



**HAL**  
open science

# Ultra-low Power Embedded Unsupervised Learning Smart Sensor for Industrial Fault Detection

Sebastián Marzetti, Valentin Gies, Valentin Barchasz, Hervé Barthélemy,  
Hervé Glotin

► **To cite this version:**

Sebastián Marzetti, Valentin Gies, Valentin Barchasz, Hervé Barthélemy, Hervé Glotin. Ultra-low Power Embedded Unsupervised Learning Smart Sensor for Industrial Fault Detection. 2020 IEEE International Conference on Internet of Things and Intelligence System (IoTaIS), Jan 2021, Bali, Indonesia. hal-03610004

**HAL Id: hal-03610004**

**<https://hal.science/hal-03610004>**

Submitted on 16 Mar 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Ultra-low Power Embedded Unsupervised Learning Smart Sensor for Industrial Fault Detection

Sebastián Marzetti\*, Valentin Gies\*<sup>‡</sup>, Valentin Barchasz<sup>‡</sup>, Hervé Barthélemy\*, Hervé Glotin<sup>†</sup>

\*Aix Marseille Université, Université de Toulon, CNRS, IM2NP - UMR 7334, France

<sup>†</sup>Aix Marseille Université, Université de Toulon, CNRS, LIS - UMR 7020, France

<sup>‡</sup>Université de Toulon, SMIoT, France

**Abstract**—In this paper, an ultra-low power embedded unsupervised learning smart vibration sensor is proposed for automatic industrial monitoring and fault detection. Using K-means algorithm, it is able to detect abnormal vibrations patterns. Architecture of the system is first presented, then embedded processing algorithms composed of feature extraction and k-means algorithm are detailed, and finally an implementation on a vibrations simulator machine is described. Results show that faults can be detected with an accuracy of  $XX\%$  using less than 0.15% of average embedded processor resources on a *ARM M4F* with an average consumption of  $80\mu W$ . This smart sensor is relevant for Industrial Internet Of Things (IoT) autonomous monitoring applications, having more than one year of battery life using a single *CR2032* coin cell.

**Index Terms**—Unsupervised Learning, K-Means, Machine Learning, Embedded Learning, Pattern Detection, Embedded system

## I. INTRODUCTION

Industrial predictive maintenance is important for preventing hardware failures having consequences ranging from equipment downtime to industrial disasters. Several solutions for Industrial Internet of Thing (IIoT) monitoring equipments are existing. Most of them are using a cloud for gathering data from distant sensors and for extracting information using artificial intelligence algorithms [1]. In applications where it is difficult to install or to maintain sensors, having a power supply is a strong operational constraint. This is particularly true in harsh environments such as explosive atmosphere (ATEX) ones where adding a power supply requires a lot of precautions, when sensors are cast in concrete in applications such as construction industry, or when maintenance is difficult such as in radioactive environments.

Consequently, power consumption is a relevant classification criterion in IoT. As described in Fig. 1, there is a trade-off between computing capabilities and power consumption. Depending on the application, this trade-off can be strongly oriented to energy saving in case of long life applications such as ultra low power detectors powered by a single battery cell [2], [3], or to computation power in traditional AI applications such as deep learning using GPU

in smartphones or embedded robots [4], [5].

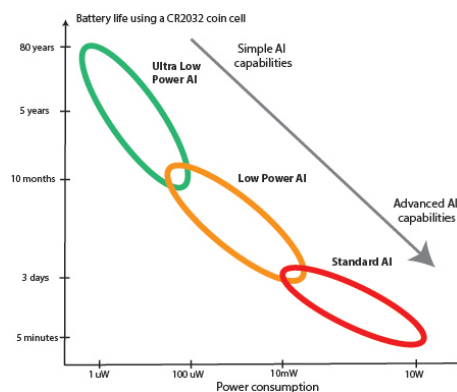


Fig. 1. Different types of embedded AI according to their power consumption

In IIoT, when adding a power supply is not possible or when maintenance is difficult, ultra low-power embedded artificial intelligence is required. Having a power consumption limited to  $100\mu W$ , it allows to run embedded artificial intelligence algorithms for a very long time without recharging or changing batteries. To set the ideas, a system using a standard *CR2032* battery cell (3V - 240mAh - 0.72Wh) and running continuously during 5 years has a power budget equal to  $16\mu W$ , that means approximately  $I \simeq 10\mu A$  on a 1.8V system. This is done at the price of reducing algorithmic complexity such as in smart wake-up systems [3], [6]–[9]. However, it can also include biomorphic neural implementations such as in [10], and it can have many applications [11].

