AMIRAL TECHNOLOGIES

BENCHMARK

Comparaison de modèles de prédiction de pannes

04/2021

Sébastien Le Gall
CTO
Thibaut Le Magueresse
Data Scientist

Résumé

La prédiction de pannes d'équipements industriels connectés en mode aveugle se base sur des algorithmes de détection d'anomalies ou de nouveautés sur des données de type séries temporelles brutes ou bien transformées. Un certain nombre d'algorithmes a été comparé sur un ensemble de bases de données de séries temporelles qui présentent une cyclicité [1]. L'évaluation des performances s'appuie sur les courbes ROC [2] et Precision-Recall [3] pour calculer un score de généricité et de performance relative des modèles. Il en ressort que la problématique de prédiction de pannes n'est pas un sujet facile mais que l'association des algorithmes d'Amiral Technologies à savoir le générateur de caractéristiques (features) DiagFeatures(DF) et le détecteur d'anomalies BlindFaultDetector(BFD) semble être le meilleur compromis généricité-performance-vitesse.
Une discussion sur la pertinence de ces scores lors du déploiement des modèles en production, nous conduit à envisager un système de scoring qui tient compte des contraintes opérationnelles sur lequel nous avons également noté les modèles.

Introduction

La prédiction de pannes dans les équipements industriels connectés consiste à anticiper une panne avant qu'elle n'arrive ou à la détecter le plus tôt possible. Les approches les plus récentes pour résoudre ce problème s'appuient sur des méthodes de traitement du signal et de machine learning en créant des modèles appris sur des données historiques. Les types de données générées par les équipements industriels peuvent être de natures différentes mais ce document se focalise sur un sous-ensemble représentant néanmoins de grandes familles d'équipements, à savoir les séries temporelles cycliques.

Pour créer des modèles prédictifs deux approches peuvent être considérées :

- l'approche supervisée qui consiste à utiliser des données saines (sans pannes) et des données non saines (avec pannes) afin de créer un modèle qui permettra de classifier au mieux une série temporelle dite « saine » ou « non saine ». Cette approche est aujourd'hui difficile à mettre en place car le nombre de pannes industrielles est souvent relativement faible, et donc insuffisant pour permettre ce genre d'apprentissage.
- l'approche non supervisée qui consiste à créer un modèle seulement à partir de données saines en créant un espace de normalité qui représente le bon fonctionnement de l'équipement. Lorsque de nouvelles données sortiront de cet espace de normalité, elles seront considérées comme non saines. Cette approche ne nécessite pas d'historique de pannes pour la création du modèle (bien que quelques occurrences soient nécessaires pour le valider), elle est donc plus facilement applicable au domaine de la prédiction de pannes industrielles.

Les algorithmes de machine learning permettant de créer des modèles en mode non supervisé font partie de la classe d'algorithmes de détection d'anomalies ou de détection de nouveautés. Les plus connus dans la littérature sont Isolation Forest [4] et OneClassSVM [5]. Ces méthodes s'appliquent sur des fenêtres de données de taille fixe et peuvent être utilisées directement sur les données brutes ou sur des données générées par des algorithmes de pré-traitement comme la transformée de Fourier [6] pour n'en citer qu'un. Plus récemment, des méthodes de deep learning ont vu le jour. Les plus connues utilisent des réseaux de neurones récurrents avec une architecture autoencoder [7]. Le principe est de compresser le signal dans un espace de dimension réduite puis de le reconstruire; la différence entre le signal d'entrée et le signal reconstruit est utilisée comme indicateur de normalité.

L'objectif de ce document est de comparer plusieurs algorithmes de détection d'anomalies sur un ensemble de bases de données pertinentes. La suite du document présente la méthodologie mise en place, les algorithmes étudiés et les bases de données (datasets) utilisées avant de présenter les résultats.

