

Vers un outil d'évaluation comparative pour la maintenance prédictive : comment comparer différentes approches ?

Antoine Guillaume*, Christel Vrain*, Elloumi Wael**

*LIFO, Université d'Orléans, 6 Rue Léonard de Vinci, 45067 Orléans

christel.vrain@univ-orleans.fr

antoine.guillaume@ensta-paris.fr

**Worldline, 19 Rue de la Vallée Maillard, 41000 Blois

wael.elloumi@worldline.com

Résumé. L'industrie 4.0 et la maintenance prédictive sont des domaines très actuels, les applications industrielles sont aujourd'hui nombreuses, ainsi que les articles présentant des modèles prédictifs pour résoudre des cas d'applications. Il y a cependant dans la littérature un manque d'articles traitant des problématiques liées à l'évaluation et la comparaison de ces modèles. Cela peut conduire à l'utilisation de protocoles expérimentaux et de métriques non adaptés à l'application. Également, la comparaison entre différentes familles de modèle prédictif (p. ex. régression et classification) n'est pas systématique, car les métriques utilisées sont différentes. Dans cet article de positionnement, nous souhaitons attirer l'attention de la communauté sur les problématiques liées à l'évaluation des modèles de maintenance prédictive. Dans un premier temps, nous introduisons le domaine de la maintenance prédictive ainsi qu'une formulation communément utilisée pour la tâche d'apprentissage. Ensuite, nous montrons les failles des protocoles expérimentaux classiques, et proposons une alternative permettant d'évaluer et de comparer des modèles en fonction du processus de maintenance associé à l'application.

1 Introduction

L'objectif de la maintenance prédictive est d'optimiser les coûts de production. Cela passe par la réduction de certains coûts liés aux opérations de maintenance et par la maximisation du temps pendant lequel les machines sont productives. En effet, le système ayant la capacité de planifier des maintenances à l'avance, les opérations peuvent être effectuées au moment où le coût d'arrêt de la machine est minimum (p. ex. lorsqu'elle n'est pas utilisée).

Cependant, la politique de maintenance peut être différente en fonction des cas d'utilisation. Par exemple, dans le cas d'un système impactant la sécurité des personnes, on jugera qu'il vaut mieux effectuer une maintenance avant tout risque d'incident, au détriment de la productivité de la machine. Chaque application définit donc sa propre politique de maintenance, située entre une fiabilité maximale et une réduction des coûts maximale.

Dans une revue récente de Esteban et al. (2022), on constate une grande diversité dans le choix des modèles et des prétraitements utilisés pour résoudre des cas pratiques. De nombreux

articles présentent des modèles sur des jeux de données publics, tel que le "NASA Turbofan Jet Engine Data Set" publié par Saxena et al. (2008). Cependant, Zschech et al. (2019) montre que sur ce même jeu de données, de nombreuses modélisations du problème sont possibles. Cela rend la comparaison entre les approches difficiles, et à notre connaissance, aucun outil n'a pour le moment été proposé pour comparer ces modèles de manière équitable.

Dans le domaine de l'ingénierie de fiabilité (O'Connor et Kleyner, 2012), la performance d'un système de maintenance est définie par une notion de coût. Celui-ci dépend des machines étudiées, du processus de maintenance et du moment où la maintenance est effectuée. C'est là qu'une première fracture apparaît entre le champ de l'ingénierie de fiabilité et celui de l'apprentissage automatique. Cette notion de coût d'un système de maintenance fait partie des notions de base dans le cadre de l'ingénierie de fiabilité, alors qu'elle n'apparaît que rarement dans les articles présentant des modèles d'apprentissage automatique. De plus, l'utilisation de métrique comme l'erreur moyenne absolue ou le F1-score peut mener à une mauvaise interprétation des performances d'un modèle pour les applications de maintenance prédictive.

Dans cet article, notre objectif est de proposer une méthodologie pour évaluer le coût d'un système de maintenance, indépendamment du type de modèle utilisé. Pour cela, nous définissons un protocole expérimental s'adaptant aux contraintes applicatives et aux différentes formulations du problème. Dans un premier temps, nous introduisons le domaine de la maintenance prédictive (types de données utilisées, extraction d'un jeu de données) et formalisons la tâche d'apprentissage. Ensuite, nous présentons un protocole de base, souvent utilisé dans la littérature, et montrons les problèmes qu'il peut causer. Enfin, pour répondre aux problématiques identifiées, nous définissons notre protocole et notre métrique d'évaluation, basée sur les coûts du processus de maintenance.

2 Formalisation du problème

Dans cette section, nous introduisons plus formellement l'application d'un modèle d'apprentissage à un cas de maintenance prédictive. Par souci de simplicité, nous considérons que nous disposons de données émises par une ou plusieurs machines et que notre objectif est simplement de prédire les défaillances, indépendamment des parties de la machine les provoquant.

