



HAL
open science

Machine learning based real time predictive maintenance at the edge for manufacturing systems: A practical example

Nicolas Ringler, Dominique Knittel, Jean-Christophe Ponsart, Mohammed Nouari, Abderrahmane Yakob, Daniel Romani

► To cite this version:

Nicolas Ringler, Dominique Knittel, Jean-Christophe Ponsart, Mohammed Nouari, Abderrahmane Yakob, et al.. Machine learning based real time predictive maintenance at the edge for manufacturing systems: A practical example. IEEE IAS Global Conference on Emerging Technologies, GlobConET 2023, May 2023, London, United Kingdom. 10.1109/GlobConET56651.2023.10150033 . hal-04255591

HAL Id: hal-04255591

<https://hal.science/hal-04255591>

Submitted on 29 Feb 2024

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.

Machine learning based real time predictive maintenance at the edge for manufacturing systems: a practical example

Nicolas Ringler
University of Lorraine, InSIC - LEM3
St Dié des Vosges, France
and
iDEMoov, Entzheim, France
nicolas.ringler@idemooov.fr

Dominique Knittel
University of Lorraine, InSIC - LEM3
St Dié des Vosges, France
and
University of Strasbourg, Strasbourg,
France
knittel@unistra.fr

Jean-Christophe Ponsart
University of Lorraine, CRAN
Nancy, France
Jean-christophe.Ponsart@univ-lorraine.fr

Mohammed Nouari
University of Lorraine, InSIC - LEM3
St Dié des Vosges, France
mohammed.nouari@univ-lorraine.fr

Abderrahmane Yakob
iDEMoov, Entzheim, France
abderrahmane.yakob@idemooov.fr

Daniel Romani
md6, Entzheim, France
daniel.romani@md6.fr

Abstract—Edge Computing is becoming more and more essential for the Industrial Internet of Things (industrial IoT) for predictive maintenance in the industry 4.0 framework. The transition from central to distributed approaches (edge nodes), will enhance the capabilities of handling real-time big data from IoT by ensuring low latency and high bandwidth. Moreover, with the developed architecture the possibility is given to have distributed Machine Learning (ML) by enabling edge devices to learn local ML models. Therefore, the performances of the global diagnostic models can be improved. In this paper, the developed setup is composed of PC's, NVIDIA Jetson Nano Developers kits (for edge computing), and a smartphone for real-time displaying. The implemented real-time supervised machine learning approaches are applied on an industrial oil-injection screw compressor instrumented with vibration sensors. Time domain features are calculated online with the help of sliding windows and the features are automatically classified. Embedded in the equipment, the used algorithms obtained very good real-time diagnostic performances.

Keywords— predictive maintenance, diagnostic, supervised machine learning, time series, embedded edge computing.

I. INTRODUCTION

Many academic research works and industrial R&D projects are focused on diagnostics and predictive maintenance (Prognostics and health management, PHM) [1] [2]. Also, several leading scientific journals are specifically dedicated to this research field involving many researchers and engineers from different disciplines: general computing, computer networks and IoT (Internet of Things), advanced statistics, signal processing, Artificial Intelligence, data science and data mining, intensive computing, embedded systems, etc..

Some diagnostic approaches are specifically designed or adapted for application fields and sectors such as manufacturing and production systems [3], industry 4.0 [4], material/tools interaction in milling [5] [6] [7], aeronautic sector [8], energy [9] [10], health [11], mobility/transport

[12], waste management/wastewater treatment plants [13], rotating machines [14], ...

Predictive diagnostics and maintenance approaches are traditionally based on different methods:

- technical expertise [15],
- *model-driven* approaches [16]. Physical models (phenomenological models) are constructed from observations and measurements,
- *data-driven* approaches [16] [17] [18]. Data are acquired from sensors or data bases, usually without physics knowledge. The data can be handled with powerful statistic's approaches [15] [19], artificial intelligence tools and *data mining* [20],
- hybrid methods [16] [21] combining the previous methods.

