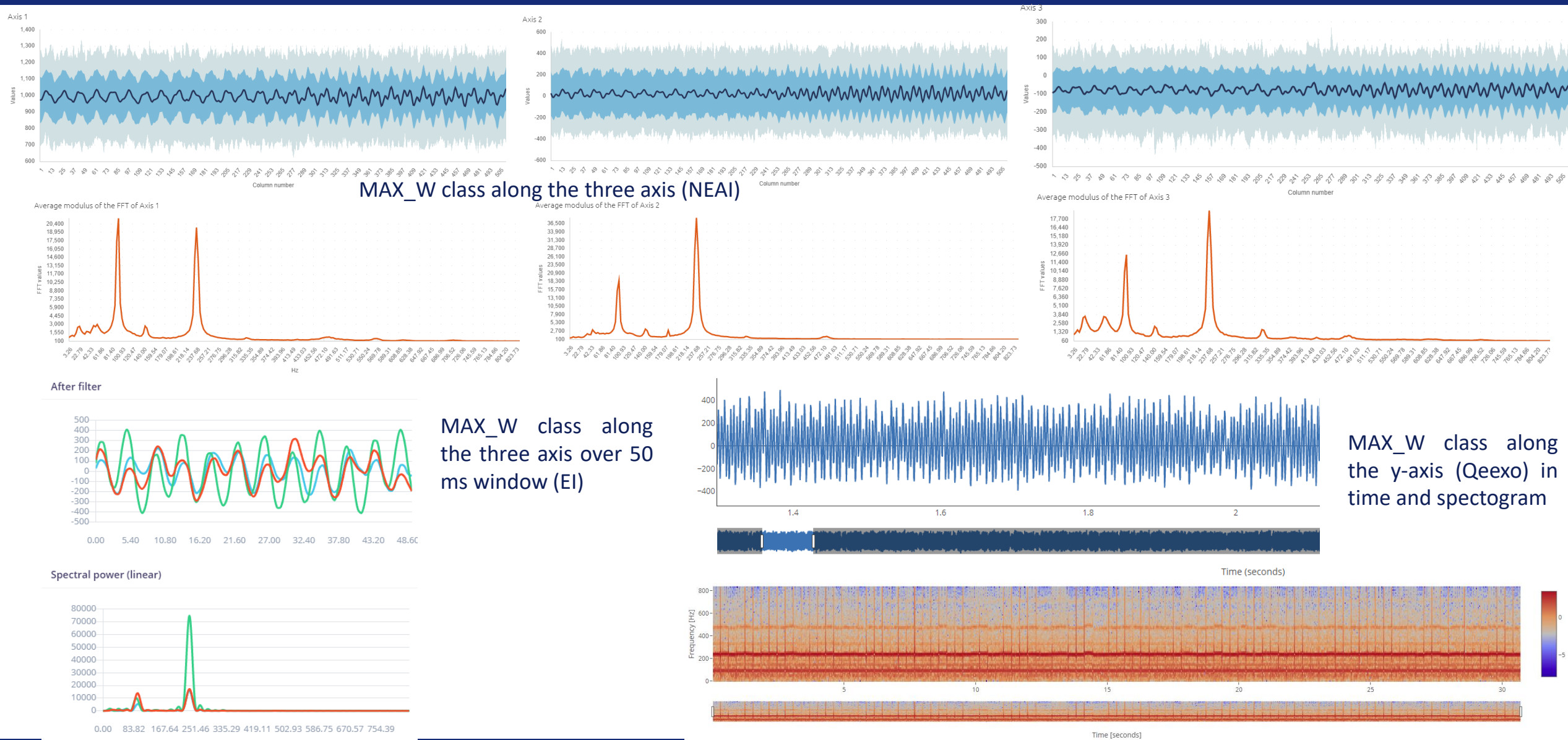
Collected



All three platforms offer both data collection (directly from sensors) and upload (from files). Three steps:

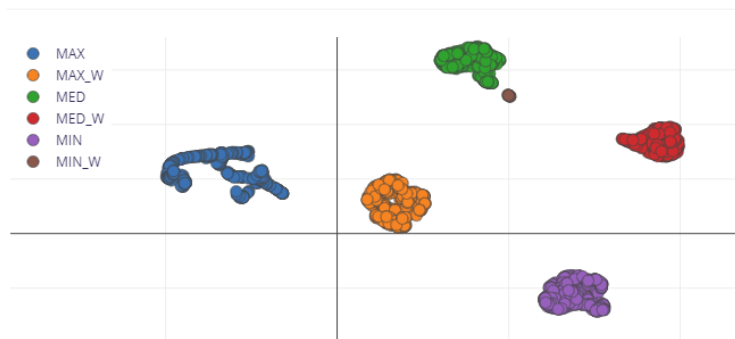
1. Collect sensor data with the platforms that allow connection with STWIN, i.e., NEAI and Qeexo
2. Cross-conversion of sensor data format (between the platforms)
3. Cross-generation of data sets (training, validation and test) and upload

Pre-analysis in Time and Frequency Domain

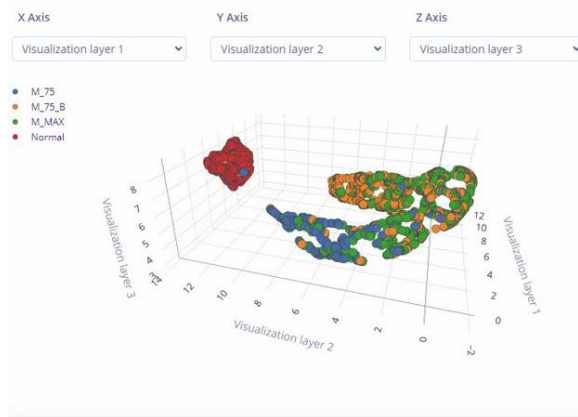


Visualization and Exploration

A useful aspect is the possibility to visualise and explore the features. The fact that the features are visually clustered is a good indication that the model can be trained to perform the classification.



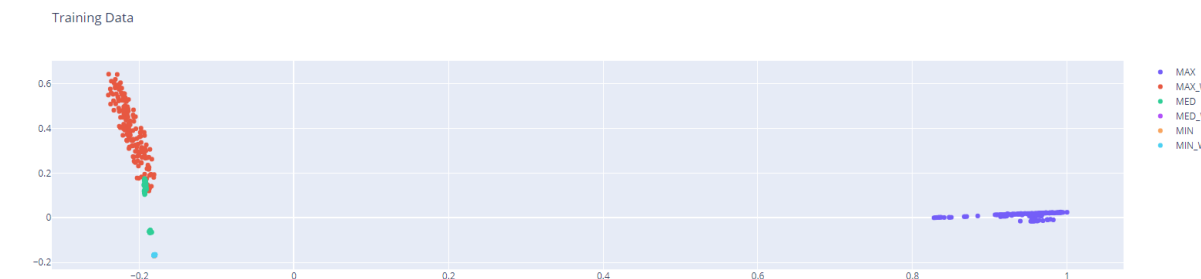
Feature explorer with EI (in 2D). Classes are distinguishable.



Example from a previous project with EI (in 3D), where classes are not very distinguishable.



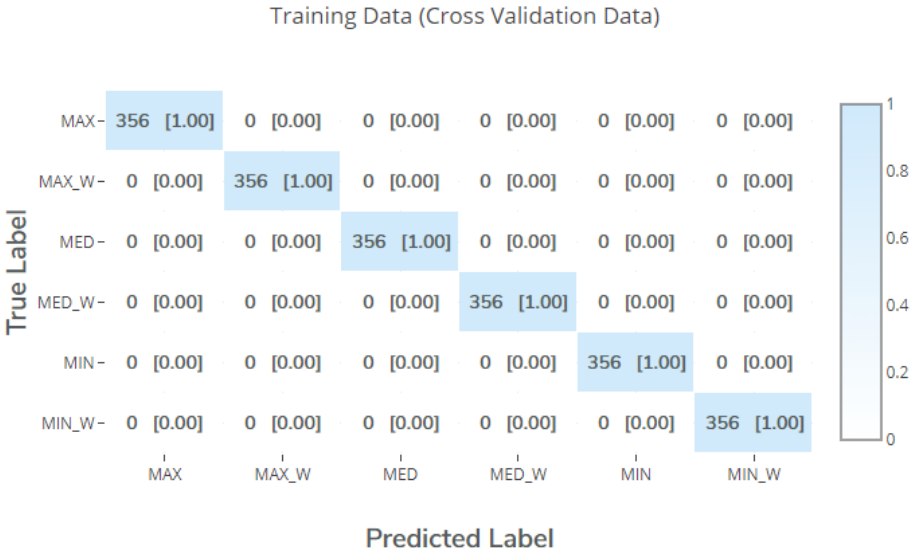
Qeexo UMAP (Uniform Manifold Approximation and Projection) plot - shows how separable the classes under consideration are with respect to the selected group of features.