AMIRAL TECHNOLOGIES

Méthodologie

La comparaison d'algorithmes nécessite certes des algorithmes, mais aussi des bases de données d'entraînement et de test, et enfin une méthode de comparaison. Concernant les algorithmes, outre les algorithmes d'Amiral Technologies, un ensemble d'algorithmes connus a été sélectionné pour l'étude. L'objectif n'a pas été de paramétrer chaque algorithme sur chaque base de données afin d'obtenir les meilleurs résultats mais plutôt d'utiliser ces algorithmes avec des hyperparamètres par défaut, ce qui correspond au cas général d'un apprentissage en aveugle. Cinq bases de données publiques ont été choisies selon certains critères que nous développerons plus tard ainsi qu'une base de données privée. Les critères de comparaison utilisés sont l'aire sous la courbe (AUC) [2] et la précision moyenne (AP) [3]. Ces deux critères se basent respectivement sur les courbes ROC et Precision-Recall.

Afin de classer les algorithmes étudiés, nous avons opté pour deux critères reflétant la généricité et la performance relative.

- **Généricité** (average score) : Ce critère est calculé en moyennant les scores de l'algorithme sur l'ensemble des datasets utilisées.
- **Performance relative** (mean rank) : Ce critère est calculé en prenant le rang moyen de l'algorithme relativement aux autres algorithmes étudiés, sur l'ensemble des datasets utilisés.

Enfin, nous avons comparé le temps nécessaire pour la production du modèle sur un seul cœur (à l'exception des réseaux de neurones qui ont nécessité les ressources de cartes graphiques (GPU). Pour automatiser les travaux de comparaison, un framework de comparaison automatique d'algorithmes a été développé. Les bases de données sélectionnées présentent généralement des données issues de plusieurs capteurs. Comme certains algorithmes ne sont pas adaptés au traitement multi-variables, un modèle par variable a été construit, ensuite les prédictions ont été fusionnées à l'aide d'une logique simple équivalente à une moyenne pondérée par leur hauteur de prédiction. L'apprentissage suit un protocole de validation croisée pour éviter tout sur-apprentissage (overfitting).

Modèles / Algorithmes

Dans la suite de ce document, un modèle (Figure 1) est défini comme une séquence d'algorithmes comprenant un algorithme de génération de caractéristiques et un algorithme de détection de nouveautés/anomalies. Dans le cas des méthodes neuronales, l'étape de génération de caractéristiques n'existe pas.

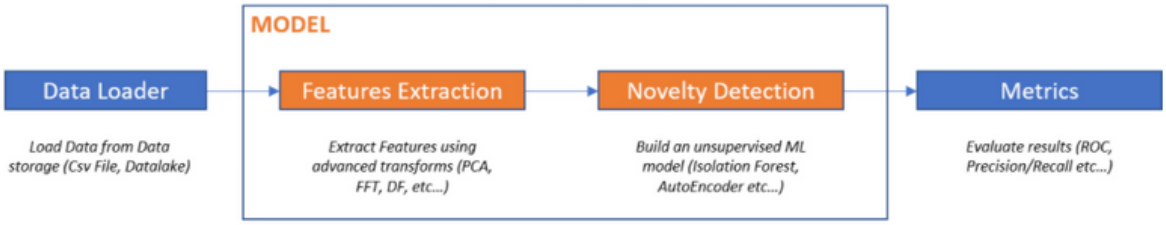

Figure 1 : Schéma d'un modèle prédictif

Dans cette étude, trois algorithmes de génération de caractéristiques et cinq algorithmes de détection d'anomalies ont été testés.

Génération de caractéristiques

- FFT [6] : La transformée de Fourier est une fonction qui transforme une série temporelle en une série d'information décrivant son spectre fréquentiel.

- Tsfresh [8] : Générateur de caractéristiques basé sur des méthodes de traitement du signal dans différents domaines (temporel, fréquentiel, spectral).

- DF : **DiagFeatures est l'ensemble de générateurs de caractéristiques propriété d'Amiral Technologies.** Il est adapté aux séries temporelles cycliques industrielles, est issu du domaine de la recherche en traitement du signal et en automatique.

Détection d'anomalies

- IF [4] : L'Isolation Forest calcule un score d'anomalie pour chaque observation de la base de données. Pour calculer ce score, l'algorithme isole la donnée de manière récursive, il choisit au hasard une variable et un seuil de coupure au hasard et évalue si cela permet d'isoler une observation en particulier.