In this work, machine learning based diagnostic models are developed and implemented for time series (coming from sensors and data bases) with real time data streaming environment and IoT devices connections [22-25]. All the developed algorithms can be embedded.

Feature selection approaches [26] have also been implemented in our framework. Indeed, these dimensionality reduction techniques lead to better learning performances and lower computational time.

II. STUDIED INDUSTRIAL SYSTEM

A. Studied System

From the 2021 PHM Data Challenge [27], recorded vibrations data of an industrial oil-injection screw compressor are used in this work for real-time diagnosing of the different plant states: 1-Normal, 2-Unbalance, 3-Belt-Looseness, 4-Belt-Looseness High, 5-Bearing-fault. The PHM Data Challenge is an international competition held annually by the PHM Society, a world-renowned academic society in the field of diagnostics, prognostics and health management.

The compressor is illustrated in Fig. 1: a vibration sensor is attached at the 15kW motor (with an axis rotating speed of 3600 rpm (rotations per minute)), the second vibration sensor is attached at the screw (with an axis rotating speed of 7200 rpm). The two-channel data have a sampling frequency of 10544 Hz.

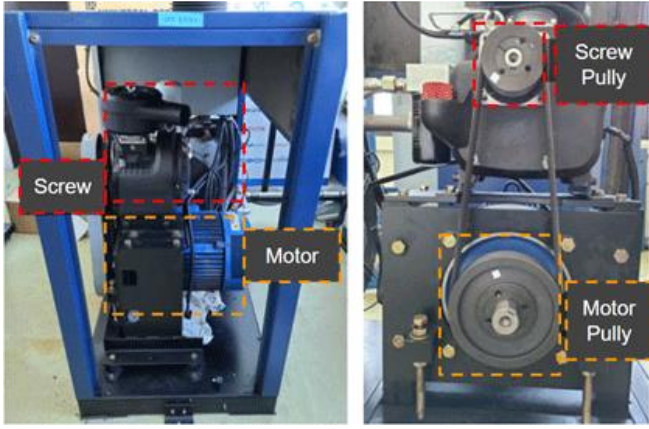


Fig. 1. Illustration of the studied oil-injection screw compressor

The significant difficulty lies in obtaining separately the different defaults, the real time adding also severe constraints.

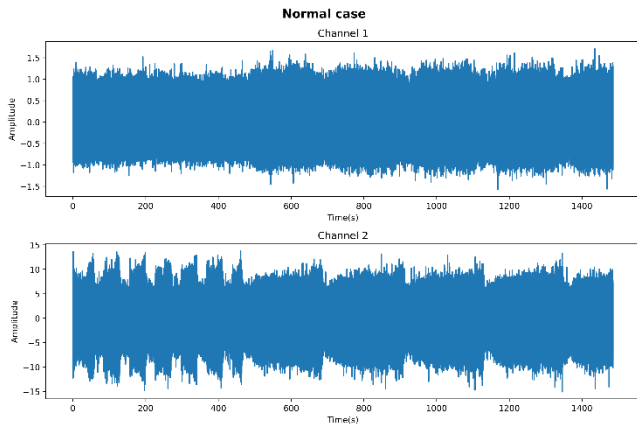


Fig. 2. Signals of the two vibration sensors in the normal operating mode.

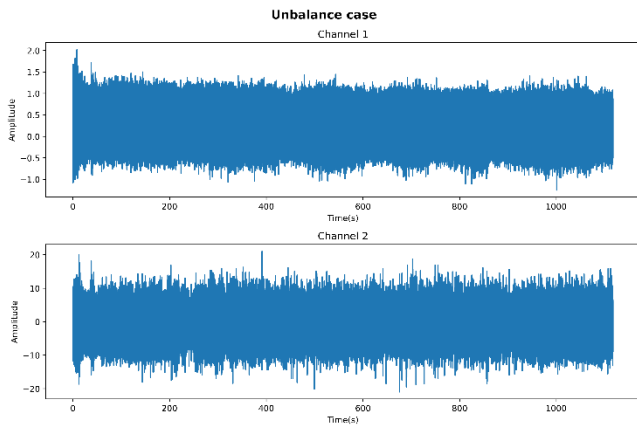


Fig. 3. Signals of the two vibration sensors in the unbalanced operating mode.