Il existe cependant des cas où l'on souhaite identifier la partie de la machine qui va causer la panne, par exemple, pour apporter des informations complémentaires lors d'un diagnostic effectué par l'équipe de maintenance, ou simplement pour ne pas avoir à réaliser de diagnostic. La formulation du problème peut facilement s'adapter à ce cas d'usage, par exemple, en considérant un problème multiclasse plutôt que binaire (c.-à-d. émettre une alerte ou non à un temps t) pour une tâche de classification. Il est aussi possible de découper le problème en sous problèmes, avec un modèle par partie à monitorer.

Notre objectif dans les sections suivantes n'est pas de fournir une revue des modélisations possibles pour les problèmes de maintenance prédictive, déjà effectuée par Esteban et al. (2022), mais d'introduire une modélisation communément utilisée dans la littérature, afin de pouvoir définir formellement notre protocole.

2.1 Type de données pour la maintenance prédictive

On peut distinguer plusieurs cas concernant l'origine des données utilisées pour apprendre un modèle prédictif :

- Le premier cas, qui est le plus courant, est celui de données issues de capteurs qui mesurent des phénomènes physiques, tels que l'amplitude d'une vibration, comme dans Orhan et al. (2006), ou la température. Ces données sont représentées sous forme de séries temporelles, qui représentent l'évolution des valeurs capturées au fil du temps.
- Le deuxième cas est celui de données issues de journaux d'événement, qui sont produits par un système informatique. Par exemple, Wang et al. (2017) présente un cas d'utilisation de données catégorielles, avec des journaux contenant des codes d'événements caractérisant une action du système ou un problème de fonctionnement. On peut distinguer les journaux d'événements spécifiquement générés pour la maintenance prédictive des journaux d'événements destinés à la maintenance logicielle.
- Plus rarement, on trouve aussi des séries d'images, chacune pouvant permettre d'évaluer l'état d'usure d'une partie du système. Par exemple, Schlagenhauf et Burghardt (2021) propose un système basé sur l'identification des marques d'usures sur des parties d'une machine.

Un point commun existe entre toutes ces sources de données : la composante temporelle. Dans la suite, nous considérerons donc que nous disposons de séries temporelles, indépendamment de leur contenu.

2.2 Extraire un jeu de donnée de maintenance prédictive

Considérons une série temporelle $X = \{x_1, \dots, x_m\}$ (ici univariée), contenant les données générées par une machine dans un intervalle de temps $[1, m]$. À partir d'un historique des défaillances sur cette période, contenant la date de début et de fin de chaque défaillance, on souhaite extraire des sous-séquences adaptées pour apprendre un modèle de maintenance prédictive. Si on a plusieurs machines, l'ensemble des sous-séquences extraites de chaque machine formera le jeu de données. Pour définir un processus d'extraction adapté au cas d'usage, on peut poser les questions suivantes :

- Doit-on déterminer la partie de la machine responsable de la défaillance ?
- Dans le cas de séries multivariées, les variables sont-elles indépendantes les unes des autres ?
- L'historique des défaillances contient-il la cause des pannes et les opérations effectuées pour les résoudre ?

Nous définissons les sous-séquences extraites de X comme des cycles de vie. Chaque cycle de vie contient les données émises par la machine pendant un intervalle de fonctionnement normal. Cet intervalle commence après la résolution d'une panne et prend fin au moment de la prochaine panne. Un cycle de vie est donc défini par une machine, depuis laquelle il est extrait, un intervalle de temps, et si la cause de la panne est connue, une partie de la machine.

En fonction des caractéristiques des séries temporelles, le processus d'extraction des cycles de vie peut changer. Par exemple, si on dispose de capteurs qui mesurent la température de plusieurs parties d'une machine, il est possible qu'une augmentation de la température causée par une partie affectera également les autres capteurs. Dans ce cas, créer des cycles de vie indépendants pour chaque partie de la machine, qui contiendrait donc uniquement les données émises

par le capteur de la partie concernée, pourrait mener à des faux positifs dus aux changements de température causés par d'autres parties.

Plus formellement, considérons une série $X = \{x_1, \dots, x_m\}$ représentant les données émises par une machine. Supposons qu'on connaisse l'existence de deux pannes, chacune définie par une date de début et de fin (c.-à-d. le moment où la panne se produit et la fin de la maintenance). On définira ces dates par $[t_1, t_2[$ pour la première panne, et $[t_3, t_4[$ pour la seconde. À partir de ces deux pannes, on peut extraire trois cycles de vie de X . Le premier serait $X_1 = \{x_1, \dots, x_{t_1-1}\}$, le second $X_2 = \{x_{t_2}, \dots, x_{t_3-1}\}$ et le troisième $X_3 = \{x_{t_4}, \dots, x_m\}$. Dans ce cas, X_3 ne se termine pas par une panne, et peut ne pas être utilisable par certaines approches, dû à l'incertitude sur la date de la panne. Par exemple, un modèle de régression devrait connaître le temps restant avant la panne à chaque point du temps pour pouvoir utiliser X_3 durant la phase d'entraînement. Enfin, si les parties responsables des pannes sont connues, on les affectera aux cycles de vie correspondants.

