



HAL
open science

Diagnostic de défauts de systèmes cyber-physiques et manufacturiers : approches basées sur l'intelligence artificielle et par topologie distribuée

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

► To cite this version:

Nicolas Ringler, Dominique Knittel, Jean-Christophe Ponsart, Mohammed Nouari, Yakob Abderrahmane, et al.. Diagnostic de défauts de systèmes cyber-physiques et manufacturiers : approches basées sur l'intelligence artificielle et par topologie distribuée. Plate-Forme Intelligence Artificielle, PFIA'23, Conférence sur l'Apprentissage Automatique, CAp 2023, Jul 2023, Strasbourg, France. 2023. hal-04255645

HAL Id: hal-04255645

<https://hal.science/hal-04255645>

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.

Diagnostic de défauts de systèmes cyber-physiques et manufacturiers : approches basées sur l'intelligence artificielle et par topologie distribuée

Nicolas Ringler^{1,3}, Dominique Knittel^{1,5}, Jean-Christophe Ponsart², Mohammed Nouari¹, Yakob Abderrahmane³, Daniel Romani⁴

¹Université de Lorraine, CNRS, LEM3, GIP-InSIC, 88100 Saint-Dié-des-Vosges, France

²Université de Lorraine, CNRS, CRAN, 54000 Nancy, France

³iDEMooV, 67960, Entzheim, France

⁴md6, 67960, Entzheim, France

⁵Université de Strasbourg, Faculté de Physiques et Ingénierie, 67000 Strasbourg, France

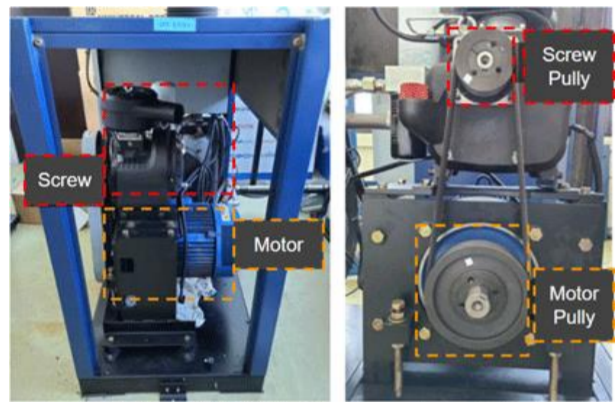
Résumé

Ce poster présente quelques résultats obtenus lors des travaux de thèse et des résultats antérieurs :

- Environnement logiciel : une première plateforme informatique de diagnostic temps réel a été développée. Elle est d'une topologie distribuée associant un pc, deux cartes NVIDIA Jetson Nano, Raspberry Pi 4, smartphone pour affichage. Des tests prometteurs de diagnostic ont été réalisés en implémentant un réseau de neurones. L'extension de la topologie en multi-sites est en cours.
- Algorithmes : des tests ont été menés à l'aide d'algorithmes d'apprentissage machine supervisé (arbres de décision, K-NN, ...). La méthodologie de sélection et réduction de features séries temporelles est en cours en vue de réduire la complexité des modèles.

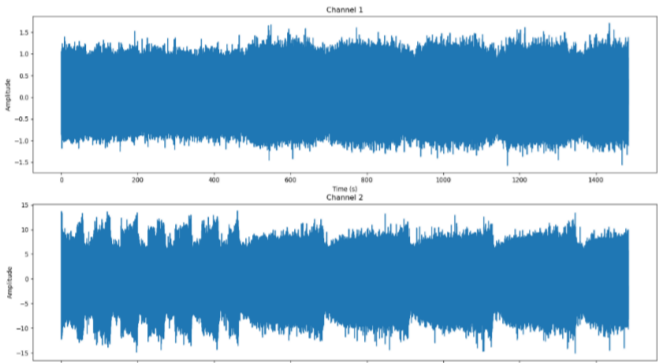
Objectif : Estimer l'état d'un compresseur parmi les cinq états possibles à partir des données.

- 1 état normal.
- 4 états défaillants : déséquilibre, relâchement de la courroie, important relâchement de la courroie, défaut de roulement.