To address this issue, sliding windows are filled in by measurements from the two vibration sensors, the windows being or not overlapped. The impacts of the windows length

and overlapping rate on the diagnostics performances (in real time) have also been analysed in this work.

The features are calculated for each window, which lead to a time serie for each feature, each point of the time serie corresponding to a window.

B. Features calculation

After a first data processing (low pass filtering), the features are calculated in time domain for the measured vibrations. Each signal, for sensor 1 or 2, is called hereafter x_k , at sample time k .

Calculated features are : maximum and minimum of $x(t)$, amplitude range, median value, maximum of the absolute value, inter quartile range, inter decile range, average value, average of the absolute value, average of the absolute value of the derivative signal, variance, energy, energy of the centred signal (1), energy of the derivative signal (2), skewness (3), kurtosis (4), moment order i (5), Shannon entropy (6), signal rate (7), etc. Some of the calculated time domain features are detailed below:

- Energy of the centred signal: (1)

$$E_c = \sum_{k=1}^N (x_k - \text{mean}(x))^2$$

- Energy of the derivative signal: (2)

$$E_d = \sum_{k=1}^{N-1} \left(\frac{dx_k}{dt} \right)^2$$

- Skewness: (3)

$$S = \frac{E(x - \text{mean}(x))^3}{\text{Var}^{3/2}}$$

- Kurtosis: (4)

$$K = \frac{E(x - \text{mean}(x))^4}{\text{Var}^2}$$

- Moment order i ($i = 5 : 10$): (5)

$$m_i = \frac{E(x - \text{mean}(x))^i}{\text{Var}^{i/2}}$$

- Shannon entropy: (6)

$$E_S(x) = - \sum_{k=1}^N x_k^2 * \log_2(x_k^2)$$

- Signal rate: (7)

$$\tau = \frac{\max(x_{k=1:N}) - \min(x_{k=1:N})}{\text{mean}(x)}$$

In this application, features calculated in frequency domain (by using Fast Fourier Transformation, FFT) are not used, because of the possible nonlinearities in the vibration signals.

All the calculated features are normalized and stored in a table whose lines and columns respectively represent the

experimental number (also called instance) and the associated feature values. The normalization can be made as follows:

$$feature_{norm} = \frac{feature - \text{mean}(feature)}{\text{var}(feature)} \quad (8)$$

This normalization leads to: (9)

$$\begin{cases} \text{mean}(feature_{norm}) = 0 \\ \text{var}(feature_{norm}) = 1 \end{cases}$$

In our study, we used the normalization given by the Equation (10). The goal is to consider also the information of the standard deviation at each time.

Normalized features: (10)

$$feature_{norm} = \frac{feature - \min(feature)}{\max(feature) - \min(feature)}$$

III. DEVELOPED SETUP

An innovate platform (hardware + software) for intelligent real time predictive maintenance has been developed in this work for industry 4.0 applications, with a distributed architecture. The transition from central to distributed topology with edge nodes will on one hand enhance the capabilities of handling real-time big data from IoT by ensuring low latency and high bandwidth, and on the other give the possibility to have distributed Machine Learning (ML) by enabling edge devices to learn local ML models. The setup is composed of several pc's, two or more NVIDIA Jetson Nano Developers kits for computing on the edge, a smartphone for real-time displaying, see Fig. 4.



Fig. 4. Setup developed within the framework of this study

The developed intelligent maintenance framework has been tested and evaluated with the help of this platform. The measured data (time series) may be collected from the connected sensors (at the edge) or from databases located in a connected device (in this paper the database from the 2021 PHM challenge was stored in a connected pc and feeds in real time the setup).

In our network architecture, the MQTT communication protocol is used for information and message transmission (raw signals, filtered signals, features, diagnostics message and results, ...) between components and devices.