Qeexo PCA (Principal Component Analysis) plot - shows how separable the classes under consideration are with respect to the selected group of features.

Benchmarking

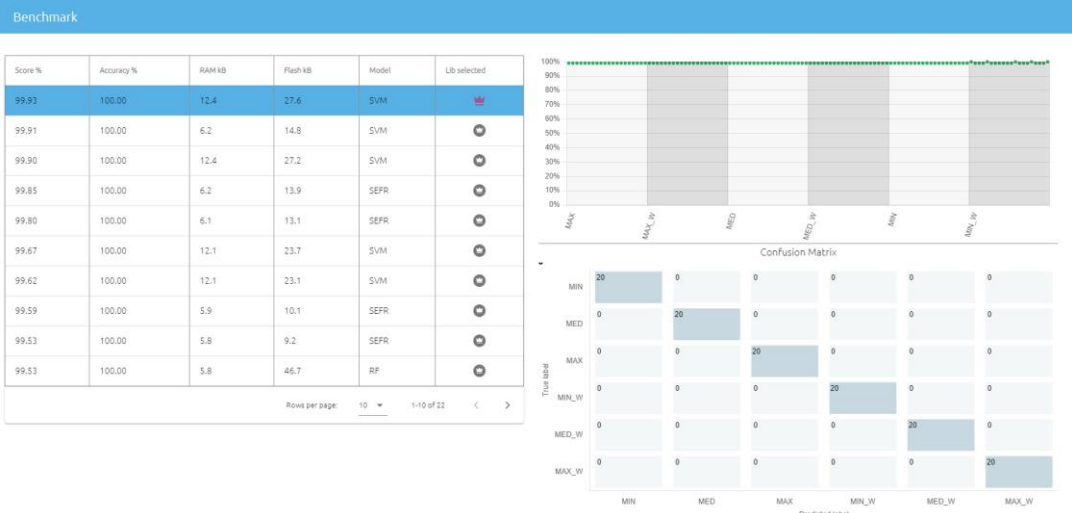
Benchmarking allows to compare the performance of different model architectures within the same framework. The confusion matrix of the validation data is a useful evaluation tool.



Benchmarking with Qeexo. Confusion matrix for SVM model

ML MODEL	CROSS VALIDATION	TEST PERFORMANCE	LATENCY	SIZE	PERFORMANCE SUMMARY	SAVE	PUSH TO HARDWARE	LIVE CLASSIFICATION ANALYSIS
Artificial Neural Network	1.0 +/- 0.0	1.00	Click	68.62 KB				LIVE TEST
Decision Tree	0.99 +/- 0.01	0.87	Click	180 B				LIVE TEST
Support Vector Machine	1.0 +/- 0.0	0.99	Click	1.08 KB				LIVE TEST

Benchmarking with Qeexo. Overview trained models.



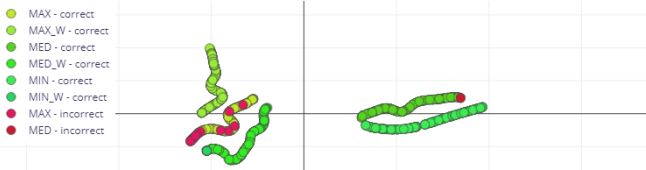
Benchmarking with NEAI. All correctly classified (green dots)



Confusion matrix (validation set)

	MAX	MAX_W	MED	MED_W	MIN	MIN_W
MAX	99.1%	0.0%	0%	0%	0%	0%
MAX_W	0%	100%	0%	0%	0%	0%
MED	0%	0%	100%	0%	0%	0%
MED_W	0%	0%	0%	100%	0%	0%
MIN	0%	0%	0%	0%	100%	0%
MIN_W	0%	0%	0%	0%	0%	100%
F1 SCORE	1.00	1.00	1.00	1.00	1.00	1.00

Data explorer (full training set)



Benchmarking with EI. Confusion matrix and data explorer.