However, in order to reduce power consumption, it is also necessary to avoid RF data transmission, except for generating alerts. Thus, it is necessary to extract these ones from the raw data, requiring artificial intelligence embedded processing called AIoT. These learning algorithms require to compute scalar product for neural computations, thus a fixed or floating point multiplier is necessary and can be achieved using a higher power micro-controller, such as

CC2652 whose power consumption is  $10mW$ . Consequently, running these learning algorithms **continuously** cannot be ultra-low power. In order to cope with this issue, it is possible to implement algorithms using an always-on ultra-low-power analog circuit coupled with a feature extractor and a classifier activated only when necessary. In this situation, the micro-controller will spend most of its time in sleep mode consuming approximately  $1\mu W$ , and will be **sparsely activated** in order to process a buffered signal. Considering an active time equal to 0.1% of the overall time, average power consumption of this micro-controller falls down to  $\simeq 10\mu W$ , leading to an ultra-low-power behaviour.

This paper presents an application of this mixed analog-digital ultra-low-power system: a long-life battery-powered industrial fault detector based on vibration analysis. Used for predictive maintenance, system behaviour can be monitored and learned autonomously using feature extraction and unsupervised learning algorithms. Each time data features are extracted from the sensor, they are classified: a new data can be part of an existing cluster, or can be considered as an abnormal signal corresponding to a failure or to a new cluster. Similar solutions have already been proposed in the literature [12]–[16] but are not ultra-low-power implementations. In our implementation, autonomy using small size batteries is very important while having an always-on fault detection. It can be used in several industrial areas [17]. Paper is divided in three parts:

- Section II presents the architecture and the electronic circuits of the system.
- Section III presents the embedded feature extractor and the K-Means algorithm used for fault detection.
- In section IV, results are presented and discussed.

## II. SYSTEM ARCHITECTURE

Proposed system architecture is presented in Fig.2. It can be supplied with a single 3.7V Li-Ion battery whereas its lifetime is XX months on a CR2032 battery, with a limited bill of materials (less than 20\$).

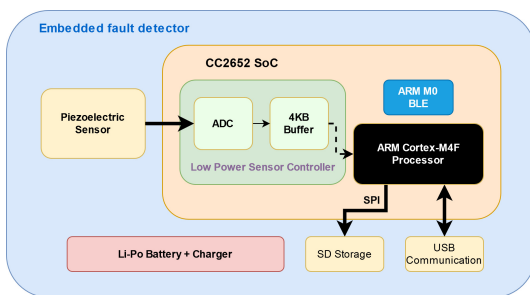


Fig. 2. System block diagram.

The most important parts of this architecture are:

- A passive piezoelectric acoustic sensor.
- An analog front end for amplifying and pre-processing the input signal.

- A Texas Instrument CC2652 system on chip (SoC) including:
  - An ultra-low-power sensor controller for analog-to-digital conversion and data buffering.
  - A low-power micro-controller ARM M4F for system management, feature extraction and fault detection using an unsupervised learning classifier.
  - A micro-controller ARM M0 for radio-transmission of alerts.

All these parts are detailed in the following sub-sections.

### A. Piezoelectric acoustic sensor

Machine vibrations produce noise propagating in the mechanical structure. This noise is characteristic of the machine behaviour: its spectrum has spectral peaks corresponding to the harmonics of the rotor frequency. Amplitudes of the harmonics are related to small or bigger imbalance of the rotor. Fig.3 shows an example of signal and spectrogram of a well-balanced rotor, whereas Fig.4 shows an example of signal and spectrogram of an unbalanced rotor. Frequency spectrum can be considered as a signature of the system that can be used for fault detection. On the proposed examples, harmonics are more important in the unbalanced system, especially at high rotation frequencies, close to the resonance.