In this setup, the Mosquitto broker [28] enables the connection between the IoT devices in real time. It is implemented into a Raspberry 4 board connected at the local network via a router. The interconnection between various sites has been made via a VPN (Virtual Private Network).

In order to obtain a complete diagnostics pipeline, several libraries are required, for example CuPy, scikit-learn or Pytorch. These libraries should be operated with the GPU (Graphics Processing Unit) accelerator and also with the possibility to an implementation on an edge node. Therefore, the Machine Learning Container for Jetson and Jetpack has been used, including all necessary libraries allowing an easy access to the GPU. Moreover, several modules provided a GPU integration specifically dedicated to Deep Learning for Edge Computing.

The NVIDIA Jetson nano board has been chosen for its very low power consumption (10W maximum), and embedded applications, while having a sufficient RAM (Random Access Memory) size for our needs.

In order to store in real time all the information circulating on the network, and to analyse them offline, a rapid writing into a database specifically dedicated for time series is required. To perform this, InfluxDB is used for many connected edge IoT devices. InfluxDB additionally includes an integrated dashboard, and can be easily connected to others HMI (Human Machine Interface) such as Grafana (which is also used in this work). For example, a real time display of some online calculated features is given in Appendix (see Fig. 13).

The developed framework is equipped with many high performance functionalities with many parameters: data cleaning, length of the sliding windows and its overlapping rate, normalizing or not of the signal and the features, choose the selection (and classification) algorithm of the features, choose the supervised machine learning model, the metrics for the performance and robustness evaluation, ...

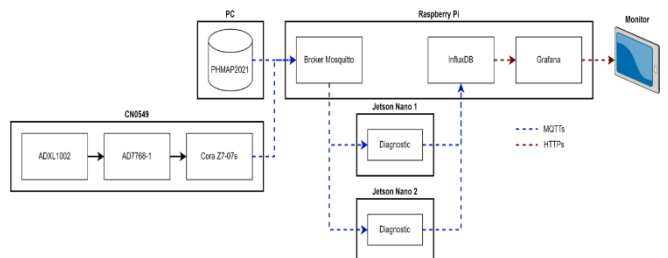


Fig. 5. Architecture of the implemented framework with vibration sensor CN0549

This architecture (Fig. 5) illustrates the multiple data sources: PHMAP 2021 conference Data or acquisition with sensors in real time from an industrial machine. The shown Database can be hosted in a PC or in a cloud [29].

IV. OBTAINED RESULTS OF IMPLEMENTED ALGORITHMS

Different machine learning models are implemented and tested in real time. The model training can be made offline or online. Nevertheless, we choose to test it online in real time.

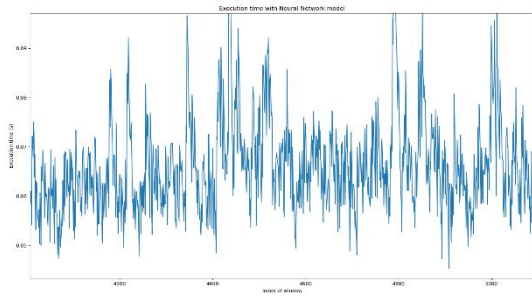


Fig. 6. Neural Network based diagnostic model implemented into the Jetson nano without GPU : computational run time for the sliding windows

Fig. 6 shows the measured execution time of a neural network based diagnostic model implemented into the NVIDIA Jetson nano board without using GPU, for each sliding window (in x coordinate are numbered the windows). These results illustrate that the execution time is nearly constant and takes 65 ms, less than the duration time of each window (several hundred ms, depending on the user's parameters choice).

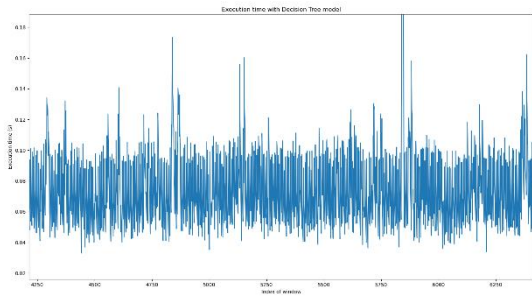


Fig. 7. Decision Trees based diagnostic model implemented into the Jetson nano without GPU : computational run time for the sliding windows

For Fig. 7, this execution time reaches 80ms for Decision Trees under the same implementation conditions. As expected, this time is little higher, NVIDIA Jetson nano board being specifically optimized for Neural Networks implementation.