À partir de maintenant, on notera $\mathcal{X} = \{X_1, \dots, X_n\}$ l'ensemble des cycles de vie extrait des données brutes émises par les machines, avec n le nombre de cycles de vie. On peut également disposer d'un vecteur $Y = \{y_1, \dots, y_n\}$ qui contient la classe (c.-à-d. la partie causant la défaillance) de chaque cycle de vie, si celle-ci est connue. Comme nous nous concentrons sur les modèles prédisant les pannes indépendamment des parties responsables, la classe y_i ne sera pas directement utilisée pour l'apprentissage d'un modèle. Il est cependant simple, par exemple, pour le cas d'un modèle de classification, de passer du cas binaire (panne/non panne) à un cas multiclasse où la cause de la panne est également prédite.

2.3 Problèmes de classification

Dans un contexte de classification, l'objectif est de déterminer si, au temps t , une alerte de maintenance devrait être lancée. Idéalement, une alerte permettra de planifier une intervention au moment où le coût de non-fonctionnement de la machine est minimum.

Pour créer un classifieur capable de prédire les pannes avec suffisamment d'avance, on peut définir des variables qui caractérisent le processus de maintenance. Bien que cette notion de variables soit présente dans la majeure partie de la littérature, il ne semble pas exister de consensus pour leur définition. Nous définissons ces variables en nous basant sur les définitions de Sipos et al. (2014), qui nous semblent les plus générales :

- **Temps de Réponse** (tr) : une durée correspondant au temps nécessaire pour effectuer la maintenance depuis le moment où une alerte est lancée. Cela comprend le temps nécessaire à l'équipe de maintenance pour arriver sur site et pour réaliser la maintenance.
- **Intervalle Prédictif** (ip) : une durée caractérisant l'avance avec laquelle il est acceptable de prédire une panne. Une alerte levée dans cet intervalle permet de réaliser une maintenance dans des conditions optimales.

L'approche la plus répandue pour intégrer ces contraintes est d'extraire des fenêtres glissantes de chaque cycle de vie, puis d'étiqueter chaque fenêtre grâce aux valeurs de tr et ip . Soit $\mathcal{W}_i = \{W_1, \dots, W_{m-(l-1)}\}$ l'ensemble des fenêtres glissantes de taille l extraites d'un cycle de vie $X_i = \{x_1, \dots, x_m\}$ qui subit une panne au temps t_{m+1} . Dans un contexte de classification binaire, où on souhaite prédire les pannes indépendamment des parties responsables, on définit la classe 1 comme les fenêtres dans lesquelles on souhaite identifier des signatures

de panne, la classe z_i d'une fenêtre $W_i = \{x_i, \dots, x_{i+(l-1)}\}$ est alors définie telle que :

$$z_i = \begin{cases} 1 & \text{si } i + (l - 1) \geq (m + 1) - (ip + tr) \\ 0 & \text{sinon} \end{cases} \quad (1)$$

La Figure 1 illustre ce processus pour un cycle de vie. Les fenêtres se terminant dans l'intervalle $[(m + 1) - (ip + tr), (m + 1) - tr]$ peuvent être utilisées pour entraîner le modèle si elles contiennent des signatures de panne qui pourraient survenir avant $(m + 1) - (ip + tr)$. Sinon, il est possible de créer une classe spécifique pour ces fenêtres, caractérisant une panne imminente.

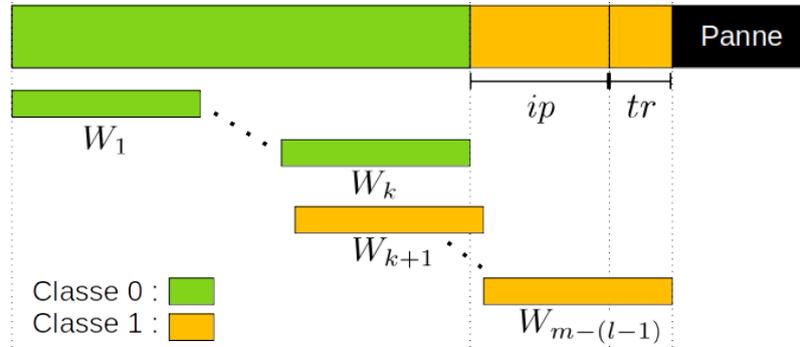


FIG. 1 – Illustration de la formulation du problème pour les tâches de classification avec une approche utilisant des fenêtres glissantes. W_i est la i^{eme} fenêtre de taille l extraite du cycle de vie, tr représente le temps nécessaire pour effectuer la maintenance, et ip l'intervalle dans lequel il est acceptable de prédire la panne.