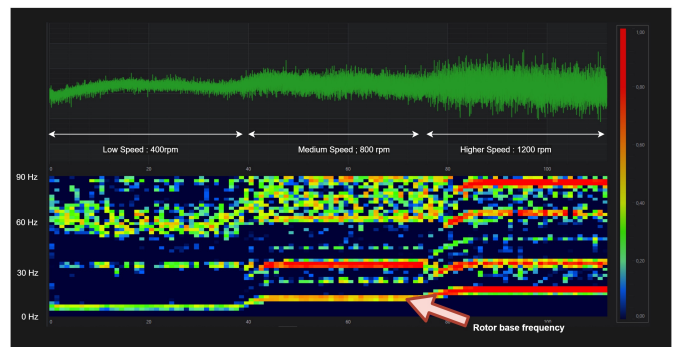


Fig. 3. Signal and spectrogram of a well-balanced rotating machine. Spectrogram composed by successive FFT spectrums using 256 samples, using a Hamming window. Sampling frequency is  $200Hz$ . Frequencies displayed are limited to  $100 Hz$ . Rotor base frequency (up to 20 Hz) corresponds to the bottom line.

In an ultra-low-power application, a piezoelectric acoustic sensor is ideal for measuring vibrations because its power consumption is null as it is passive, and power consumption of the associated charge amplifier is very low. Sensor has to be in contact with the machine surface, firmly maintained as shown on Fig. 5.

### B. Analog front-end

Piezoelectric signal is first amplified using a charge amplifier having a biasing around  $V_{dd}/2$ , where  $V_{dd}$  is power supply voltage of the system. This amplified signal has to be sampled using an ADC to be processed by

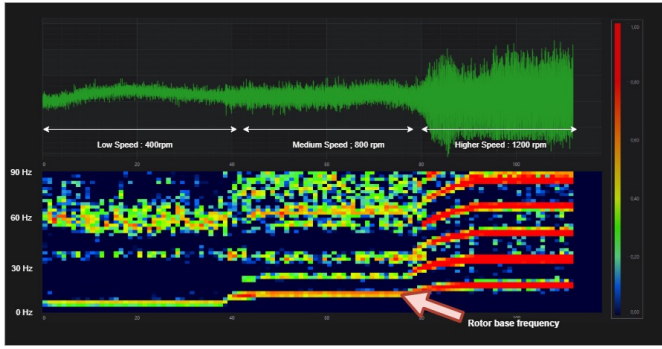


Fig. 4. Signal and spectrogram of an unbalanced rotating machine. Spectrogram composed by successive FFT spectrums using 256 samples, using a Hamming window. Sampling frequency is  $200\text{ Hz}$ . Frequencies displayed are limited to  $100\text{ Hz}$ . Rotor base frequency (up to  $20\text{ Hz}$ ) corresponds to the bottom line.

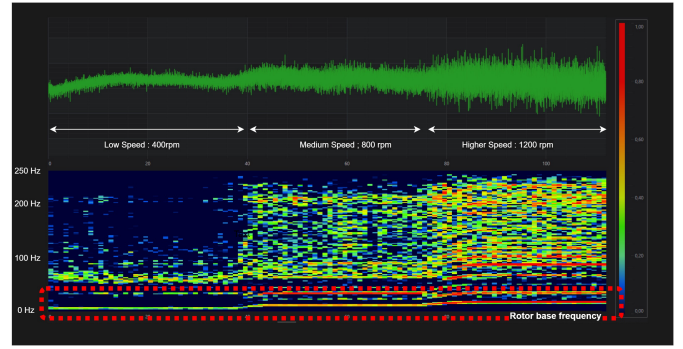


Fig. 6. Signal and spectrogram of a well-balanced rotating machine, composed by successive FFT spectrums, using a Hamming window. Sampling frequency is  $500\text{ Hz}$ . High frequency noise is important at high speeds.

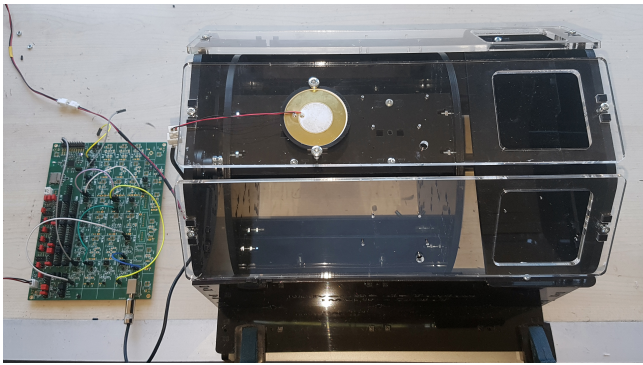


Fig. 5. Rotating test bench with piezoelectric sensor attached on it (golden circle), and the ultra-low power board at the left. Rotor can be unbalanced using additional asymmetrical weights.

the micro-controller. In order to have an ultra-low power consumption in always-on mode, it is necessary to limit the ADC data acquisition frequency. As shown in Fig. 3, using the first 3 main peaks (frequency and amplitude) of the spectrum is a relevant signature of the motor mode. As in most industrial rotating machines, rotor maximum frequency is  $25\text{ Hz}$ , having a system able to analyse vibration spectrum up to  $100\text{ Hz}$  is a good trade-off between power-consumption and efficiency. This leads to an optimal sampling frequency of  $200\text{ Hz}$  for the ADC in our application.