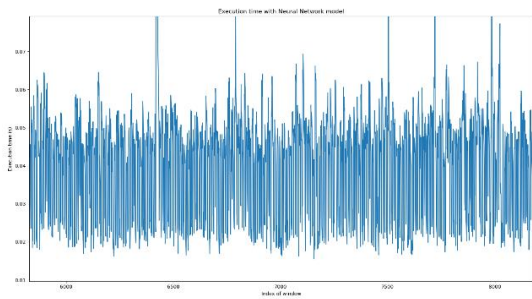


Fig. 8. Neural Network based diagnostic model implemented into the CPU without GPU : computational run time for the sliding windows

By contrast, the execution time of a Neural Network based diagnostic model implemented into a CPU/PC (Central Processing Unit), also without GPU, takes 40ms as illustrated in Fig. 8.

With the developed architecture, the possibility is given to have distributed Machine Learning (ML) by enabling edge devices to learn local ML models. Distributed tasks can also be implemented, many combinations are possible. For example, the Jetson nano boards located close to the plant and sensors can be dedicated to sensor data processing (data cleaning and features calculation), whereas the diagnostic model runs in a central CPU connected remotely to all boards and sensors via network data streaming. Moreover, real-time setting and results displaying are done locally via touchscreens and remotely via smartphones (see Fig. 4).

Several supervised machine learning models [30] have been implemented and tested : Neural Networks, Decision Trees, Random Forest, KNN (K-nearest neighbors), SVM (Support Vector Machine), Gaussian Naive Bayes. 70% of the labelled data are used for the training (online or offline training) of the models, whereas 30% of the labelled data are used for the model tests in real-time (necessarily with the help of sliding windows). The workflow of the ML model training is illustrated in Appendix (Fig. 12).

Fig. 9 shows the obtained confusion matrix of the Neural Network based diagnostic implemented into the NVIDIA Jetson nano board (Fig. 9 is associated with the tests illustrated on Fig. 6). It can be seen that the results are very hopeful.



Fig. 9. Confusion matrix of the Neural Network based diagnostic model

For the studied oil-injection screw compressor, the best diagnostic results are obtained with the Neural Networks model : excellent accuracy rates are achieved, as with the Random Forest, but with the smallest execution time (in real time with sliding windows).

Within the developed framework many diagnostic settings can be selected by the users, as for example:

- signal : x% training, 100-x% test,
- signal : can be normalized or not,
- window size : 512 or 1024 or 2048 samples,
- windows : without overlapping or with 50% overlapping,
- features: can be normalized or not, moreover all or only the "best" automatically selected features can be extracted,
- machine learning models: Neural Network, Decision Trees, Random Forest, KNN, SVM, Gaussian Naive

Bayes. One can also choose multi-models approaches,

- evaluation metrics: accuracy, error rate, sensitivity, specificity, precision, recall, F1 score, execution time.

By considering all calculated features, some models can lead to overfitting. Only the most meaningful features should be used. Therefore we implemented and tested different features selection algorithms into the framework: relief, mRMR (minimum redundancy maximum relevance), SPEC, Decision Trees, Random Forest.

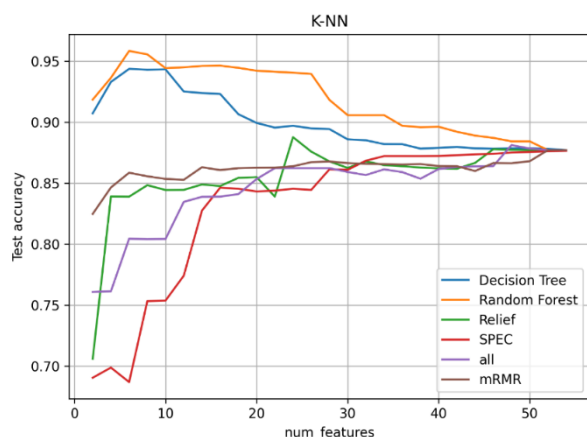


Fig. 10. Accuracy of the KNN model according to the number of the ordered features and their selection method

