

E. N. S. T

المدرسة الوطنية العليا للتكنولوجيا
Ecole Nationale Supérieure de Technologie
The National Higher School of Technology



الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire

وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
المدرسة الوطنية العليا للتكنولوجيا

Ecole Nationale Supérieure de Technologie
Department : Génie Industriel et Maintenance

Mémoire de projet de fin d'études
pour l'obtention du diplôme d'ingénieur d'État en Génie Industriel
Spécialité
Génie Industriel

Application du Machine Learning à la Maintenance Prédicative : Un Processus basé sur la Sélection d'Algorithmes et l'Alignement Stratégique

Réalisé par

AMROUCHE Ikram

LAHOUSSA Malak Sakina

Présenté et soutenu publiquement le (03/07/2023)

Composition du Jury :

GOUCEM Med ALI TAHAR	Président
BELAYADI Djahida	Promoteur
REZGUI WAIL	Examineur

Année universitaire 2022/2023

Alger, le 03/07/2023 .

Remerciements

Tout d'abord, nous tenons à exprimer notre gratitude envers Allah, le Tout-Puissant, qui nous a accordé le courage et la patience nécessaires pour mener ce travail à son terme.

Nous tenons à exprimer notre profonde gratitude à toutes les personnes qui ont contribué de près ou de loin à la réalisation de ce projet. Tout d'abord, nous remercions nos chers parents et toute notre famille pour leur soutien moral, intellectuel et technique inestimable tout au long de cette aventure. Leur sacrifice et leur clairvoyance ont été une source d'inspiration constante.

Nous souhaitons également adresser nos remerciements à notre directrice de recherche, Mme BELAYADI Djahida, pour son encadrement attentif et son soutien indéfectible. Ses conseils avisés, sa disponibilité et sa patience ont joué un rôle essentiel dans la réalisation de ce projet.

Nous exprimons notre gratitude envers les membres du jury de notre mémoire, pour l'intérêt qu'ils ont porté à notre travail et leur contribution à l'amélioration de sa qualité.

Nous souhaitons exprimer nos remerciements à l'entreprise Condor pour sa précieuse contribution à notre intégration dans le monde professionnel. Nous tenons également à exprimer notre sincère gratitude envers tous les membres du département GIM de l'École Nationale Supérieure de Technologie (ENST), ainsi qu'à nos amis et collègues, en particulier les étudiants GI3, pour leur apport significatif tout au long de notre cursus d'ingénieurs, contribuant ainsi au bon déroulement de notre formation.

Résumé

La quatrième révolution industrielle est caractérisée par une disponibilité accrue de la technologie numérique et de la connectivité, entraînant un changement fondamental dans les processus de fabrication. La maintenance prédictive (PdM) est identifiée comme un facteur essentiel pour une transformation numérique réussie des entreprises, car elle peut augmenter les cycles de vie des machines, réduire les temps d'arrêt et les coûts associés, et améliorer la qualité et les taux de production. Dans le domaine de la PdM, où la planification proactive et l'intervention opportune sont cruciales pour prévenir les pannes, les algorithmes de machine learning (ML) jouent un rôle vital.

Dans ce mémoire, en nous appuyant sur un état de l'art, nous avons réalisé une comparaison des algorithmes de ML les plus couramment utilisés pour la PdM. Nous avons identifié les critères essentiels pour atteindre le succès du modèle dans l'application de la PdM. Une sélection de trois algorithmes a été effectuée. Par la suite, nous avons constaté l'insuffisance de la sélection réalisée et la nécessité d'un processus spécifique pour l'application du ML à la PdM. Nous avons donc proposé une deuxième sélection d'algorithmes basée sur la méthode de vote pondéré et nous avons proposé notre processus en suivant l'approche DMAIC. Enfin, la pertinence de notre processus et de la deuxième sélection a été démontrée dans le contexte de deux problèmes de maintenance.

Mots clés : Maintenance prédictive, Apprentissage automatique, Classification.

Abstract

The fourth industrial revolution is characterized by an increased availability of digital technology and connectivity, leading to a fundamental change in manufacturing processes. Predictive maintenance (PdM) is identified as a key factor for successful digital transformation of businesses, as it can increase machine lifecycles, reduce downtime and associated costs, and improve quality and production rates. In the field of PdM, where proactive planning and timely intervention are crucial in preventing breakdowns, machine learning (ML) algorithms play a vital role.