Input noise having a wide spectrum as shown on 6, an anti-aliasing filter is necessary before sampling at  $200\text{ Hz}$  : an active second order Butterworth low-pass filter has been chosen with a cut-frequency of  $f_C = 90\text{ Hz}$ .

Analog front-end circuit is shown in Fig. ???. Ultra low power ADC are used for the charge amplifier and for the active filter, each one having a Gain-Band Product (GBP) equal to  $12\text{ kHz}$  and a power consumption equal to  $0.4\mu\text{ A}$ . Power consumption of the analog front-end including linear voltage regulator losses is lower than  $2\mu\text{ A}$ .

### C. Digital part : TI CC2652 system on chip

Dedicated to Internet of Things (IoT) application, an low-power System on Chip (SoC) from TI is used in this detector, for data acquisition, processing, and alert transmission using a radio link. It is composed of the following parts :

1) *Ultra low-power sensor controller*: Implemented in the SoC for data acquisition and basic processing, this dedicated ultra low-power processor features a  $4\text{ KB}$  SRAM for storing sampled input data waiting for processing it. Sampling is done using a  $12\text{ bits}$  ADC converter included also in the controller. In our application, two sampling frequency have been tested :  $200\text{ Hz}$  and  $500\text{ Hz}$ . At  $200\text{ Hz}$ , power consumption of this always on digital part is  $30\mu\text{ A}$  whereas at  $500\text{ Hz}$ , power consumption of this always on digital part is :  $75\mu\text{ A}$  .

2) *ARM Cortex M4F Processor*: After having enough data in the input buffer, the ultra low power sensor controller wakes-up the *ARM M4F* processor for feature extraction and classification using K-means optimized algorithm.

3) *ARM M0 for radio transmission*: In case of a failure detection, an alert is sent through a hardware interface called the RF doorbell using a third processor, an *ARM<sup>®</sup> Cortex<sup>®</sup>-M0* in charge of handling the Bluetooth Low Energy RF protocol used for alert message communication and monitoring.

4) *Other circuits*: In addition to these analog and digital circuits, a non-rechargeable  $3\text{ V}$  battery is used. A micro SD data storage using SPI and a USB connection for real time monitoring have also been added for debugging purpose.

### III. EMBEDDED SIGNAL PROCESSING ALGORITHMS

Once a sequence of vibration input signal has been captured and buffered in ultra-low power mode by the sensor controller, it has to be analysed using more powerful and power consuming tools. *ARM M4F* micro-controller contained in the *CC2652* is used for this task, implementing embedded



artificial intelligence algorithms such as feature extraction, unsupervised learning using K-Means algorithm and failure detection as shown in Fig. 7. Each block is described in the following subsections :

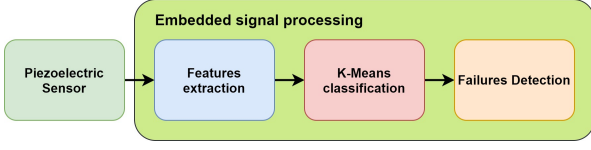


Fig. 7. Signal processing chain.

### A. Features extraction

First step of the proposed algorithm is to extract relevant features in the input signal [?]. Considering that the detector will be used for detecting failures in industrial equipments used repetitively for long periods, a frequency analysis has been chosen. It is based on the following sequence of operations :

- Embedded computation of frequency spectrum (Fig.??) using a Fast Fourier Transform (FFT) on 256 points.
- Embedded detection of FFT peaks : amplitude and frequency of the 3 main peaks of the FFT spectrum are used as inputs of the k-means algorithm. Peaks are ordered by increasing frequency, in order to avoid misclassification due to peaks having nearly the same amplitudes.

Features used for classification are the frequency and the amplitude of the 3 peaks.