À partir d'un ensemble de fenêtres étiquetées, on peut alors apprendre un classifieur. À ce stade, une étape d'ingénierie de caractéristiques est souvent mise en œuvre pour créer des descripteurs capturant le processus de dégradation de la machine. On appliquera ensuite le modèle de classification choisi, qui, à partir de la fenêtre de taille l la plus récente d'un nouveau cycle de vie, prédira si une alerte doit être levée.

2.4 Problèmes de régression

Dans un contexte de régression, le but est d'estimer au temps t le temps restant avant la prochaine défaillance. Cette quantité est souvent désignée par le terme "remaining useful life" (RUL). Pour lancer une alerte de maintenance, un système utilisant un modèle de régression devra définir un seuil, en dessous duquel une alerte devra être lancée. Ce seuil peut être exprimé en utilisant l'intervalle prédictif (ip) et le temps de réponse (tr) définis dans la section précédente, tel que si $RUL \leq (ip + tr)$, une alerte sera lancée.

Comme pour les modèles de classification, pour apprendre un modèle de régression à partir d'un ensemble de cycles de vie, il est commun d'extraire des fenêtres glissantes. À partir d'un cycle de vie $X_i = \{x_1, \dots, x_m\}$ et d'une taille de fenêtre l , on extrait un ensemble de fenêtres

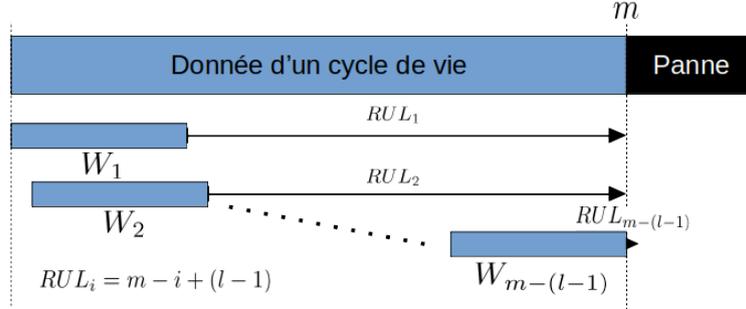


FIG. 2 – Illustration de la formulation du problème pour les tâches de régression avec une approche utilisant des fenêtres glissantes.

$\mathcal{W} = \{W_1, \dots, W_{m-(l-1)}\}$ avec $W_j = \{x_j, \dots, x_{j+(l-1)}\}$. On peut ensuite calculer le RUL de chaque fenêtre par $RUL_j = m - j + (l - 1)$. Ces opérations sont illustrées par la Figure 2.

À partir des fenêtres glissantes et de leur RUL, comme pour le cas de la classification, on peut construire un ensemble de descripteurs qui caractérisent le processus de dégradation de la machine, et utiliser un modèle de régression pour estimer les coefficients à affecter à chaque descripteur. On pourra ainsi utiliser les l derniers points de données émis par une machine pour avoir une estimation du temps restant avant la prochaine défaillance.

3 Protocoles existants

Considérons un modèle de classification ou de régression quelconque dont on souhaite évaluer la performance. Pour cela, il est commun d'utiliser un processus de validation où on divise les données en entrée en deux ensemble disjoints, l'un pour l'entraînement du modèle et l'autre pour l'évaluation de ses performances. On trouve dans la littérature l'utilisation de validation simple, où on spécifie seulement la taille des deux ensemble, comme dans Gutsch et al. (2019), mais aussi de validation croisée où cette étape est répétée pour que toutes les données soient apparues dans l'ensemble de test, comme dans le framework décrit par Vallim Filho et al. (2022).

L'évaluation des performances s'effectue ensuite avec des métriques adaptées au type modèle choisi. Par exemple, Bampoula et al. (2021) utilise le F1-score et l'exactitude pour évaluer un modèle de classification. On retrouve aussi ce type de métrique pour des modèles de régression, comme dans Gutsch et al. (2019), où à partir d'une prédiction du temps restant avant la panne, et d'un seuil de déclenchement d'une alerte, on peut calculer une mesure d'exactitude. On pourra noter que plusieurs points d'importance sont parfois occultés dans les expérimentations :

- Si une seule étape de validation est effectuée (et non une validation croisée), pourquoi ce choix ? Cela peut être justifier par des temps de calcul important, mais en l'absence d'argument, cela peut cacher un choix intentionnel des échantillons pour maximiser les métriques (c.-à-d. le phénomène de "cherry picking").

- La création des ensembles d'entraînement et de test est-elle faite sur les cycles de vie ou sur les fenêtres glissantes ? Dans ce dernier cas, garantit-on que toutes les fenêtres issues d'un cycle de vie X_i soit dans un seul des deux ensembles à chaque étape de validation ?

Dû à l'absence d'articles traitant des protocoles expérimentaux à utiliser pour les applications de maintenance prédictive, nous présentons un protocole "de base" pour l'évaluation des modèles d'apprentissage.