- LOF [9] : Cet algorithme permet de trouver des anomalies en mesurant des déviations locales d'un point par rapport à ses voisins.

- OCSVM [5] : L'algorithme One Class SVM est basé sur la méthode machine à vecteurs de support mais adapté à la détection d'anomalies.

- AE [10] : Le LSTM-AutoEncoder est basé sur les réseaux de neurones. Le principe consiste à encoder la série temporelle dans un espace de dimension inférieure puis de tenter de reconstruire le signal d'entrée. L'algorithme est entraîné sur des données saines. La différence entre le signal reconstruit et le signal d'origine est vu comme la distance par rapport à l'espace de normalité. L'architecture du réseau est basée sur des cellules LSTM.

- FORE [11] : Le LSTM-Forecasting a la même approche que celle du LSTM-Autoencoder mais au lieu de reconstruire le signal d'entrée, l'algorithme essaie de prédire les données futures. L'écart observé entre la prédiction et les données réelles correspond à la distance par rapport à l'espace de normalité. L'architecture du réseau est basée sur des cellules LSTM.

- BFD : **Le Blind Fault Detector est un algorithme développé par Amiral Technologies** fonctionnant sur le principe de détection d'anomalies au même titre que l'Isolation Forest ou encore le OneClassSVM. Il est rapide en vitesse de calcul et a été conçu pour être utilisé avec DiagFeatures.

Bases de données

Pour cette étude, il a fallu sélectionner un ensemble de bases de données pertinentes ayant les critères suivants :

- Contenant des séries temporelles cycliques
- Spécifiques à la détection d'anomalies sur des équipements industriels
- Idéalement multi-capteurs
- Ayant des occurrences de pannes afin de valider le modèle
- Ayant une fréquence d'échantillonnage supérieure ou égale à 1Hz

Cinq bases de données publiques ont été sélectionnées ainsi qu'une base de données propriétaire :

- Zema [12] : Les données traitent de l'évaluation de l'état d'un banc d'essai hydraulique sur des mesures provenant de plusieurs capteurs. La base de données présente plusieurs occurrences de pannes avec différents niveaux de sévérité.

- Hai [13] : Cette base de données a été développée pour de la recherche en détection d'anomalies sur les systèmes cyber-physiques comme les chemins de fer, le traitement des eaux et les centrales électriques.

- **BearingVibration [14] :** Les données de cette base contiennent des signaux de vibration collectés à partir de systèmes de roulement de différents états de santé, et dans des conditions de vitesse de rotation variant dans le temps.

- **CentrifugalPump [15] :** Les données de vibration sont collectées sur un système d'acquisition de données de pompe centrifuge auto-amorçante. Les données sont collectées dans des conditions normales et des conditions de défaut, y compris l'usure des rouleaux de roulement, l'usure de la bague intérieure et les conditions de défaut d'usure de la bague extérieure, ainsi que l'état de défaut d'usure de la roue.

- **BatteryAging [16] :** Cet ensemble de données a été collecté à partir d'un banc de test de pronostics de batterie personnalisé au centre d'excellence de pronostics de la NASA Ames (PCoE). Les batteries Li-ion ont subi 3 profils opérationnels différents (charge, décharge et spectroscopie d'impédance électrochimique) à différentes températures. Les décharges ont été effectuées à différents niveaux de charge de courant jusqu'à ce que la tension de la batterie tombe aux seuils de tension prédéfinis. Certains de ces seuils étaient inférieurs à celui recommandé par le constructeur (2,7 V) afin d'induire des effets de vieillissement par décharge profonde. Des cycles répétés de charge et de décharge entraînent un vieillissement accéléré des batteries.

- DigitalTwin : Il s'agit d'une base de données synthétique créée par Amiral Technologies. Bien qu'elle soit synthétique, elle représente néanmoins un système réel, plus précisément un système pendulaire double, à savoir un système non linéaire à 6 états. A partir de ce « jumeau numérique », une base de données de 10 capteurs a été générée. La base contient 290 cycles de taille 200.