In this paper, based on a literature review, we conducted a comparison of the most commonly used ML algorithms for PdM. We identified the essential criteria for achieving model success in PdM application. A selection of three algorithms was made. Subsequently, we found the insufficiency of the selected algorithms and the need for a specific process for applying ML to PdM. Therefore, we proposed a second algorithm selection based on the weighted voting method, and following the DMAIC approach, we proposed our process. Finally, the relevance of our process and the second selection was demonstrated in the context of two problems of maintenance.

Keywords : Predictive maintenance, Machine learning, Classification.

ملخص

تتميز الثورة الصناعية الرابعة بتوفر متزايد للتكنولوجيا الرقمية والاتصال ، مما يؤدي إلى تغيير أساسي في عمليات التصنيع. تم تحديد الصيانة التنبؤية (PdM) على أنها عامل تمكين مهم للتحويل الرقمي الناجح للأعمال التجارية ، حيث يمكنها زيادة دورات حياة الماكينة وتقليل وقت التوقف عن العمل والتكاليف المرتبطة به وتحسين الجودة ومعدلات الخدمة الإنتاج. في PdM ، حيث يعد التخطيط الاستباقي والتدخل في الوقت المناسب أمرًا بالغ الأهمية لمنع الانقطاعات ، تلعب خوارزميات التعلم الآلي (ML) دورًا حيويًا.

في هذه الأطروحة ، بناءً على أحدث ما توصلت إليه التكنولوجيا ، قمنا بإجراء مقارنة بين خوارزميات ML الأكثر استخدامًا لـ PdM. لقد حددنا المعايير الأساسية لتحقيق نجاح النموذج في تطبيق PdM. تم اختيار ثلاث خوارزميات بعد ذلك ، لاحظنا عدم كفاية الاختيار الذي تم تنفيذه والحاجة إلى عملية محددة لتطبيق ML على PdM. لذلك اقترحنا اختيارًا ثانيًا للخوارزميات بناءً على طريقة التصويت الموزون واتباع نهج DMAIC ، اقترحنا عمليتنا. أخيرًا ، تم توضيح أهمية عمليتنا والاختيار الثاني في سياق مشكلتي صيانة.

كلمات مفتاحية :

الصيانة التنبؤية ، تعلم الآلة ، التصنيف.

Table des matières

Remerciements	1
Résumé	2
Abstract	3
4	ملخص
Introduction générale	1
1 La Maintenance Prédictive	4
1.1 Introduction	5
1.2 Industrie 4.0	5
1.2.1 Définition de l'industrie 4.0	5
1.2.2 Les technologies de l'industrie 4.0 :	5
1.3 La maintenance industrielle	7
1.3.1 Définition de la maintenance industrielle	7
1.3.2 Evolution de la Maintenance Industrielle	7
1.3.3 Les types de maintenance	8
1.4 La maintenance prédictive	10
1.4.1 Données essentielles pour la maintenance prédictive	10
1.4.2 Les sources de données pour la maintenance prédictive :	10
1.4.3 Démarche d'une approche de maintenance prédictive	11
1.4.4 Techniques de surveillance utilisées en maintenance prédictive	11
1.4.5 Objectif de la maintenance prédictive	12
1.4.6 Avantages de la maintenance prédictive	13
1.4.7 Limites de la maintenance prédictive	13
1.5 Conclusion	14
2 La Machine Learning	15
2.1 Introduction	16
2.2 L'intelligence artificielle	16
2.3 Machine learning	17
2.3.1 Définition de la Machine learning	17
2.3.2 Approches de la Machine Learning	17
2.4 Apprentissage automatique supervisé	18
2.4.1 Processus de l'apprentissage automatique supervisé	18