Ce protocole utilise une validation croisée en k étapes, utilisant une proportion de $\frac{1}{k}$ des cycles de vies pour l'ensemble d'entraînement. Le découpage en fenêtres glissantes devient alors une étape de prétraitement de la méthode d'apprentissage, afin d'éviter les fuites de données entre l'ensemble d'entraînement et de test. En effet, si on effectuait une validation croisée sur les fenêtres glissantes sans aucune contrainte, une fenêtre $W_i = \{x_i, \dots, x_{i+(l-1)}\}$ pourrait être dans l'ensemble d'entraînement, et la fenêtre W_{i+1} dans l'ensemble de test. Cela créerait une fuite de données, due aux points partagés par W_i et W_{i+1} . Pour évaluer la performance du modèle, les métriques sont calculées entre la classe prédite \hat{z}_i de chaque fenêtre W_i et sa vraie classe z_i .

Notons que lors de l'utilisation réelle du modèle appris, on traitera un flux de données, par lots ou en temps réel, qui représentera les données du cycle de vie actuel de la machine : si au temps t le modèle donne $\hat{z}_{t-(l-1)} = 1$ pour la fenêtre $W_{t-(l-1)}$, contenant les dernières données émises par la machine, un processus de maintenance sera déclenché.

3.1 Problèmes liés au protocole de base

Considérons un cycle de vie $X_i = \{x_1, \dots, x_m\}$ et les fenêtres glissantes de taille l préalablement extraites de X_i . À partir d'un modèle prédictif, on obtient un vecteur $\hat{Z}_i = \{\hat{z}_1, \dots, \hat{z}_{m-(l-1)}\}$.

Si le protocole de base défini dans la section précédente est suffisant pour évaluer la performance du modèle sur la tâche d'apprentissage, il n'est en aucun cas adapté pour estimer la performance du système de maintenance prédictive qui utilisera ce modèle. On doit en fait distinguer deux contextes d'évaluation : l'évaluation du modèle sur la tâche d'apprentissage, et l'évaluation du modèle sur l'application.

Dans le cadre des métriques sur la tâche d'apprentissage, les coûts liés au processus de maintenance ne sont pas pris en compte. De plus, il est important de définir les hypothèses suivantes :

- Un diagnostic est-il réalisé avant chaque opération de maintenance déclenchée suite à une alerte du modèle ? Ce diagnostic peut-il annuler une intervention déclenchée suite à un faux positif ?
- Quel est l'impact d'une maintenance sur les données générées par la machine ? Comment le comportement de la machine est-il affecté par ces opérations ?

En effet, les métriques ne doivent pas être les mêmes si un diagnostic peut détecter les faux positifs sans influencer sur le comportement de la machine ou si une maintenance qui modifie le comportement (p. ex. via le changement d'une pièce) est obligatoirement effectué à chaque fois qu'on a $\hat{z}_t = 1$. Notons qu'il est possible d'affecter des poids aux différents types d'erreurs, mais l'emplacement de l'erreur est rarement pris en compte.

Si une maintenance effectuée au temps t suite à une alerte modifie le comportement de la machine, il n'y a aucune garantie que, dans le cas réel, le modèle voit les données situées

Vers un outil d'évaluation comparative pour la maintenance prédictive

après t . Supposons qu'il existe un biais qui pousse le modèle à prédire $\hat{z}_1 = 1$ pour tous les cycles de vie, mais que toutes les autres prédictions sont correctes. Dans ce cas, des métriques telles que l'exactitude entre \hat{Z}_i et Z_i seraient très bonnes, alors que le modèle déclencherait systématiquement des maintenances précoces. La Figure 3 illustre cet exemple. Dans le cas où un diagnostic est effectué, si les faux positifs peuvent être détectés, il faut alors considérer le coût du diagnostic.

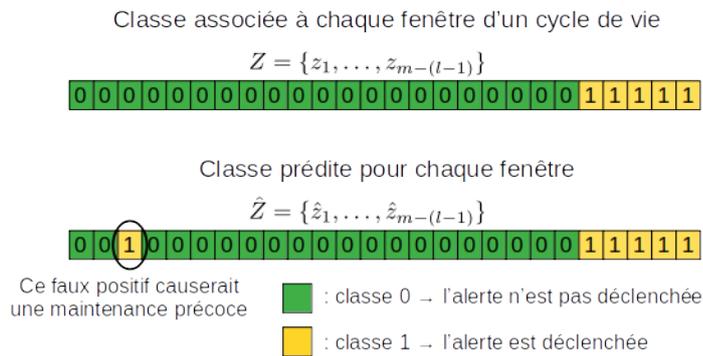


FIG. 3 – Malgré une bonne précision, si aucun diagnostic n'est effectué avant les maintenances, un faux positif précoce conduira à une maintenance inutile.

Les métriques d'évaluation de la tâche d'apprentissage (à moins d'un score parfait), en plus d'être spécifique à la formulation du problème, ne sont pas fiables pour estimer la performance du modèle sur l'application. Elles ne prennent pas en compte les particularités du processus de maintenance, ni les coûts qui y sont associés. Dans la prochaine section, nous proposons un protocole adapté aux problématiques identifiées dans cette section, ainsi qu'une métrique permettant d'estimer le coût d'un système de maintenance prédictive.