Fig. 10 shows the accuracy obtained with a KNN based diagnostic model for several features selection methods. The x-axis represents the number of ordered features used by the diagnostic model. Only the features are normalized, not the sensor signals. It seems clearly that the best features selection algorithm is the Random Forest, and the number of features should not exceeds 25. More performance index should be considered for the features selection, for example the robustness to signal noise, the execution time, ... In fact, the features selection is a multi-objectives problem.

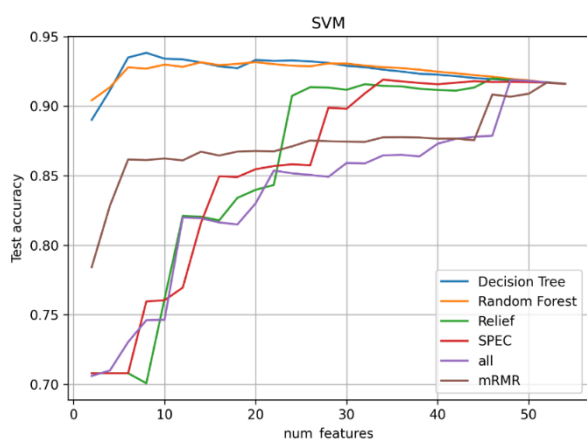


Fig. 11. Accuracy of the SVM model according to the number of the ordered features and their selection method

Fig. 11 highlights the accuracy obtained with a SVM based diagnostic model for several features selection methods. The best results are obtained for the Decision Trees selection algorithm. Its obtained performance is practically

constant, but a little less in comparison with the previous figure (Fig. 10). Between 8 and 30 features should be selected, depending also on other performance index (for example execution time).

V. CONCLUSION

This work concerns the development of a machine learning based predictive maintenance setup, with the associated framework, for industry 4.0 applications. The setup is composed of CPU's, NVIDIA Jetson Nano boards (for computing on the edge), screens and smartphones (for real-time display). The distributed architecture enables edge computing close to the sensors and equipments. The practical example of the industrial oil-injection screw compressor illustrates the developed approaches. Several supervised machine learning algorithms have been implemented and compared. To achieve this, features were firstly calculated from measured data and were automatically selected by features selection algorithms. From the prediction results, the Neural Network model seems to be the most efficient diagnostic algorithm for the presented application. In forthcoming work, the actual setup architecture will be extended to multi-sites. Moreover, parameter optimization [31] of the predictive maintenance framework, for expected performances (for example for low resources), will also be investigated.

ACKNOWLEDGMENTS

The authors wish to express their thanks and appreciation to members of the following organizations for their support: iDEMoov Company and Association Nationale de la Recherche et de la Technologie (ANRT), Cifre grant number of Nicolas RINGLER: 2022/0473