Correctly classified (green dots) and misclassified (red dots).

Testing

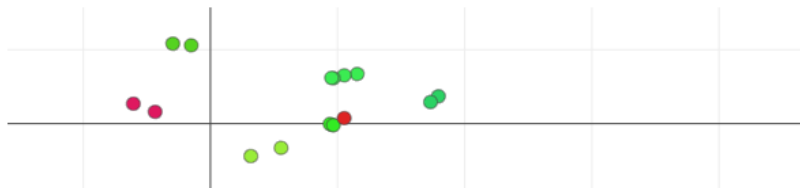
- Testing is evaluation of the trained model performance on the **testing dataset** or **live** to analyse how well the model performs against unseen data prior to its deployment on the device.
- Live testing ensures unbiased evaluation of model effectiveness (completely new signals, not seen before).
- The results show that the classifier manages to properly reproduce and detect all classes with reasonable certainty percentages, and these are comparable

 **ACCURACY**
97.43%

	MAX	MAX_W	MED	MED_W	MIN	MIN_W	UNCERTAIN
MAX	89.6%	0%	8.6%	0%	0%	0%	1.8%
MAX_W	0%	96.6%	0%	0%	0%	3.1%	0.3%
MED	0%	0%	100%	0%	0%	0%	0%
MED_W	0%	0%	0%	99.2%	0%	0%	0.8%
MIN	0%	0%	0%	0%	100%	0%	0%
MIN_W	0%	0.3%	0%	0%	0%	99.2%	0.5%
F1 SCORE	0.95	0.98	0.96	1.00	1.00	0.98	

Feature explorer ?

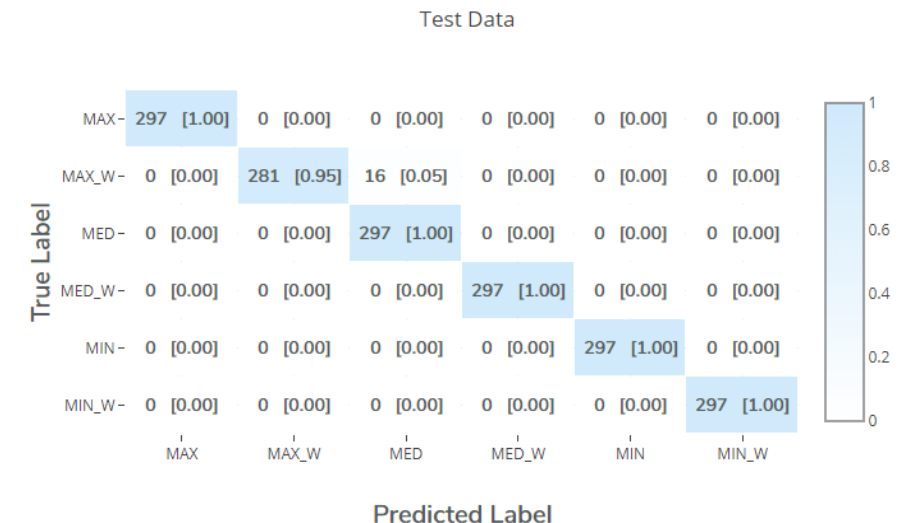
- MAX_W - correct
- MED - correct
- MED_W - correct
- MIN - correct
- MIN_W - correct
- MAX - incorrect
- MED_W - incorrect



El testing with test datasets based on signals collected with NEAI



Live testing using NEAI microcontroller emulator



Qeexo testing with test datasets collected with Qeexo

Video Test - NanoEdgeAI

FROM FILE

FROM SERIAL (USB)

COM Port
COM8

↻

Baudrate
115200

↻

☐ Maximum number of lines
100


▶ START/STOP

↻

Number of lines: 0

Serial output
1050 -45 -54 1044 -45 -54 1030 -38 -87 1015 -20 -116 995 17 -140 970 76 -163 948 143 -176 938 198 -171 941 221 -146 951 210 -108 959 174
-71 960 116 -41 965 38 -23 986 -42 -10 1005 -107 -4 1016 -147 -8 1024 -171 -18 1027 -185 -37 1023 -184 -62 1022 -157 -84 1022 -102 -97 1020
-22 -114 1019 75 -121 1017 178 -101 1012 253 -89 993 278 -81 967 259 -76 949 207 -76 937 142 -69 927 75 -66 927 4 -74 937 -58 -85 952 -102