4 Protocole expérimental pour la maintenance prédictive

Dans la suite, pour définir nos solutions aux problématiques introduites dans la section précédente, nous nous plaçons dans le cadre suivant :

- On souhaite prédire l'occurrence d'une panne, sans chercher à déterminer la partie la provoquant
- Une procédure de diagnostic permet de détecter les faux positifs. On supposera que les signes de dégradation peuvent être identifiés par l'équipe technique si on a $\hat{z}_j = z_j = 1$, et qu'une maintenance sera alors effectuée.
- Le diagnostic provoque un arrêt de la machine.
- Il n'y a pas de dérive conceptuelle pour la prédiction des pannes.

Nous pensons que notre protocole reste facilement adaptable à des hypothèses différentes. Au fur et à mesure de nos définitions, nous fournirons des exemples de modifications possibles pour s'adapter à d'autres hypothèses.

Dans le cadre où on souhaite prédire les pannes indépendamment des parties les provoquant, on a en entrée du protocole un ensemble de cycles de vie $\mathcal{X} = \{X_1, \dots, X_n\}$ avec $X_i = \{x_1, \dots, x_m\}$ un cycle de vie de taille m . Si les parties fautives devaient être identifiées, on aurait également un vecteur de classe $Y = \{y_1, \dots, y_n\}$, contenant la partie défaillante à la fin de chaque cycle. Pour un modèle de classification, les fenêtres glissantes extraites de X_i qui avait, dans le cas binaire, la classe 1, prendrait alors la classe y_i .

Puisqu'on considère qu'il n'y a pas de dérive conceptuelle, on peut conserver l'utilisation d'une validation croisée, où \mathcal{X} est divisé en deux ensembles disjoints, \mathcal{X}_{Train} et \mathcal{X}_{Test} à chaque étape de validation. Si une dérive conceptuelle était présente, il faudrait utiliser une validation croisée temporelle, définie par Bergmeir et Benítez (2012), pour tenir compte de l'adaptation du modèle au changement de concepts.

Enfin, pour définir des métriques propres à l'évaluation de l'application, puisqu'on suppose qu'un diagnostic est effectué avant chaque maintenance, on peut prendre en compte tous les faux positifs émis par le modèle lors du calcul, jusqu'à la première occurrence de $\hat{z}_j = z_j = 1$. On rappellera z_j correspond à la vraie classe, et \hat{z}_j à la classe prédite par le modèle. Si une maintenance était effectuée sans vérification préalable, seulement la première alerte ($\hat{z}_j = 1$) de chaque cycle devrait être considéré. En effet, si la maintenance modifie le comportement de la machine, les données situées après j n'ont aucune garantie d'être vues par le modèle lors de l'application sur le cas réel.

4.1 Estimer le coût d'un système de maintenance prédictive

Sachant les contraintes imposées par nos hypothèses sur le processus de maintenance, on souhaite définir une métrique estimant la performance d'un modèle sur l'application. Pour cela, nous proposons une métrique prenant en compte les différents coûts liés au processus de maintenance. Elle permet d'estimer le coût associé aux opérations de maintenance d'un cycle de vie, et, par extension, d'estimer la rentabilité du système de maintenance. Dans de nombreux ouvrages traitant de l'ingénierie de fiabilité, on trouve des listes exhaustives des coûts liés au processus de maintenance (voir Arquès, 2009, Chapitre 1 Section 4) :

- Le coût lié au non-fonctionnement de la machine.
- Le coût de formation du personnel de maintenance
- Le coût du diagnostic, incluant les frais de déplacement de l'équipe technique.
- Le coût des pièces à remplacer lorsqu'elles sont disponibles.
- Si une pièce n'est pas disponible, il faut majorer son coût par celui de l'immobilisation de la machine, car elle n'est plus productive.
- Le coût de la gestion des stocks et de leur immobilisation.
- Les coûts énergétiques liés à la maintenance et au fonctionnement de la machine en fonction de son état (p. ex. un problème d'injection ou d'allumage sur un moteur entraîne une surconsommation).

Nous pouvons nous aider de cette liste pour définir notre estimation du coût d'un cycle de vie, en modélisant ces coûts par les variables suivantes :

- t_m : le moment où le cycle de vie se termine.
- t_f : le moment où une alerte est levée dans le cycle de vie.
- $C_{fail}(t)$: le coût associé au non-fonctionnement de la machine et à son immobilisation. Ce coût est fonction du temps pour tenir compte des périodes où la machine devrait être

Vers un outil d'évaluation comparative pour la maintenance prédictive

productive, et des périodes où elle n'est pas utilisée (p. ex. avec un planning hebdomadaire).