2.4.2	Algorithmes d'apprentissage automatique supervisé prédominants en maintenance prédictive	19
2.4.3	Analyse des avantages et des limites des algorithmes	21
2.5	La machine Learning pour la maintenance prédictive	23
2.5.1	Étude de l'existant	23
2.5.2	Étude comparative et première sélection des algorithmes	26
2.5.3	Discussion	28
2.6	Conclusion	29
3	Modélisation de processus de PdM basé sur la ML	30
3.1	Introduction	31
3.2	Démarche DMAIC : Modélisation de notre processus	31
3.3	Définir	32
3.3.1	Problématique	32
3.3.2	Objectif	32
3.4	Mesurer	32
3.4.1	Amélioration de la compréhension des exigences actuelles en PdM	33
3.4.2	Limitations des méthodes de choix d'algorithmes et de processus de ML	35
3.4.3	Lacunes à améliorer dans l'application de ML à la PdM	35
3.5	Analyser	35
3.5.1	Analyse des informations recueillies	35
3.5.2	Les avantages potentiels de notre processus	36
3.6	Améliorer	36
3.6.1	Amélioration de processus standard de ML	36
3.6.2	Modélisation de notre processus	39
3.7	Contrôler	39
3.8	Conclusion	40
4	Tests et Résultats	41
4.1	Introduction	42
4.2	La méthodologie expérimentale	42
4.3	Description et analyse des scénarios de test	42
4.3.1	Scénario 1	42
4.3.2	Scénario 2	43
4.4	Description des données	44
4.4.1	Base de données 1	44
4.4.2	Base de données 2	44
4.5	Préparation des données	45
4.5.1	Base de données 1	45
4.5.2	Base de données 2	48
4.6	Implémentation des algorithmes et résultats	49
4.6.1	Sélection d'algorithmes	50
4.6.2	Construction de modèles	52
4.6.3	Optimisation des hyperparamètres	53
4.6.4	Résultats des tests	58

Table des matières

4.6.5	Comparaison entre la première et la deuxième sélection	63
4.6.6	Discussion	64
4.7	Comparaison avec un autre travail	65
4.8	Conclusion	65
Conclusion et perspectives		67
Annexes		76
A Présentation de l'entreprise Condor		77
B Métriques d'évaluation		79
C Optimisation des hyperparamètres		82

Table des figures

1.1	Systèmes cyber-physiques.	6
1.2	L'évolution des approches de maintenance industrielle.	8
1.3	Les types de maintenance.	9
1.4	Les six étapes-clés de l'analyse prédictive [14].	11
2.1	Exemple d'utilisation de l'AI dans le domaine de PdM.	16
2.2	Taxonomie des différentes approches du machine learning.	17
2.3	Apprentissage automatique supervisé.	18
2.4	Le processus de machine learning standard [31]	19
2.5	Structure d'un arbre de décision	19
2.6	Le modèle SVM	20
2.7	Structure de RF	20
2.8	Le modèle de kNN	21
2.9	Structure d'ANN	21
2.10	Mind map	23
2.11	Construction du modèle et approche d'évaluation	24
2.12	La méthode proposée	25
2.13	Nombre de papier concernes par chaque critère	27
3.1	Les phases de la démarche DMAIC	31
3.2	Diagramme en toile d'araignée	33
3.3	Les éléments de coûts couramment utilisés dans les systèmes de production pour optimiser les programmes de PdM [35]	34
3.4	Processus de sélection de la stratégie de maintenance	37
3.5	Choix de l'approche de ML	37
3.6	Processus de ML pour la PdM	39
4.1	Analyse de premier scénario	43
4.2	Analyse du deuxième scénario	44
4.3	les données d'entrées et la sortie de premier problème	44
4.4	les données d'entrées et la sortie de deuxième problème	45
4.5	Visualisation de la répartition des classes de BDD 1	45
4.6	Analyse de la corrélation entre les variables	46
4.7	Visualisation des valeurs aberrantes	46
4.8	Suppression des valeurs aberrantes	47
4.9	La sélection des fonctionnalités	47
4.10	Encodage des données	47
4.11	Équilibrage des classes	48
4.12	Visualisation de la répartition des classes de BDD 1	48

4.13	Visualisation de la corrélation entre les variables	49
4.14	Nombre de données manquantes	49
4.15	Remplacement des données manquantes	49
4.16	Code de RF	52
4.17	Code de DT	52
4.18	Code de SVM	53
4.19	Code de KNN	53
4.20	Code de ANN	53
4.21	Code d'optimisation des hyperparamètres pour RF	54
4.22	Code d'optimisation des hyperparamètres pour DT	55
4.23	Code d'optimisation des hyperparamètres pour SVM	55
4.24	Code d'optimisation des hyperparamètres pour KNN	56
4.25	Code d'optimisation des hyperparamètres pour ANN	56
4.26	Matrice de confusion de RF	58
4.27	Matrice de confusion de KNN	58
4.28	Matrice de confusion de DT	58
4.29	La courbe ROC pour le premier scénario	60
4.30	Matrice de confusion RF	61
4.31	Matrice de confusion SVM	61
4.32	Matrice de confusion ANN	61
4.33	La courbe ROC pour le 2 éme scenario	62
A.1	Organigramme de l'entreprise	78
B.1	Matrice de confusion	79