REFERENCES

- [1] Tania Cerquitelli, Nikolaos Nikolakis, Niamh O'Mahony, Enrico Macii, Massimo Ippolito, Sotirios Makris, Predictive Maintenance in Smart Factories, Springer 2021
- [2] J. Lee, Industrial IA, Springer, Feb. 2020
- [3] Diez, L., Marangé, P., Levrat, É., Regeneration Management Tool for Industrial Ecosystem, IFAC-paperOnline, Volume 50, Issue 1, July 2017, Pages 12950-12955
- [4] D'Emilia G., Gaspari A. Data Validation Techniques for Measurements Systems Operating in a Industry 4.0 Scenario a Condition Monitoring Application. Workshop on Metrology for Industry 4.0 and IoT, Brescia, 112-116, 2018.
- [5] D. Knittel, H. Makich,, M. Nouari, "Milling Diagnosis using Machine Learning Approaches ", International Survishno conference (Surveillance, Vibration Shock and Noise), Lyon, France, 2019.
- [6] L. Codjo, M. Jaafar, H. Makich, D. Knittel, M. Nouari, "Honeycomb Core Milling Diagnosis Using Machine Learning in the Industry 4.0 Framework", IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2018), Torino, Italy, 2018.
- [7] D. Knittel, H. Makich,, M. Nouari, "Milling Diagnosis using Artificial Intelligence Approaches ", Journal Mechanics & Industry, Volume 20, Number 8, 2019
- [8] Florent Forest, PhD thesis, Unsupervised Learning of Data Representations and Cluster Structures: Applications to Large-scale Health Monitoring of Turbofan Aircraft Engines, University Sorbonne Paris Nord, France, 2021
- [9] Simon Henriet, PhD thesis, On solving the non intrusive load monitoring problem in large buildings : analyses, simulations and factorization based unsupervised learning, Telecom Paris, France, 2020
- [10] P. Wira and I. Szilagyi, "An Intelligent System for Smart Buildings using Machine Learning and Semantic Technologies: A Hybrid Data-

- Knowledge Approach," IEEE International Conference on Industrial Cyber-Physical Systems (ICPS 2018), Saint-Petersburg, Russia, 2018
- [11] John Anderson Garcia Henao, PhD thesis, Green artificial intelligence to automate medical diagnosis with low energy consumption, University Côte d'Azur, France, 2022
- [12] Amina Alaoui Belghiti, PhD thesis, Prediction of abnormal situations by machine learning in a predictive maintenance context : Optimal transport theory for anomaly detection , University Paris-Saclay, France, 2021
- [13] Nagy Kiss A.M, PhD thesis, Analysis and synthesis of multiple models for diagnosis : application to a wastewater treatment plant. University of Lorraine, France, 2010.
- [14] Nishchal K. Verma, Al Salour, Intelligent Condition Based Monitoring, For Turbines, Compressors, and Other Rotating Machines; Springer 2020
- [15] Douglas Goodman, James P. Hofmeister, Ferenc Szidarovszky, Prognostics and Health Management: A Practical Approach to Improving System Reliability Using Condition-Based Data; Wiley 2019
- [16] Morteza Soleimani, Felician Campean, Daniel Neagu, Diagnostics and prognostics for complex systems: A review of methods and challenges, Volume37, Issue8, December 2021, Pages 3746-3778, Quality and Reliability Engineering International journal, Wiley
- [17] Yaguo Lei , Naipeng Li , Xiang Li, Big Data-Driven Intelligent Fault Diagnosis and Prognosis for Mechanical Systems, Springer 2023
- [18] Niu, Data-Driven Technology for Engineering Systems Health Management: Springer, 2017
- [19] Hoang Pham, Reliability and Statistical Computing; Springer 2020
- [20] Michael G. Pecht, Myeongsu Kang, Prognostics and Health Management of Electronics: Fundamentals, Machine Learning, and the Internet of Things; Wiley 2018
- [21] Omer F. Eker., Fatih Camci, and Ian K. Jennions, A New Hybrid Prognostic Methodology, International Journal of Prognostics and Health Management, 2019 009
- [22] Joao Gama *et al*, IoT Streams for Data-Driven Predictive Maintenance and IoT, Edge, and Mobile for Embedded Machine Learning; Springer 2021
- [23] Alireza Entezami, Structural Health Monitoring by Time Series Analysis and Statistical Distance Measures; Springer 2021
- [24] Alexandre Cury, Diogo Ribeiro, Filippo Ubertini, Michael D. Todd, Structural Health Monitoring Based on Data Science Techniques; Springer 2022
- [25] R. Anandan *et al*, Industrial Internet of Things: Intelligent Analytics for Predictive Maintenance; Wiley 2022
- [26] Urszula Stańczyk, Lakhmi C. Jain, Feature Selection for Data and Pattern Recognition, Springer, 2015
- [27] <http://phmap.org/data-challenge/>
- [28] The Journal of Open Source Software, Mosquitto: server and client implementation of the MQTT protocol, <https://joss.theoj.org/papers/10.21105/joss.00265>
- [29] D. Ning, J. Huang, J. Shen and D. Di, "A cloud based framework of prognostics and health management for manufacturing industry," 2016 IEEE International Conference on Prognostics and Health Management (ICPHM), Ottawa, 2016
- [30] Miroslav Kubat, An Introduction to Machine Learning, Springer 2017
- [31] F. Bennis, R. K. Bhattacharjya, Nature-Inspired Methods for Metaheuristics Optimization, Springer, Feb. 2020.