- C_{tech} : le coût de déplacement de l'équipe technique et du diagnostic de la machine. On suppose ce coût constant.
- $C_{replace}(t_m - t_f)$: le coût lié à la pièce à remplacer, ainsi qu'à son remplacement précoce. Nous supposons que le coût des pièces est fixe, et que le manque à gagner pour un remplacement précoce peut être calculé en multipliant la différence $t_m - t_f$ par une constante.
- Δ_{alert} : le temps écoulé entre une panne, son observation et le départ de l'équipe technique.
- Δ_{tech} : le temps écoulé entre le départ de l'équipe technique, l'arrivée sur site et la réalisation du diagnostic.
- Δ_{repair} : le temps nécessaire à la réalisation de la maintenance.

Par simplicité, nous considérons que l'équipe de maintenance est toujours disponible et que les pièces nécessaires à une maintenance sont toujours en stock. Ainsi, le coût d'une panne non prédite, noté C_{neg} , est égale à :

$$C_{neg} = C_{tech} + C_{replace}(0) + \int_{t_m}^{t_m + \Delta_{alert} + \Delta_{tech} + \Delta_{repair}} C_{fail}(dt) \quad (2)$$

On peut ensuite distinguer deux coûts pour les opérations de maintenance prédictive. Le premier coût est celui d'une alerte justifiée, qui sera confirmée par le diagnostic de l'équipe technique et donnera lieu à une maintenance, noté C_{pos} . Le second coût, est celui d'une alerte non justifiée, qui ne sera pas confirmée par le diagnostic, noté C_{fpos} . Ces deux coûts sont définis tels que :

$$C_{pos} = C_{tech} + C_{replace}(t_m - t_f) + \int_{t_f}^{t_f + \Delta_{tech} + \Delta_{repair}} C_{fail}(dt) \quad (3)$$

$$C_{fpos} = C_{tech} + \int_{t_f}^{t_f + \Delta_{tech}} C_{fail}(dt) \quad (4)$$

L'avantage du système de maintenance prédictive est de pouvoir planifier des interventions pendant des périodes où $C_{fail}(t)$ est minimum, afin de minimiser les pertes de production. Considérons un cycle de vie $X_i = \{x_1, \dots, x_m\}$ et les fenêtres glissantes de taille l préalablement extraites de X_i . À partir d'un modèle prédictif, on obtient un vecteur $\hat{Z}_i = \{\hat{z}_1, \dots, \hat{z}_{m-(l-1)}\}$, contenant les prédictions pour chaque fenêtre. Le coût C_{X_i} de ce cycle de vie est alors défini tel que :

$$C_{X_i} = \sum_{j=1}^{m-(l-1)} \begin{cases} C_{pos} & \text{si } \hat{z}_j = z_j = 1, \text{ ensuite on arrête la somme} \\ C_{fpos} & \text{si } \hat{z}_j = 1 \text{ et } z_j = 0 \end{cases} \quad (5)$$

Si aucune maintenance n'a été effectuée pour le cycle X_i (c.-à-d. aucune occurrence de $\hat{z}_j = z_j = 1$), on affectera alors $C_{X_i} = C_{X_i} + C_{neg}$. Le coût global du système de maintenance

prédictive peut alors être comparé à système de maintenance réactive, qui intervient quand les pannes sont constatées :

$$C_{global} = \sum_{i=1}^n (C_{neg_i} - C_{X_i}) \quad (6)$$

Une valeur positive de C_{global} indiquera que le système se basant sur le modèle prédictif en entrée de notre protocole est rentable par rapport au système contre lequel il est comparé. À l'inverse, une valeur négative indiquera que le modèle ne permet pas de réaliser des économies. On peut aussi remplacer C_{neg_i} par le résultat obtenu avec un autre modèle, pour comparer différents modèles prédictifs.

Dans le cadre où ces coûts ne sont ni connus ni estimables, on peut remplacer chaque partie des équations de C_{neg} , C_{pos} , C_{fpos} par des constantes. Une stratégie possible pour fixer les valeurs de ces constantes est de considérer des coûts relatifs. Par exemple, considérons λ_{neg} , la constante remplaçant l'intégrale de C_{neg} . Une valeur $\lambda_{neg} = 1$ donne alors le coût lié au non-fonctionnement de la machine pour une panne non prévue. Si on suppose que le coût de non-fonctionnement peut être réduit de moitié pour une maintenance prévue, on fixera alors $\lambda_{pos} = 0.5$ pour remplacer l'intégrale de C_{pos} . En fonction de la complexité du processus de diagnostic, on pourra par exemple fixer $\lambda_{fpos} = 0.1$ pour remplacer l'intégrale de C_{fpos} . On pourra fixer de la même manière les valeurs de C_{tech} et de la constante à multiplier par $t_m - t_f$ dans $C_{replace}$.

Si les coûts ne sont pas connus, il faudra veiller à présenter les résultats avec plusieurs configurations pour les valeurs des constantes de remplacement. L'objectif n'étant évidemment pas de trouver la configuration qui maximise C_{global} , mais de comprendre dans quelles configurations le modèle pourrait être bénéfique.