Liste des tableaux

2.1	Avantages et limites des techniques de ML	22
2.2	A summary of the state of the art in ML applied to PdM	26
2.3	Evaluation des algorithmes selon les critères sélectionnés.	28
3.1	Collecte d'informations supplémentaires	32
3.2	Métriques d'évaluation et besoins de maintenance	38
3.3	Exemple d'un tableau pondéré	38
4.1	Sélection d'algorithmes adaptés au scénario 1	51
4.2	Sélection d'algorithmes adaptés au scénario 2	51
4.3	Les meilleurs hyperparamètres de RF	57
4.4	Les meilleurs hyperparamètres de KNN	57
4.5	Les meilleurs hyperparamètres de DT	57
4.6	Les meilleurs hyperparamètres de RF	57
4.7	Les meilleurs hyperparamètres de ANN	57
4.8	Les meilleurs hyperparamètres de SVM	58
4.9	Précision et spécificité obtenues après l'optimisation des hyperparamètres .	59
4.10	Sensibilité et F1 score obtenues après l'optimisation des hyperparamètres .	61
4.11	Tableau comparatif des résultats des deux méthodes de sélection (scénario 1)	63
4.12	Tableau comparatif des résultats des deux méthodes de sélection (scénario 2)	64
4.13	Comparaison entre la méthode de vote pondéré et la méthode basée sur la somme des évaluations	65

Liste des sigles et acronymes

PdM	<i>Maintenance prédictive (Predictive Maintenance)</i>
ML	<i>Apprentissage automatique (Machine Learning)</i>
Cps	<i>Système cyber-physique (Cyber Physical System)</i>
IoT	<i>Internet des objets (Internet of Things)</i>
PvM	<i>Maintenance préventive (Preventive Maintenance)</i>
R2F	<i>Run To Failure</i>
CBM	<i>Condition-Based Maintenance</i>
QVQQ	<i>Quantité de variables qualitatives et quantitatives</i>
AI	<i>Intelligence artificielle (Artificial Intelligence)</i>
IoS	<i>Internet des services (internet of services)</i>
I-ERP	<i>Planification des ressources de l'entreprise intelligente (Intelligent Enterprise Resource Planning)</i>
DMAIC	<i>Approche DMAIC (Define, Measure, Analyze, Improve, Control)</i>
AutoML	<i>Machine Learning Automatique</i>
DT	<i>Arbre de décision (Decision Tree)</i>
RF	<i>Forêt aléatoire (Random Forest)</i>
KNN	<i>Voisinage le plus proche (K-Nearest Neighbors)</i>
SVM	<i>Machine à vecteurs de support (Support Vector Machine)</i>
ANN	<i>Réseau de neurones artificiels (Artificial Neural Network)</i>

Liste des tableaux

ARIMA	<i>Modèle autorégressif intégré à moyenne mobile (Autoregressive Integrated Moving Average)</i>
NASA	<i>National Aeronautics and Space Administration</i>
TP	<i>True Positive (Vrai Positif)</i>
TN	<i>True Negative (Vrai Négatif)</i>
FP	<i>False Positive (Faux Positif)</i>
FN	<i>False Negative (Faux Négatif)</i>

Introduction générale

Contexte

À la lumière de la tendance vers la fabrication intelligente et l'avancée de l'Internet des objets (IoT), de l'exploration de données et de l'intelligence artificielle (IA), entre autres, la maintenance prédictive est proposée comme un nouveau paradigme de maintenance visant à effectuer uniquement des opérations de maintenance lorsque les modèles analytiques prévoient des pannes ou des dégradations spécifiques [23]. Les coûts liés à la maintenance représentent de 15 à 60% des dépenses opérationnelles de toute la production [17]. Afin de réduire les coûts liés aux temps d'arrêt et aux produits défectueux, notamment dans les industries manufacturières de pointe très concurrentielles, la gestion efficace des activités de maintenance devient cruciale [38]. Elle peut éviter le remplacement inutile de l'équipement, réduire les temps d'arrêt des machines, trouver la cause profonde de la défaillance et ainsi économiser des coûts et améliorer l'efficacité [16]. Pour permettre une détection précoce des éventuelles défaillances, les systèmes de maintenance prédictive utilisent des outils de prédiction basés sur des données historiques, des indicateurs de santé définis ad hoc, des méthodes d'inférence statistique et des méthodologies d'ingénierie. Selon [35], les méthodes d'inférence statistique, en particulier celles qui intègrent la machine learning (ML), sont les plus efficaces pour créer des modèles pour les problèmes de maintenance.