Pour les bases de données publiques, un formatage des données ainsi que des opérations de nettoyage (remplacement des valeurs Nan, linéarisation des timestamps, découpage en cycles, rééchantillonnage, sélection de capteurs, adaptation des labels) ont dû être appliquées pour que l'ensemble des modèles puisse fonctionner et que notre outil de comparaison puisse opérer.

La Figure 2 résume les caractéristiques des bases de données de test formatées.

Nom	Nombre de cycles sains	Nombre de cycles non sains	Nombre de capteurs	Fréquence d'échantillonnage	Taille de cycles
ZEMA	10	119	7	100 Hz	6000
HAI	5303	25	2	1 Hz	100
BEARINGVIBRATION	198	216	1	5000 Hz	5000
CENTRIFUGALPUMP	81	80	5	100 Hz	256
BATTERYAGING	438	206	6	10 Hz	1000
DIGITALTWIN	200	90	10	N/A	200

Figure 2 : Tableau récapitulatif des bases de données formatées

Résultats

Le nommage des modèles dans la suite de ce document est la concaténation de l'algorithme de détection de nouveautés avec le nom du générateur de caractéristiques séparés par le caractère "_". Par exemple, pour un modèle utilisant Isolation Forest (IF) et DiagFeatures (DF), son nom est « IF_DF ». S'il n'y a pas de générateur de caractéristiques, le terme None est utilisé, hormis pour les réseaux de neurones.

La Figure 3 présente un tableau récapitulatif des aires sous la courbe (AUC). Pour rappel, les critères de classement utilisés sont :

- Average Score : Score moyen obtenu sur l'ensemble des bases de données. Ce critère est une mesure de la généricité des modèles.
- Mean rank : Rang moyen du modèle sur l'ensemble des bases de données. Ce critère est une mesure de la performance relative des modèles.
- Time(s) : Temps cumulé d'apprentissage et de prédiction en seconde sur un seul cœur. Seules les méthodes basées sur les réseaux de neurones utilisent les ressources GPU.

	ZEMA	BearingVibration	HAI	CentrifugalPump	DigitalTwin	BatteryAging	Average Score	Mean Rank	Time (s)
BFD_DF	0.881033	0.995669	1.000000	0.998211	0.968254	0.877758	0.953487	4.083333	87.392006
LOF_DF	0.841190	0.999104	0.996781	0.984797	0.755767	0.928937	0.917763	7.666667	88.944392
BFD_None	0.937149	0.855884	0.891344	0.996752	0.991323	0.790977	0.910571	8.333333	13.807104
OCSVM_DF	0.824916	1.000000	0.999790	0.999574	0.738677	0.815495	0.896409	5.916667	74.165996
IF_None	0.959596	0.789576	0.722406	0.997557	0.998095	0.836769	0.884000	7.833333	39.481851
OCSVM_Tsfresh	0.805836	1.000000	0.999824	0.998345	0.681164	0.775511	0.876780	7.416667	24218.436123
IF_Tsfresh	0.590348	0.989546	0.869537	0.990727	0.969418	0.838949	0.874754	9.166667	26614.977597
OCSVM_None	0.926487	0.910096	1.000000	0.998911	0.560317	0.834653	0.871744	6.416667	7.974381
IF_DF	0.662177	0.994698	0.768736	0.993675	0.979471	0.793858	0.865436	8.666667	103.598173
BFD_Tsfresh	0.597082	0.989098	0.998474	0.998363	0.747937	0.805319	0.856045	8.166667	26048.705819
LOF_Tsfresh	0.656566	0.936231	0.998038	0.981921	0.895873	0.659883	0.854752	10.833333	25599.985191
FORE	0.659371	0.872312	0.999891	0.883511	0.975026	0.709556	0.849945	10.000000	1056.018176
AE	0.944444	0.895759	0.991197	0.992311	0.821481	0.306461	0.825276	9.833333	1849.135605
LOF_FFT	0.554433	0.775314	0.932257	0.977148	0.921958	0.650278	0.801898	14.166667	19.529669
OCSVM_FFT	0.543771	0.874627	0.997233	0.999899	0.611323	0.779295	0.801025	10.166667	28.978763
LOF_None	0.864759	0.848865	0.988556	0.988509	0.461693	0.621294	0.795613	12.666667	16.724266
IF_FFT	0.448934	0.787186	0.954039	0.987464	0.936508	0.244445	0.726429	14.166667	38.111975
BFD_FFT	0.355219	0.793832	0.894856	0.987581	0.530794	0.413764	0.662674	15.500000	67.101667