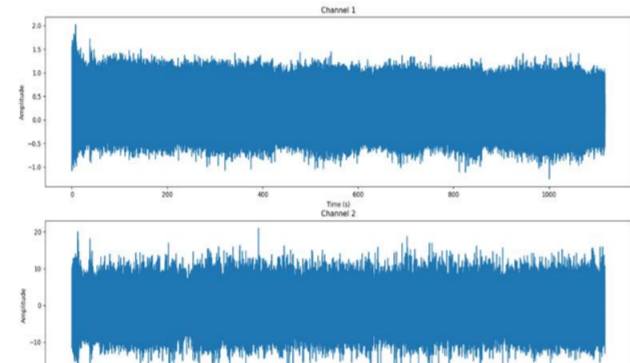


Source : <https://phmsociety.org>

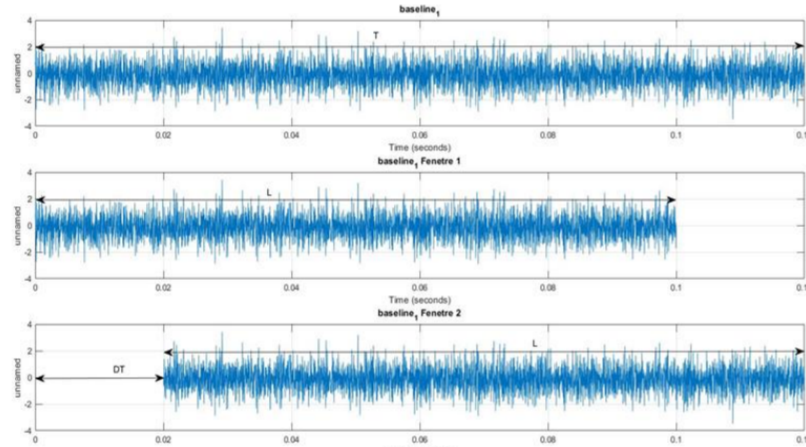
- Compresseur.
- 2 capteurs.
- Moteur : 15KW, 3600tr/min.
- Vis : 7200tr/min.
- Fréquence d'échantillonnage : 10 544Hz.



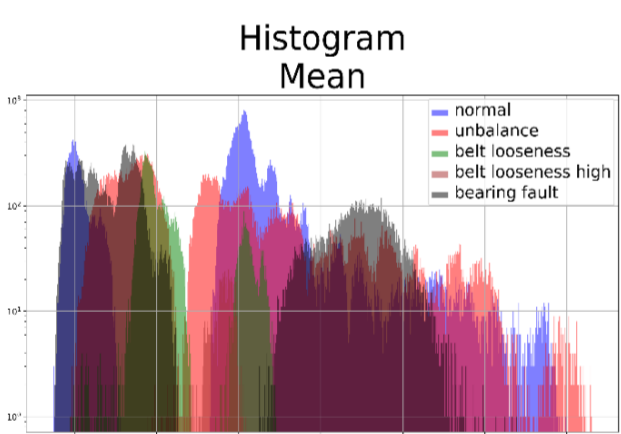
Signal temporel / cas normal



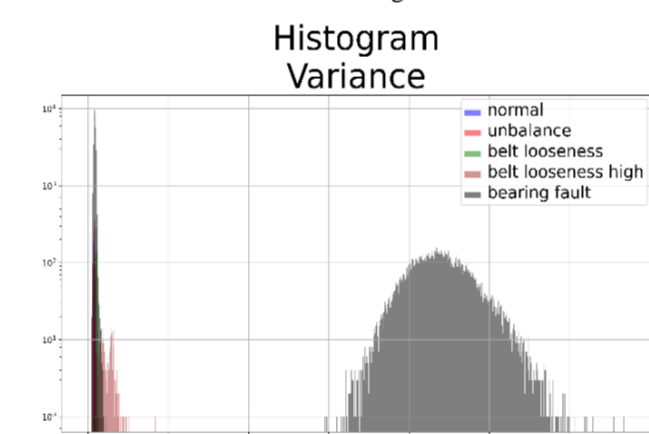
Signal temporel / cas déséquilibre



Création de fenêtres glissantes pour analyser de manière précise les signaux



Histogramme de la moyenne en base logarithmique.