Questions de recherche

- Quelle méthode permet de sélectionner l'algorithme de ML le plus approprié pour résoudre les problèmes de maintenance ?
- Comment choisir judicieusement les métriques d'évaluation des algorithmes pour les problèmes de maintenance ?
- Comment améliorer le processus standard de ML pour l'appliquer à la PdM ?

Objectif

Notre objectif est de combler les lacunes identifiées dans l'analyse de la littérature en proposant un processus qui prend en considération les aspects spécifiques de la maintenance ainsi que les spécifications du problème, en particulier son objectif. Nous visons à

faire le bon choix d'algorithme en fonction des spécifications du problème. De plus, nous cherchons à améliorer l'évaluation des performances des algorithmes en sélectionnant les bonnes métriques d'évaluation adaptées à notre scénario.

Démarche

Notre démarche de travail repose sur six étapes clés, à savoir :

1. Réaliser une revue de littérature approfondie sur l'application de la machine learning à la maintenance prédictive. Cette étape vise à recueillir les connaissances existantes et à comprendre les avancées et les pratiques courantes dans ce domaine.
2. Effectuer une analyse comparative des algorithmes de machine learning les plus utilisés en maintenance prédictive. À partir de cette analyse, nous sélectionnerons trois algorithmes.
3. Identifier les lacunes existantes dans ce domaine, en mettant particulièrement en évidence les limitations du sélection précédente. Nous examinerons ces lacunes, afin de proposer des solutions pour les combler.
4. Appliquer l'approche DMAIC (Définir, Mesurer, Analyser, Améliorer, Contrôler) pour développer un processus spécifique à l'application de la machine learning à la maintenance prédictive. Ce processus comprendra la méthodologie proposée pour la deuxième sélection des algorithmes, en tenant compte des spécifications du problème et des mesures d'évaluation appropriées.
5. Mettre en pratique notre processus sur deux scénarios de maintenance réels. Le premier scénario sera basé sur notre expérience de stage à l'entreprise Condor, tandis que le second scénario utilisera des données réelles téléchargées à partir de Kaggle.
6. Interpréter les résultats obtenus afin de démontrer l'intérêt de la deuxième sélection proposée, ainsi que l'importance du choix de mesures adaptées aux spécifications du problème et de l'alignement stratégique. Nous mettrons en évidence l'influence de ces éléments sur la prise de décision basée sur les objectifs de maintenance.

Organisation du mémoire

Ce mémoire est organisé en quatre chapitres :

Le premier chapitre "**La maintenance prédictive**" : Introduction à la maintenance, en mettant l'accent sur les différents types de maintenance. Une attention particulière sera accordée à la maintenance prédictive, ses techniques et ses outils.

Le deuxième chapitre "**La machine learning**" : Définition de la ML, en détaillant ses types, ses éléments constitutifs. Une focalisation sera faite sur l'apprentissage supervisé et ses algorithmes. Une comparaison entre ces algorithmes sera réalisée en fonction de critères spécifiques.

Le troisième chapitre “**Démarche DMAIC : Modélisation de notre processus**” : Application de l’approche DMAIC pour proposer un nouveau processus. Dans un premier temps, nous identifierons les lacunes existantes en effectuant une collecte d’informations complémentaires. Ensuite, nous analyserons ces lacunes et proposerons des méthodes pour les combler. Enfin, nous décrirons en détail le nouveau processus proposé.

Le quatrième chapitre “**Tests et résultats** ” : Présentation de deux scénarios de problèmes à résoudre en appliquant le nouveau processus et ces méthodes associées.

Chapitre 1

La Maintenance Prédicative

1.1 Introduction

L'industrie 4.0, caractérisée par l'intégration des technologies numériques avancées dans les processus de fabrication, a ouvert la voie à une nouvelle ère de production industrielle. Dans ce contexte, la maintenance prédictive joue un rôle crucial en exploitant des données et des modèles prédictifs pour anticiper les défaillances et planifier efficacement les activités de maintenance. Ce chapitre explore les différences entre les types de maintenance existants (préventive, corrective et prédictive) et se concentre ensuite sur la maintenance prédictive (PdM) elle-même, en examinant ses fondements théoriques, les types de données utilisés et les étapes clés de sa mise en œuvre. Les avantages et les limites de la maintenance prédictive sont discutés, ainsi que les techniques de surveillance utilisées, mettant en évidence son impact sur la disponibilité des équipements, la productivité, les coûts et la sécurité des opérations industrielles.