Figure 3 : Tableau récapitulatif des résultats basés sur AUC

La Figure 4 présente le même tableau que la Figure 3 mais cette fois-ci basé sur le critère de précision moyenne (AP).

	ZEMA	BearingVibration	HAI	CentrifugalPump	DigitalTwin	BatteryAging	Average Score	Mean Rank	Time (s)
BFD_DF	0.792909	0.998572	1.000000	0.999398	0.976737	0.756046	0.920610	3.750000	87.392006
OCSVM_DF	0.736075	1.000000	0.925926	0.999852	0.743661	0.627378	0.838815	5.416667	74.165996
OCSVM_None	0.871220	0.976937	1.000000	0.999638	0.499089	0.658957	0.834307	6.083333	7.974381
FORE	0.471509	0.965421	0.980036	0.947441	0.979015	0.555920	0.816557	10.000000	1056.018176
OCSVM_Tsfresh	0.711766	1.000000	0.937778	0.999443	0.644025	0.528742	0.803626	7.250000	24218.436123
LOF_Tsfresh	0.589459	0.931487	0.590439	0.993128	0.903417	0.559739	0.761278	12.000000	25599.985191
LOF_DF	0.683307	0.999745	0.478295	0.994815	0.613524	0.791022	0.760118	8.500000	88.944392
IF_None	0.911428	0.928551	0.018725	0.999162	0.998075	0.689193	0.757522	8.000000	39.481851
BFD_Tsfresh	0.439287	0.996209	0.727323	0.999450	0.704581	0.644903	0.751959	8.166667	26048.705819
BFD_None	0.833622	0.949929	0.111158	0.998891	0.993268	0.600963	0.747972	7.833333	13.807104
OCSVM_FFT	0.423945	0.966262	0.717141	0.999966	0.582139	0.598535	0.714665	9.666667	28.970763
IF_DF	0.513834	0.998347	0.049807	0.997892	0.985131	0.624633	0.694941	8.666667	103.598173
AE	0.787161	0.969058	0.214047	0.997647	0.871615	0.271312	0.685140	10.500000	1849.135605
IF_Tsfresh	0.463724	0.996985	0.033348	0.996547	0.952959	0.661861	0.684237	9.500000	26614.977597
LOF_FFT	0.403434	0.935494	0.086716	0.992186	0.926153	0.430430	0.629069	14.333333	19.529669
LOF_None	0.679502	0.949138	0.252884	0.995655	0.433708	0.460374	0.628544	13.166667	16.724266
BFD_FFT	0.297965	0.943292	0.592499	0.995855	0.535633	0.288794	0.609006	14.000000	67.101667
IF_FFT	0.335304	0.941683	0.094885	0.995800	0.925588	0.239865	0.588854	14.166667	38.111975

Figure 4 : Tableau récapitulatif des résultats basés sur AP

D'après les 2 tableaux ci-dessus et quelles que soient les méthodes de classement utilisées, les conclusions sont les suivantes :

- La générateur de caractéristiques d'Amiral Technologies en amont d'un algorithme de détection de nouveautés permet d'améliorer les performances du modèle.
- Les réseaux de neurones ont des résultats honorables sur certaines bases mais de très mauvais sur d'autres. Ces méthodes semblent moins génériques en appliquant un paramétrage commun à l'ensemble des bases de données.
- Il n'y a pas de modèle qui surpasse les autres, mettant ainsi en avant la difficulté de déployer un modèle unique.