Histogramme de la variance en base logarithmique.

- Utilisation de fenêtres glissantes (~100ms) avec ou sans recouvrement (overlapping).

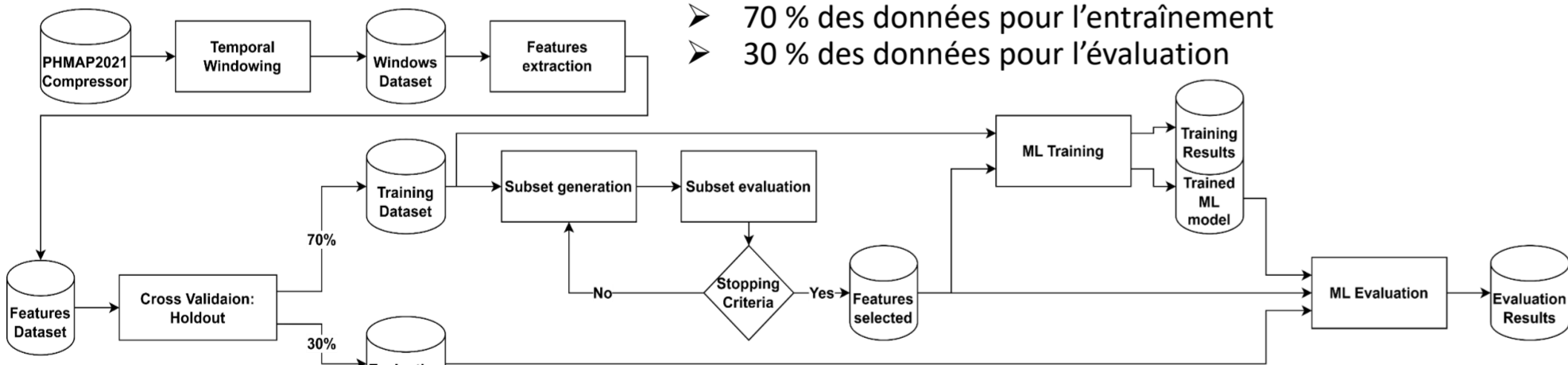
- Calcul de caractéristiques statistiques dans le domaine temporel : maximum et minimum, amplitude, médiane, énergie, skewness, kurtosis, entropie de Shannon, ...

- Dans notre cas, les distributions se chevauchent. On ne peut pas estimer directement l'état.

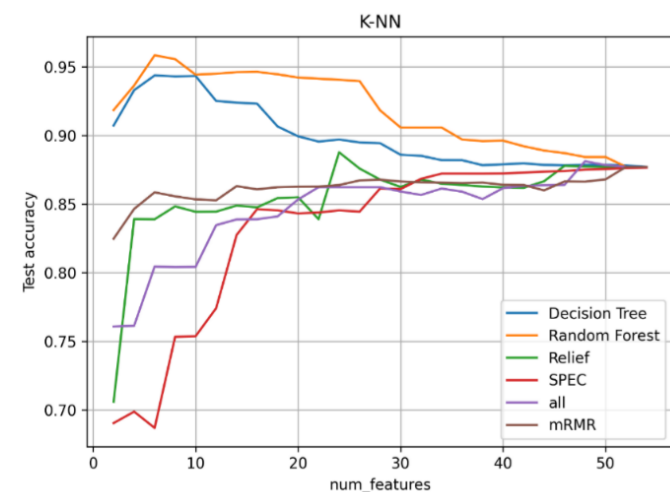
- Quelles caractéristiques sont les plus pertinentes pour maximiser les performances du modèle IA ?