### B. K-Means unsupervised learning and detection algorithm

In order to know whether a input signal corresponds to a well known situation or not, extracted features have to be classified. Considering that the detector will be used in many different applications and contexts, it is necessary to learn automatically without any supervision the reference situations in which the detector will be used. This is done using K-Means optimized algorithm. It is optimized as classification is done in real time without storing any dataset. This allows an implementation in a micro-controller. Training and detection algorithms are presented in Fig. 8 and 9 and work as follow :

Each cluster  $C_i$  is described by by a 6 components vector  $C_i = [\overline{A_{i1}}, \overline{A_{i2}}, \overline{A_{i3}}, \overline{f_{i1}}, \overline{f_{i2}}, \overline{f_{i3}}]$ , where  $\overline{A_{ik}}$  is the average amplitude and  $\overline{f_{ik}}$  is the average frequency of the  $k^{th}$  peak.  $\sigma_{A_{ik}}$  and  $\sigma_{f_{ik}}$  are the corresponding dispersions of each component of the cluster.  $n_i$  is the number of samples classified in the cluster.

Each time a feature extraction has been finished on a new sample, distance  $D$  between its signature  $S_i = [A_{i1}, A_{i2}, A_{i3}, f_{i1}, f_{i2}, f_{i3}]$  and each cluster is evaluated as shown in Eq. 1:

$$D = \sum_{k=1}^3 \sqrt{\left(\frac{A_{i1} - \overline{A_{i1}}}{\sigma_{A_{ik}}}\right)^2 + \left(\frac{f_{i1} - \overline{f_{i1}}}{\sigma_{f_{ik}}}\right)^2} \quad (1)$$

If  $D < D_{Max}$  where  $D_{Max}$  is a classification parameter to be optimized offline, sample belongs to the cluster and corresponding centroid is updated as presented in Eq. 2 :

$$\begin{aligned} \overline{A_{ik}} &= (1 - \alpha)\overline{A_{ik}} + \alpha A_{ik} \\ \overline{f_{ik}} &= (1 - \alpha)\overline{f_{ik}} + \alpha f_{ik} \\ \sigma_{A_{ik}} &= (1 - \alpha)\sigma_{A_{ik}} + \alpha |A_{ik} - \overline{A_{ik}}| \\ \sigma_{f_{ik}} &= (1 - \alpha)\sigma_{f_{ik}} + |f_{ik} - \overline{f_{ik}}| \\ n_i &= n_i + 1 \end{aligned} \quad (2)$$

Eq. 2 shows that  $\overline{A_{ik}}$  and  $\overline{f_{ik}}$  are moving averages of  $A_{ik}$  and  $f_{ik}$ . Computation is much lighter than using an arithmetical mean, thus reducing power consumption. It is the same for dispersion which is not a standard deviation computation for reducing power consumption. Number of elements  $n_i$  of the cluster is incremented in order to have an indication on cluster usefulness.

If this  $D > D_{Max}$ , there are two options :

- If the sensor is in learning phase, a new cluster  $C_i$  is created with the following initializations :

$$\begin{aligned} \overline{A_{ik}} &= A_{ik} \\ \overline{f_{ik}} &= f_{ik} \\ \sigma_{A_{ik}} &= \frac{A_{jk}}{2} \\ \sigma_{f_{ik}} &= \min_{existing j} \left| \frac{f_{jk} - \overline{f_{jk}}}{2} \right| \\ n_i &= 1 \end{aligned} \quad (3)$$

Eq. 3 shows that new cluster is centred on the considered features. Dispersion initialisation of each parameter is set to half the minimum size between the considered parameter value and the existing cluster parameter average values. For the first cluster,  $\sigma_{A_{0k}} = \frac{A_{0k}}{2}$  and  $\sigma_{f_{0k}} = \frac{f_{0k}}{2}$ . Training algorithm is shown in Fig. 8.

- If the sensor in detection phase, an alert is generated as shown in Fig. 9.

## IV. RESULTS AND DISCUSSION

In this section classifications results, computation time and characteristic system consumption are presented, followed of a discussion.

### A. Classification methodology

For evaluating the performance of the fault detector, a rotating test bench has been used, having a rotor that can be unbalanced (Fig. 5) by adding asymmetrical weights.