En combinant l'ensemble des métriques, le modèle BFD_DF ressort comme étant le meilleur modèle sur ces datasets de comparaison. Dans la majorité des applications, la prédiction de pannes doit se faire en temps réel sur plusieurs équipements en même temps. Le temps de calcul est donc un critère qui n'est pas à négliger.

La Figure 5 montre le compromis Performance (AUC)-Vitesse sur un graphique 2D. Sur ce critère, le modèle BFD_DF d'Amiral Technologies se démarque des autres modèles.

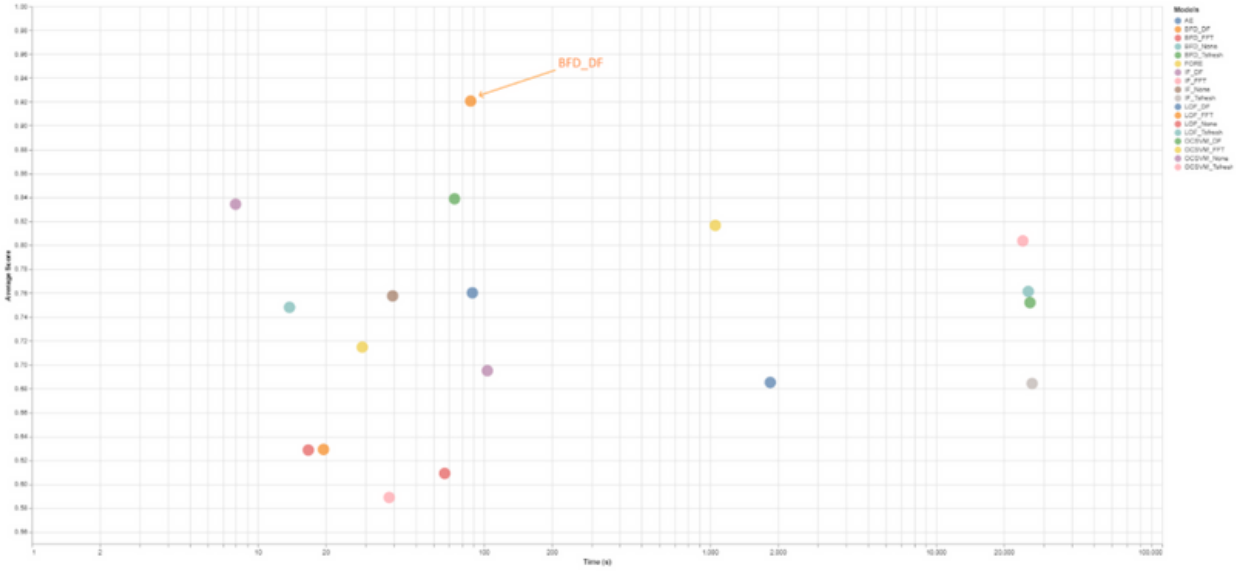

Figure 5 : Compromis Vitesse-Performance

Discussion

Le document présente une première comparaison d'algorithmes de détection de nouveautés sur des données de séries temporelles cycliques. Les métriques qui se basent sur les courbes ROC et Precision-Recall sont des métriques sujettes à discussion. Premièrement, un modèle qui permet de prédire une panne bien avant que la vérité terrain l'ait labellisée en tant que telle est pénalisé. Deuxièmement, le calcul de la courbe se base sur les prédictions des algorithmes de manière individuelle sans prendre en compte la notion temporelle des anomalies. Il y a donc un travail sur la modification du calcul des courbes qui mériterait d'être entrepris. D'autre part, le modèle qui est déployé en production pour prédire si un cycle est sain ou non doit forcément avoir un seuil de décision. Ainsi le modèle ne correspond qu'à un point de la courbe ROC (ou Precision-Recall). Une mauvaise sélection de ce seuil peut avoir de grandes conséquences sur la performance du modèle.
Pour mieux illustrer le problème, un calcul de seuil automatique se basant sur le taux de fausse acceptation désiré (fixé à 1% dans cette étude) a été appliqué sur l'ensemble des modèles.