5 Conclusion et perspectives

Dans cet article, nous avons introduit le domaine de la maintenance prédictive et ainsi qu'une approche pour construire un jeu de donnée de maintenance prédictive à partir des données émises par une machine. Nous avons ensuite présenté une formulation de la tâche d'apprentissage pour les problèmes de régression et de classification. Après un rapide constat des protocoles expérimentaux utilisés dans la littérature et des problèmes qu'ils peuvent créer, nous avons défini un protocole expérimental adaptable aux contraintes imposées par le processus de maintenance. Enfin, nous avons introduit une métrique, permettant d'estimer le coût d'un système de maintenance prédictive.

Pour définir notre protocole, nous avons assumé un bon nombre d'hypothèses sur la formulation du problème, le traitement des données brutes et le type de modèles utilisé. Notre premier objectif sera d'étudier la capacité de notre protocole à s'adapter aux autres formulations du problème, et aux autres méthodes de prétraitements des données. L'idée étant que, indépendamment de l'approche, on peut calculer la métrique de coût en fonction des alertes émises par un modèle. Il faudra ensuite vérifier empiriquement l'intérêt de ce protocole sur des cas concrets.

À plus long terme, notre objectif est de pouvoir comparer équitablement une grande variété d'approches. Nous souhaitons mettre à disposition de la communauté un outil permettant

Vers un outil d'évaluation comparative pour la maintenance prédictive

de comparer équitablement un modèle aux autres approches existantes, notamment en proposant une interface pour évaluer un modèle sur des jeux de données publics de maintenance prédictive.

Références

- Arquès, P. (2009). *Diagnostic prédictif et défaillances des machines : théorie, traitement, analyse, reconnaissance, prédiction*. Editions Technip.
- Bampoula, X., G. Siaterlis, N. Nikolakis, et K. Alexopoulos (2021). A deep learning model for predictive maintenance in cyber-physical production systems using lstm autoencoders. *Sensors* 21(3).
- Bergmeir, C. et J. M. Benítez (2012). On the use of cross-validation for time series predictor evaluation. *Information Sciences* 191, 192–213.
- Esteban, A., A. Zafra, et S. Ventura (2022). Data mining in predictive maintenance systems : A taxonomy and systematic review. *WIREs Data Mining and Knowledge Discovery* 12(5), e1471.
- Gutsch, C., N. Furian, J. Suschnigg, D. Neubacher, et S. Voessner (2019). Log-based predictive maintenance in discrete parts manufacturing. *Procedia CIRP* 79, 528–533. 12th CIRP Conference on Intelligent Computation in Manufacturing Engineering, 18-20 July 2018, Gulf of Naples, Italy.
- O'Connor, P. et A. Kleyner (2012). *Practical Reliability Engineering, Fifth Edition*.
- Orhan, S., N. Aktürk, et V. Çelik (2006). Vibration monitoring for defect diagnosis of rolling element bearings as a predictive maintenance tool : Comprehensive case studies. *NDT & E International* 39(4), 293 – 298.
- Saxena, A., K. Goebel, D. Simon, et N. Eklund (2008). Damage propagation modeling for aircraft engine run-to-failure simulation. *International Conference on Prognostics and Health Management*.
- Schlagenhauf, T. et N. Burghardt (2021). Intelligent vision based wear forecasting on surfaces of machine tool elements. *SN Applied Sciences* 3(12), 1–13.
- Sipos, R., D. Fradkin, F. Moerchen, et Z. Wang (2014). Log-based predictive maintenance. pp. 1867–1876. Association for Computing Machinery.
- Vallim Filho, A. R. d. A., D. Farina Moraes, M. V. Bhering de Aguiar Vallim, L. Santos da Silva, et L. A. da Silva (2022). A machine learning modeling framework for predictive maintenance based on equipment load cycle : An application in a real world case. *Energies* 15(10).
- Wang, J., C. Li, S. Han, S. Sarkar, et X. Zhou (2017). Predictive maintenance based on event-log analysis : A case study. *IBM J. Res. Dev.* 61.
- Zschech, P., J. Bernien, et K. Heinrich (2019). Towards a taxonomic benchmarking framework for predictive maintenance : The case of nasa's turbofan degradation.

Summary

Industry 4.0 and predictive maintenance are hot topics, industrial applications are numerous, as well as articles presenting predictive models to solve specific use-cases. However, there is a lack of work dealing with issues related to the evaluation and comparison of these models. This can lead to the use of experimental protocols and metrics, which are not suitable for predictive maintenance. Also, the comparison between different types of predictive models (e.g. regression and classification) is not systematic, as the metrics used are different. In this position paper, we wish to draw the attention of the community to the problematics behind the evaluation of predictive maintenance models. First, we introduce the field of predictive maintenance as well as a commonly used formulation for the learning task. Then, we show the flaws of classical experimental protocols, and propose an alternative for evaluating and comparing models according to the maintenance process associated with the application.