Cross Validation : Holdout

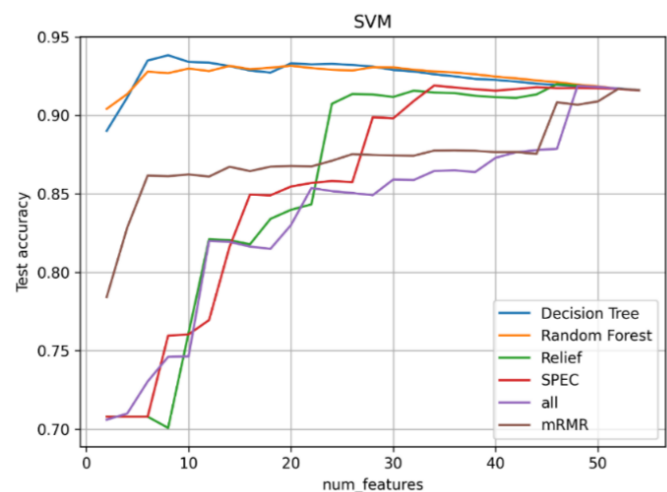
- 70 % des données pour l'entraînement
- 30 % des données pour l'évaluation



Workflow d'entraînement et de test des modèles Machine Learning avec utilisation de Feature Selection.



Précision du modèle k-NN en fonction du nombre de caractéristiques triées par des méthodes de sélection.



Précision du modèle SVM en fonction du nombre de caractéristiques triées par des méthodes de sélection.

- Pour les algorithmes FS Decision Tree et Random Forest, le classifieur K-NN obtient sa meilleure précision avec seulement **10 features**. Des performances similaires sont obtenues avec le classifieur SVM. Pour ces deux modèles, la réduction du temps d'entraînement est significative.

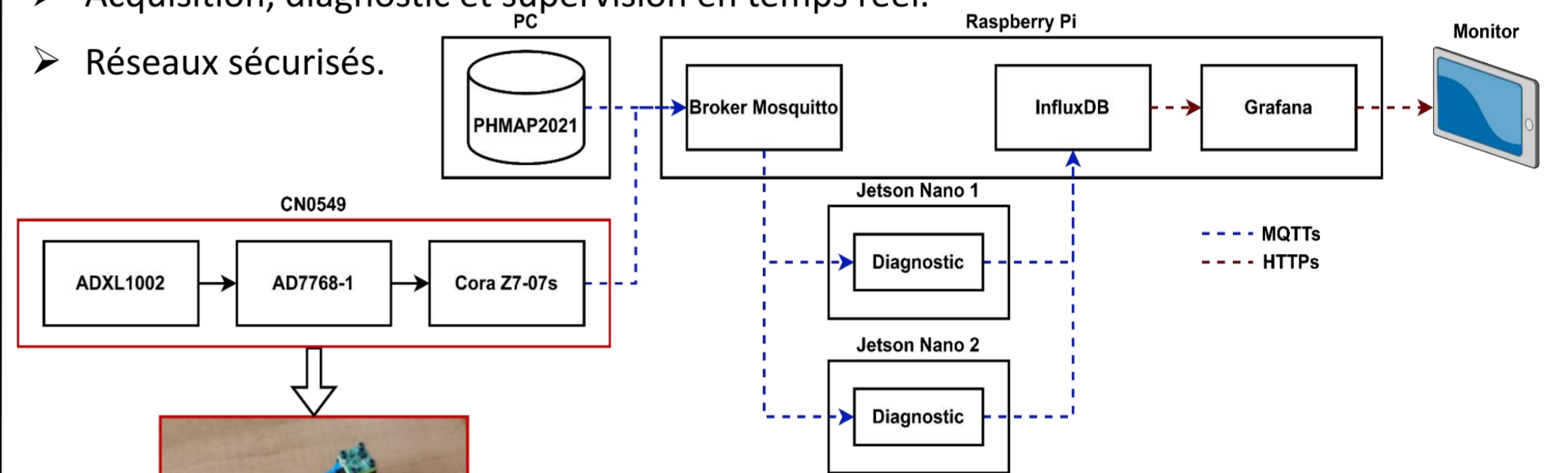
Publication des travaux

N. Ringler, D. Knittel, J.-C. Ponsart, M. Nouari, A. Yakob and D. Romani, "Machine Learning based Real Time Predictive Maintenance at the Edge for Manufacturing Systems: A Practical Example", IEEE IAS Global Conference on Emerging Technologies, Track Smart manufacturing and Industry 4.0, May 19-21, 2023, London.