La Figure 6 montre les résultats de comparaison de ces modèles en se basant sur le F1-Score [3].

	ZEMA	BearingVibration	HAI	CentrifugalPump	DigitalTwin	BatteryAging	Average Score	Mean Rank	Time (s)
BFD_DF	0.771429	1.000000	0.991332	0.993423	0.700000	0.784067	0.873375	4.750000	87.392006
OCSVM_DF	0.792793	0.975207	0.983271	0.887895	0.695652	0.781316	0.852556	8.666667	74.165996
IF_DF	0.771429	0.925373	0.851549	0.978286	0.700000	0.784067	0.835117	7.583333	103.598173
LOF_DF	0.771429	0.851852	0.809482	0.990359	0.700000	0.786540	0.818277	6.583333	88.944392
BFD_FFT	0.765957	0.540881	0.996115	0.920979	0.700000	0.764643	0.781429	8.000000	67.101667
AE	0.650000	0.571429	0.988016	0.920913	0.700000	0.781415	0.768629	9.083333	1849.135605
IF_Tsfresh	0.765957	0.202899	0.805305	0.970736	0.700000	0.784067	0.704827	9.916667	26614.977597
LOF_FFT	0.727273	0.121212	0.880863	0.979783	0.700000	0.786540	0.699279	9.166667	19.529669
LOF_Tsfresh	0.765957	0.092308	0.933855	0.919053	0.700000	0.782700	0.698979	10.000000	25599.985191
IF_FFT	0.757143	0.342342	0.961869	0.726175	0.700000	0.651163	0.689782	11.250000	38.111975
LOF_None	0.739130	0.031746	0.916913	0.983393	0.700000	0.785263	0.679408	10.750000	16.724266
BFD_Tsfresh	0.765957	0.389610	0.919855	0.308066	0.700000	0.786540	0.645005	9.416667	26048.705819
FORE	0.739130	0.171429	0.987049	0.555032	0.700000	0.697059	0.641616	11.666667	1056.018176
BFD_None	0.771429	0.031250	0.854504	0.469114	0.700000	0.785263	0.601927	10.750000	13.807104
OCSVM_None	0.888889	0.000000	0.922903	0.212984	0.700000	0.777542	0.583720	11.500000	7.974381
OCSVM_FFT	0.000000	0.390805	0.990692	0.611504	0.702509	0.786540	0.580342	7.416667	28.970763
IF_None	0.776978	0.000000	0.804652	0.354053	0.700000	0.784067	0.569959	11.916667	39.481851
OCSVM_Tsfresh	0.607595	0.062500	0.992503	0.052845	0.700000	0.776246	0.531948	12.583333	24218.436123

Figure 6 : Comparaison des F1-Score

En prenant comme référence le modèle OCSVM_None, bien qu'il ait une courbe ROC honorable sur la base de données BearingVibration, le F1-Score est de 0.0. Ce modèle n'est donc pas du tout opérationnel. Un bon modèle est donc un modèle ayant une bonne courbe avec un bon seuil de décision. Ici les modèles utilisant DiagFeatures sont les plus robustes.

L'amélioration de la méthode de calcul des performances d'un modèle ainsi que la recherche et le développement d'un meilleur algorithme de détection automatique de seuil de décision feront l'objet d'une prochaine publication scientifique.

Enfin, la comparaison n'inclut pas la mise en place d'un processus d'Auto-ML [17] sur les algorithmes, ce qui serait intéressant à tester pour aller au bout du travail de comparaison.

Conclusion

Cet article est une première version qui permet d'avoir une rapide vue d'ensemble du comportement des algorithmes de prédiction de pannes en mode non supervisé. Il compare un ensemble d'algorithmes provenant à la fois du traitement du signal, de l'automatique, du machine learning et du deep learning. La comparaison a été faite sur un ensemble de bases de données lié au domaine de la prédiction de pannes industrielles. Il en ressort que le modèle BFD_DF d'Amiral Technologies présente le meilleur compromis généricité-performance-vitesse sur l'ensemble des bases de données de test. Dans le cas qui nous intéresse, à savoir, la prédiction de pannes en mode aveugle, ces critères sont déterminants. De plus, lorsqu'on aborde l'implémentation réelle avec définition du seuil de décision, le générateur DiagFeatures ressort comme étant celui qui donne les meilleurs résultats.