Training has been done with a balanced rotor, at 3 different speeds during 1 hour for each speed. 2 sample rates have

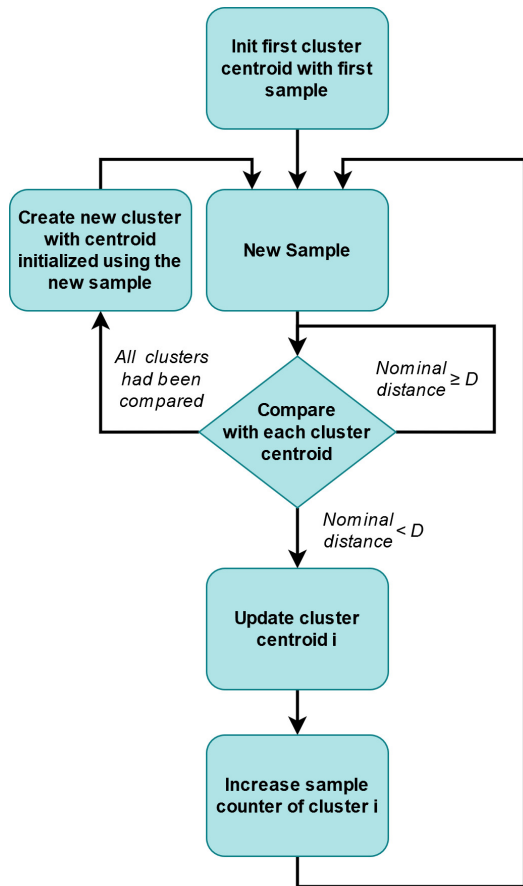


Fig. 8. K-Means training algorithm.

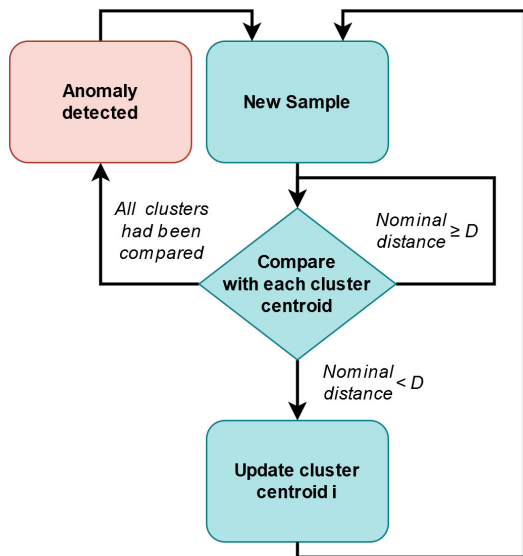


Fig. 9. K-Means anomalies detection algorithm.

been evaluated using an identical sequence : lowest one is 200 Hz, highest one is 500 Hz. FFT is computed on 512 samples sequences, thus feature extraction is performed every

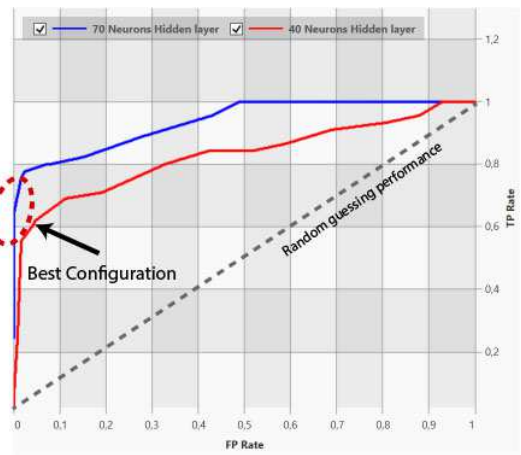


Fig. 10. ANN ROC curves.

2.5 seconds when sampling at 200Hz and approximately 1 second when sampling at 500Hz. Consequently, 4320 samples have been used for the unsupervised learning of the system at 200Hz and 10800 samples have been used when using a 500Hz sample rate.

After learning, a test sequence of 1 hour has been presented to the classification system. In this test, half of the samples correspond to situations learned during training and half of the samples have not been presented before : these last ones have either a different speed or are using an unbalanced rotor.

Classification system can be tuned using  $D_{Max}$ , as this a parameter adjusting cluster belonging tolerance. If a sample is classified correctly in a known cluster, true positive counter is incremented. If a sample is classified correctly as an unknown situation, true negative counter is incremented. If a sample is classified incorrectly in a known cluster, false positive counter is incremented. If a sample is classified incorrectly as an unknown situation, false negative counter is incremented.

### B. Classification results

For evaluating the performance of the classification embedded system, Receiver Operating Characteristic (ROC) curve and average precision (AP) are used as proposed in [18]. ROC curve is presented in Fig. 11. AP of the classification system is presented in Table I.