Emulator function outputs



```
{  
  "lib_id": "6324aa8ee710b7132c09f5ad",  
  "class_name": ["MIN", "MED", "MAX", "MIN_W", "MED_W", "MAX_W"]  
}
```


Video Test - QeeXO

< Live Testing

HARDWARE CONNECTION
USB - Connected

✓ READY

Ann

CLASS LABEL	SENSITIVITY WEIGHTS	DATE
MAX	1	9/16/2022 9:17 PM
MAX_W	1	
MED	1	
MED_W	1	
MIN	1	
MIN_W	1	

EXPAND ▾

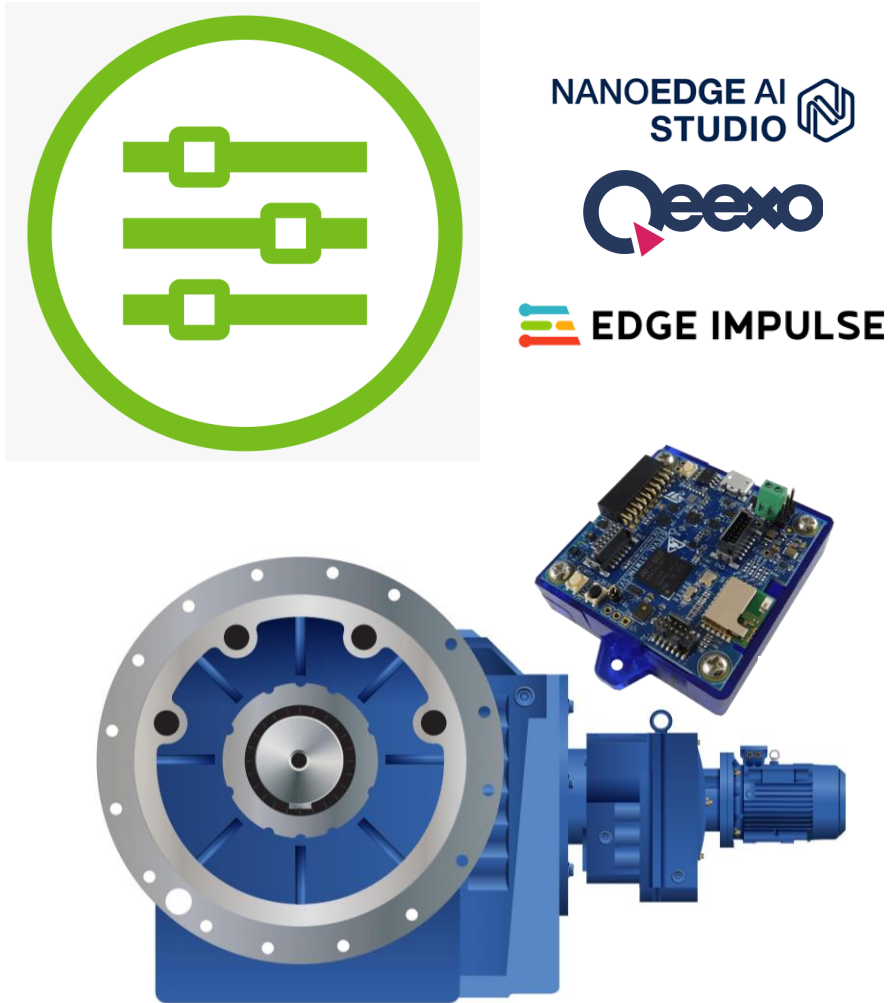
☒ Continuous Classification

☐ Event Classification



MIN_W

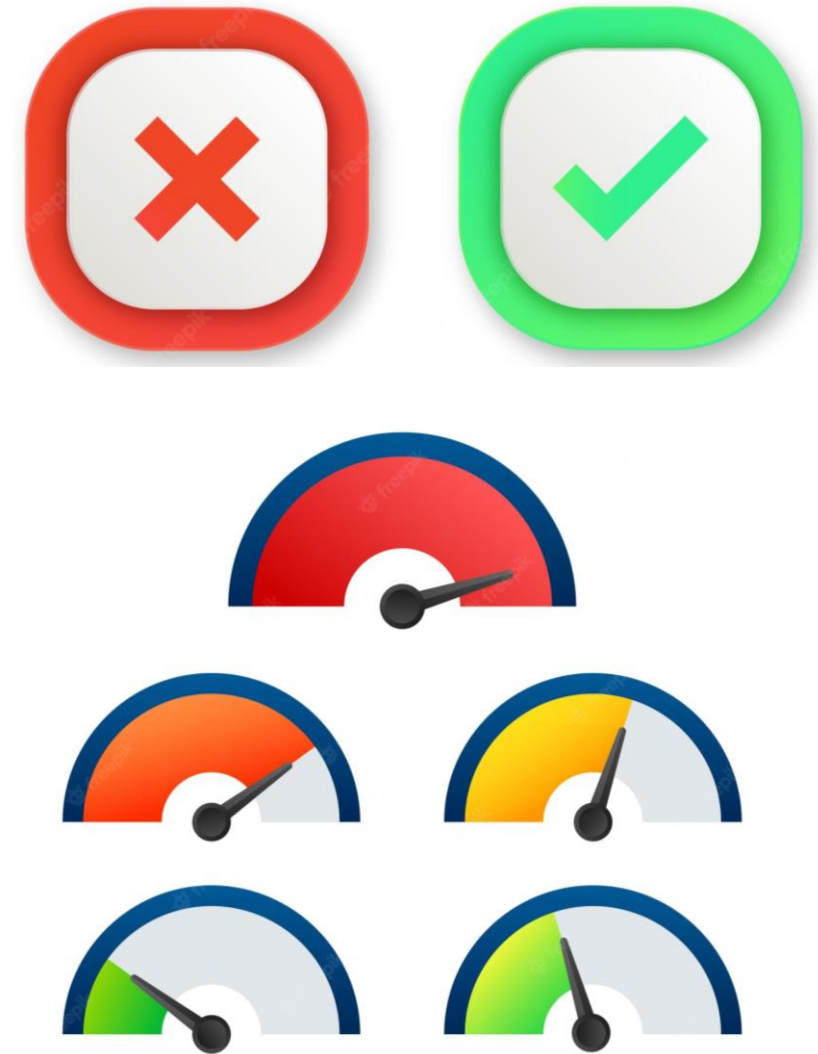
Summary and Future Work



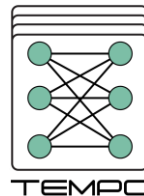
- Each framework presented was benchmarked by assessing some of the most important KDIs in AI/ML platforms. It also describes the most relevant factors that might affect the KDIs.
- Transforming raw sensor data into actionable insights is complex, time consuming, and costly, requiring a systematic engineering approach to building, deploying, and monitoring ML solutions
- The benchmarking findings indicate that no single AI framework can outperform all other frameworks across all KDIs. The frameworks have different approaches for core tasks, such as model selection, (hyper)parameter optimisation and deployment, thus possessing unique capabilities and weaknesses.

Summary and Future Work

- All frameworks provide relevant results, and as they evolve and borrow ideas from each other, they will also gain more strength and overcome weaknesses.
- The particularities of verification and validation when deploying AI at the edge require at least one complementary workflow implemented with another framework.
- Future work is intended to enlarge the comparison by considering additional frameworks and KDIs.



Event Organisers



The Key Digital Technologies Joint Undertaking - the Public-Private Partnership for research, development and innovation – funds projects for assuring world-class expertise in these key enabling technologies, essential for Europe's competitive leadership in the era of the digital economy. KDT JU is the successor to the ECSEL JU programme. www.kdt-ju.europa.eu

The AI4DI project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826060. The JU receives support from the European Union's Horizon 2020 research and innovation programme and the national authorities. www.ai4di.eu

The TEMPO project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826655. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Belgium, France, Germany, The Netherlands, Switzerland. www.tempo-ecsel.eu

The ANDANTE project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 876925. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Belgium, France, Germany, The Netherlands, Portugal, Spain, Switzerland. www.andante-ai.eu



Thank You

For your attention

@ Ovidiu.Vermesan@sintef.no