Références

[1] V.Heurtin, "Data Cyclicity," January 2021. [Online]. Available: https://www.amiraltechnologies.com/actualite/2020/12/data-cyclicity/.
[2] A. Géron, Machine Learning avec Scikit-Learn, mise en oeuvre et cas concret", 2nd édition, p. 90.
[3] A. Géron, Machine Learning avec Scikit-Learn, mise en oeuvre et cas concret", 2nd édition, p. 94.
[4] F. T. Liu, K. M. Ting and Z.-H. Zhou, "Isolation Forest," 2008 Eighth IEEE International Conference on Data Mining: 413–422. doi:10.1109/ICDM.2008.17. ISBN 978-0-7695-3502-9. S2CID 6505449, 2008.
[5] S. S. Khan and M. G. Madden, "A Survey of Recent Trends in One Class Classification," Artificial Intelligence and Cognitive Science. Lecture Notes in Computer Science. Springer Berlin Heidelberg. 6206: 188–197. doi:10.1007/978-3-642-17080-5_21. hdl:10379/1472. ISBN 9783642170805, 2010.
[6] J. W. a. J. W. T. Cooley, "An algorithm for the machine calculation of complex Fourier series," Math. Comput. 19: 297-301, 1965.
[7] Wikipedia contributors, "Autoencoder," [Online]. Available: https://en.wikipedia.org/w/index.php?title=Autoencoder&oldid=1008370584.
[8] M. K.-L. A. a. F. M. Christ, "Distributed and parallel time series feature extraction for industrial big data applications," 2016.
[9] M. M. Breunig, H.-P. Kriegel, R. T. Ng and J. Sander, "LOF: Identifying Density-based Local Outliers," Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data. SIGMOD. pp. 93–104. doi:10.1145/335191.335388. ISBN 1-58113-217-4, 2000.
[10] Machine Learning Mastery, "A Gentle Introduction to LSTM Autoencoders," [Online]. Available: https://machinelearningmastery.com/lstm-autoencoders/.
[11] Publici sapiens, "Long Short-Term Memory (LSTM) Networks for Time Series Forecasting," October 2020. [Online]. Available: https://blog.engineering.publicissapient.fr/2020/09/23/long-short-term-memory-lstm-networks-for-time-series-forecasting/ .
[12] E. P. A. S. Nikolai Helwig, "Condition Monitoring of a Complex Hydraulic System Using Multivariate Statistics," Proc. I2MTC-2015 - 2015 IEEE International Instrumentation and Measurement Technology Conference, paper PPS1-39, Pisa, Italy, May 11-14, 2015, doi: 10.1109/I2MTC.2015.7151267.
[13] W. L. J.-H. Y. a. H. K. Hyeok-Ki Shin, "HAI 1.0: HIL-based Augmented ICS Security Dataset," 13th USENIX Workshop on Cyber Security Experimentation and Test (CSET 20), Santa Clara, CA, 2020.
[14] N. B. M. L. H. Huang, "Bearing fault diagnosis under unknown time-varying rotational speed conditions via multiple time-frequency curve extraction," J. Sound Vib., 414 (2018).
[15] Y. W. R. C. Chen Lu, "Fault Diagnosis for Rotating Machinery: A Method based on Image Processing," 2016.
[16] DAWN MCINTOSH, "Li-ion Battery Aging Datasets," [Online]. Available: https://c3.nasa.gov/dashlink/resources/133/.
[17] Wikipedia contributors, "Automated machine learning," [Online]. Available: https://en.wikipedia.org/w/index.php?title=Automated_machine_learning&oldid=1002595487.

AMIRAL TECHNOLOGIES

AMIRAL TECHNOLOGIES

CONTACTEZ-NOUS

contact@amiraltechnologies.com

Amiral Technologies
31 rue Gustave Eiffel
38000 Grenoble
France