$D_{Max}$	0.25	0.5	0.75	1	1.25	1.5
Average Precision	0.73	0.88	0.88	0.88	0.88	0.88

TABLE I  
CLASSIFICATION AVERAGE PRECISION DEPENDING ON  $D_{Max}$ .

### C. Computation time

In this section, ARM M4F embedded controller used for classification is running at full 48 MHz speed. Embedded processing times for classification of 512 samples are the following ones:

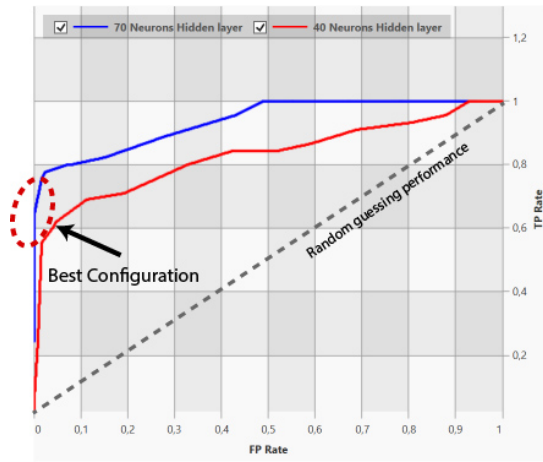


Fig. 11. ANN ROC curves.

- Feature extraction :  
Fast Fourier Transform:  $3.4ms$ .  
Peak Detection:  $250\mu s$ .
- K-Means Classification :  $30\mu s$ .

Considering classification is triggered one time every second at a  $500Hz$  sample rate, it leads to  $0.37\%$  average usage of the *ARM M4F* embedded processor for this task. Considering classification is triggered one time every 2.5 seconds at  $200Hz$ , it leads to a  $0.15\%$  average usage of the embedded processor for this task.

#### D. Power consumption

Power consumption is divided in 2 parts, first one is the always on ultra-low power part composed of : a piezo-electric sensor coupled with it charge amplifier, a low pass anti-aliasing active filter, and the sensor controller including the ADC for signal sampling and buffering. Power consumption of this always on part is equal to  $32\mu A$  when sampling at  $200Hz$  and  $77\mu A$  when sampling at  $500Hz$ .

Second part is the classification process. *ARM M4F* processor is awoken consuming  $3.4mA$ . Fortunately, this power consumption only happens during FFT and classification computation :  $0.37\%$  of the overall time at a sample frequency equal to  $200 Hz$ , leading to an average power consumption of  $12\mu A$ . At a sampling frequency of  $500 Hz$ , this leads to an average power consumption of  $31.4\mu A$ .

Finally, average overall current consumption of the system is  $44\mu A$  at  $200 Hz$  and  $98.4\mu A$  at  $500 Hz$ . This leads to a power consumption of less than  $80\mu W$  at  $200 Hz$ . Using a standard CR2032 coin cell having a capacity of  $720mW$ , our system can run during more than one year in always on learning and detection.

#### E. Discussion

#### V. CONCLUSION

In this paper, a low power unsupervised embedded learning implementation using a *TI CC2652* system on chip is proposed for automatic industrial monitoring and fault detection. Reference situation is learned using an embedded implementation of K-means algorithm. After a training period, device is able to detect abnormal vibrations patterns and is also able to fine tune its cluster according to the latest data.

Architecture of the system is first presented, then embedded processing algorithms composed of feature extraction and k-means algorithm are detailed, and finally an implementation on a vibrations simulator machine is finally described.

Results show that faults can be detected with an accuracy of  $XX\%$  using less than  $0.15\%$  of average embedded processor resources on a *ARM M4F* with an average consumption of  $80\mu W$ .

This smart sensor can be used for Industrial Internet Of Things (IoT) applications, thanks to its very interesting battery life (more than one year using a single CR2032 coin cell battery). It could also be used for many other applications in IoT considering its very low cost (less than  $10\%$ ) and power.

#### ACKNOWLEDGMENT

The authors would like to thank the reviewers for their insightful comments and advices, Pôle INPS of Toulon University and the Embedded Electronics Technology Platform of Toulon University named SMIoT (Scientific Microsystems for Internet of Things) [?] for its support in the achievement of this smart FES.