N. Ringler, D. Knittel, J.-C. Ponsart, M. Nouari, A. Yakob and D. Romani, "Diagnostic de défauts de systèmes cyber-physiques et manufacturiers : approches basées sur l'intelligence artificielle et par topologie distribuée, congrès PFA – CAP (conférence sur l'apprentissage automatique), 3-5 juillet, Strasbourg.

Objectif : Permettre l'acquisition, le diagnostic et la supervision de systèmes en temps réel.

- Multisites, multi-plateformes et évolutif.
- Acquisition, diagnostic et supervision en temps réel.
- Réseaux sécurisés.



Implemented architecture

- **Acquisition des données** : chaîne d'acquisition vibrations CN0549, Analog Device.
- **Communication réseau local** : protocole MQTTv3.1.1 avec le broker Mosquitto.
- **Traitement des données** : image Docker développée par iDEMooV.
- **Stockage des données** : InfluxDB.
- **Supervision** : Grafana.

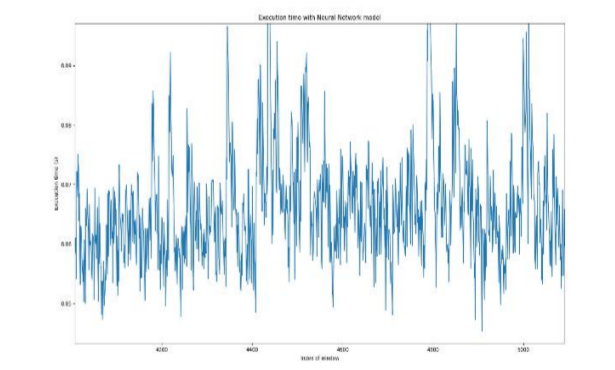
- Quel modèle utilisé pour du diagnostic en temps réel en Edge Computing ?

- Comparaison de deux modèles : Deep Neural Network et Decision Tree.

- Temps d'exécution pour des fenêtres glissantes (~100ms) :

Deep Neural Network :

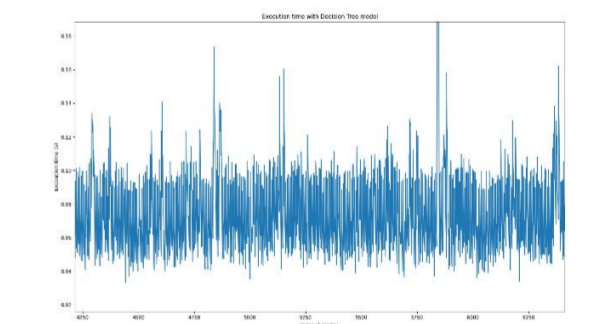
- Construit, entraîné et évalué avec PyTorch.
- Utilisation avec Data Streaming avec le ONNXruntime sur CPU.
- Temps d'exécution : ~65 ms.



Temps d'exécution du modèle Deep Neural Network : ~65 ms

Decision Tree :

- Construit, entraîné et évalué avec Scikit-Learn.
- Utilisation avec Data Streaming avec ONNXruntime sur CPU.
- Temps d'exécution : ~80 ms.



Temps d'exécution du modèle Decision Tree : ~80 ms

- Le réseau de neurones est plus rapide que le modèle Decision Tree sans GPU.

- Entraînement et test du réseau de neurones avec les données compresseur.



Matrice de confusion du modèle réseau de neurones sur la partition test Normal / Déséquilibre / Relâchement de la courroie Important relâchement de la courroie / Défaut de roulement.

- L'approche par réseaux de neurones est donc très intéressante pour du diagnostic en temps réel dans une architecture Edge Computing.

Conclusion

- L'étude sur la sélection des caractéristiques a montré la possibilité de réduire le nombre de caractéristiques tout en gardant (voire améliorant) les performances des classificateurs. On peut ainsi réduire le temps d'entraînement / test / utilisation, réduire la complexité et améliorer la compréhensibilité des modèles.
- De plus, l'utilisation d'un réseau de neurones profonds pour estimer l'état du compresseur donne des résultats encourageant tant au niveau de la précision du modèle, que de son temps de calcul (inférieur à l'arbre de décision) en embarqué.