APPENDIX

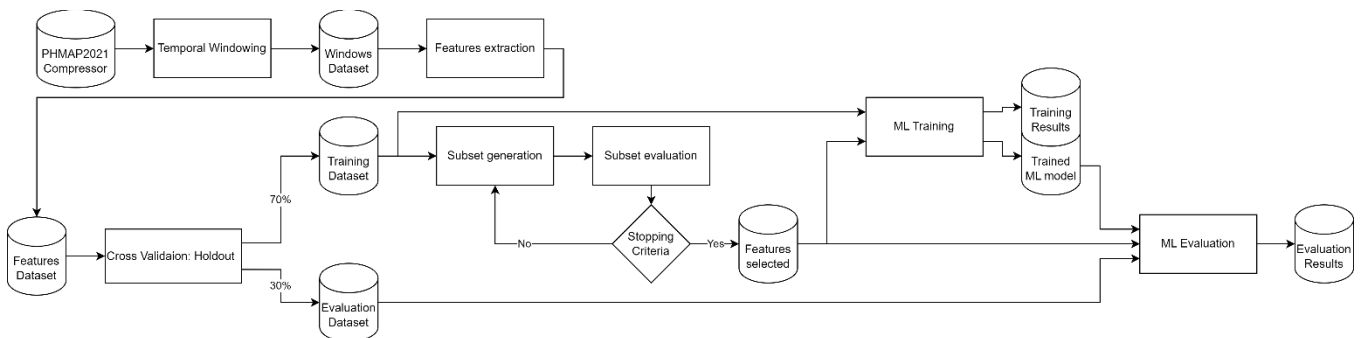


Fig. 12. Workflow of the ML model training

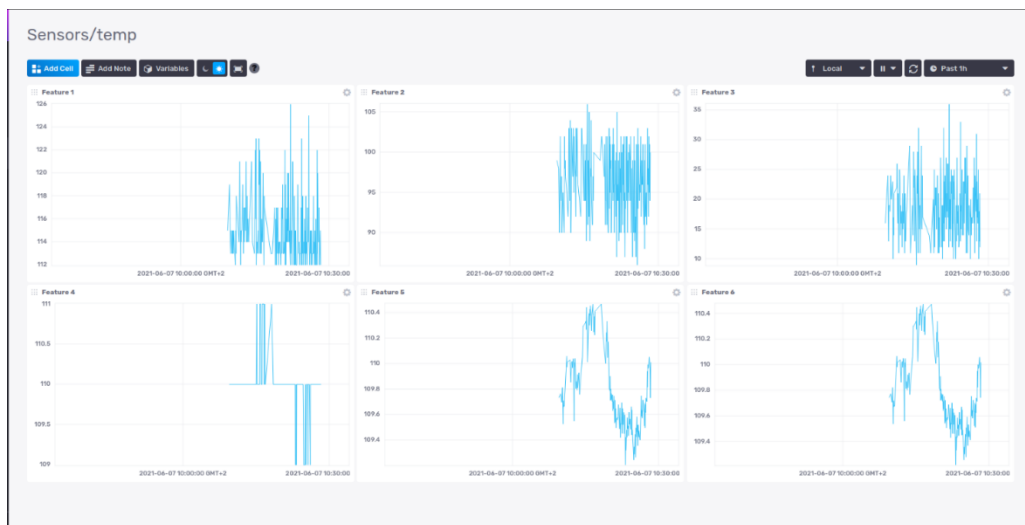


Fig. 13. Real time display : features 1 to 6 with InfluxDB