#### REFERENCES

- [1] E. Elazab, T. Awad, H. Elgamel, and B. Elsouhily, "A cloud based condition monitoring system for industrial machinery with application to power plants," in *2017 Nineteenth International Middle East Power Systems Conference (MEPCON)*, 2017, pp. 1400–1405.
- [2] U. Antao, A. Dibazar, J. Choma, and T. Berger, "Low power false positive tolerant event detector for seismic sensors," in *2013 IEEE SOI-3D-Subthreshold Microelectronics Technology Unified Conference (S3S)*, 2013, pp. 1–2.
- [3] M. Fourniol, V. Gies, V. Barchasz, E. Kussener, H. Barthelemy, R. Vauch, and H. Glotin, "Analog ultra low-power acoustic wake-up system based on frequency detection," in *2018 IEEE International Conference on Internet of Things and Intelligence System (IOTAIS)*, 2018, pp. 109–115.
- [4] Y. Kwon, Y. P. Cho, J. Yang, J. Chung, K. Shin, J. Han, C. Kim, C. Lyuh, H. Kim, I. Jeon, and M. Choi, "Ai 32tflops autonomous driving processor on ai-ware with adaptive power saving," in *2019 International SoC Design Conference (ISOCC)*, 2019, pp. 174–175.
- [5] A. Anilkumar, A. P. A. S. Nalluri, A. Santhanam, and P. D. Kedhari, "A low power artificial intelligence processor for autonomous mobile robots," in *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, 2019, pp. 495–499.
- [6] N. Goux and F. Badets, "Review on event-driven wake-up sensors for ultra-low power time-domain design," in *2018 IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS)*, 2018, pp. 554–557.

- [7] G. Lu, D. De, M. Xu, Wen-Zhan Song, and J. Cao, "Telosw: Enabling ultra-low power wake-on sensor network," in *2010 Seventh International Conference on Networked Sensing Systems (INSS)*, 2010, pp. 211–218.
- [8] R. Jafari and R. Lotfian, "A low power wake-up circuitry based on dynamic time warping for body sensor networks," in *2011 International Conference on Body Sensor Networks*, 2011, pp. 83–88.
- [9] M. Fourniol, V. Gies, V. Barchasz, E. Kussener, H. Barthelemy, R. Vauch, and H. Glotin, "Low-power wake-up system based on frequency analysis for environmental internet of things," in *2018 14th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA)*, 2018, pp. 1–6.
- [10] A. Rubino, M. Payvand, and G. Indiveri, "Ultra-low power silicon neuron circuit for extreme-edge neuromorphic intelligence," in *2019 26th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, 2019, pp. 458–461.
- [11] M. Fourniol, V. Gies, V. Barchasz, E. Kussener, and H. Glotin, "Applications of an ultra low-power analog wake-up detector for environmental iot networks and military smart dust," in *2018 IEEE International Conference on Internet of Things and Intelligence System (IOTAIS)*, 2018, pp. 16–22.
- [12] G. K. Durbhaka and S. Barani, "Fault behaviour pattern analysis and recognition," in *2016 International Conference on Information Science (ICIS)*, 2016, pp. 191–193.
- [13] F. Jia, Y. Lei, S. Xing, and J. Lin, "A method of automatic feature extraction from massive vibration signals of machines," in *2016 IEEE International Instrumentation and Measurement Technology Conference Proceedings*, 2016, pp. 1–6.
- [14] M. M. Etefagh, M. Ghaemi, and M. Yazdanian Asr, "Bearing fault diagnosis using hybrid genetic algorithm k-means clustering," in *2014 IEEE International Symposium on Innovations in Intelligent Systems and Applications (INISTA) Proceedings*, 2014, pp. 84–89.
- [15] C. Yiakopoulos, K. Gryllias, and I. Antoniadis, "Rolling element bearing fault detection in industrial environments based on a k-means clustering approach," *Expert Systems with Applications*, vol. 38, no. 3, pp. 2888 – 2911, 2011. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0957417410008791>
- [16] X. B. Liu, B. B. Deng, and L. N. Shen, "The improved k-means cluster analysis on diagnosis data fusion of the aero-engine," in *Mechanical Science and Engineering III*, ser. Applied Mechanics and Materials, vol. 328. Trans Tech Publications Ltd, 7 2013, pp. 463–467.
- [17] R. B. Randall, *Vibration-based condition monitoring: industrial, aerospace and automotive applications*. John Wiley & Sons, 2011.
- [18] A. K. Nandi and H. Ahmed, *Classification Algorithm Validation*. IEEE, 2019, pp. 307–319. [Online]. Available: <https://ieeexplore.ieee.org/document/8958927>