1.2 Industrie 4.0

1.2.1 Définition de l'industrie 4.0

Les quatre grandes révolutions industrielles sont illustrées par la première révolution qui a été celle des machines à vapeur vers 1780. La deuxième fut celle de la production de masse avec l'électricité et le travail à la chaîne à partir de 1870. Ensuite vient la troisième révolution industrielle, celle de l'automatisation, de l'électronique et des robots. De nos jours, plusieurs industries sont à cette étape dans leurs usines et d'autres sont encore à la deuxième étape[37]. Aujourd'hui, nous sommes au stade de la quatrième grande révolution industrielle également connue sous le nom Industrie 4.0 c'est un terme qui a été introduit pour la première fois à la foire de Hanovre en 2011 [7]. Il s'agissait d'un programme stratégique visant à développer des systèmes de production avancés dans le but d'augmenter la productivité et l'efficacité de l'industrie nationale [16]. L'industrie 4.0 est marquée par l'intégration technique des systèmes cyber-physiques (CPS) dans la fabrication et l'internet des objets (IoT) et des services (IoS) dans les processus industriels [31]. Non-seulement l'IoT et le CPS, mais d'autres technologies importantes telles que le cloud computing, l'intelligence artificielle (AI), la réalité augmentée, le jumeau numérique, la planification intelligente des ressources d'entreprise (I-ERP) et d'autres technologies innovantes jouent un rôle crucial dans la transformation numérique [17]. Le véritable potentiel de l'Industrie 4.0 se trouve dans le réseau des machines connectées numériquement qui produisent et partagent des informations [4]. Toutes les activités sont impactées par cette évolution : la maintenance opérationnelle, la supply chain globale, entre autres.

1.2.2 Les technologies de l'industrie 4.0 :

Capteurs intelligents :

Un capteur est un dispositif qui convertit une grandeur physique en une grandeur utilisable pour recueillir des informations sur l'environnement. Il permet de mesurer dif-

férentes grandeurs telles que les tensions électriques, les vibrations, les températures, les pressions, les sons, la consommation d'énergie, etc [33] . Un capteur intelligent intègre des composants de traitement numérique de données, lui permettant de comprendre et d'interpréter les données collectées de manière autonome [19] .

Systèmes cyber-physiques :

Cps est souvent appelé système informatique de nouvelle génération qui utilise des techniques de calcul intelligentes associées au monde physique et aux unités de calcul [41]. Ils peuvent recueillir des informations sur eux-mêmes et sur leur environnement, les analyser et les traiter, se connecter et communiquer avec d'autres systèmes et agir. Cette technologie peut amener le monde réel dans le virtuel voir (1.1) [35]. Les technologies de communication CPS jouent un rôle crucial dans la maintenance, l'amélioration de l'efficacité, la qualité, les prévisions, les opérations de service à distance et les diagnostics.

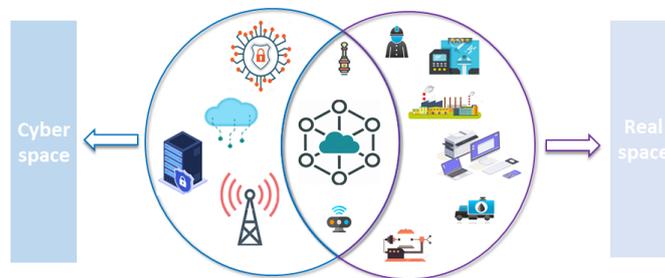


FIG. 1.1 : Systèmes cyber-physiques.

Internet des objets

C'est un réseau qui relie des systèmes cyber-physiques et offre une interconnexion et une interaction entre des objets physiques. IoT est capable de collecter, trier, synchroniser et organiser des données provenant de nombreuses sources au sein d'une usine ou d'une organisation, ainsi que de fournir une plate-forme de gestion de données sans fil et liée avec des capacités de diffusion et de traitement en temps réel [25].

Big Data

Les Big Data font référence aux données massives qui ne pourraient pas être collectées, stockées, gérées, partagées, analysées et calculées par des outils de données standard dans un délai tolérable [34].

Cloud Computing

Le cloud computing fait référence à la fourniture de services informatiques, tels que les logiciels, le stockage, les réseaux, les serveurs, les analyses, les bases de données et l'intelligence, sur Internet (le Cloud) afin de faciliter une innovation plus rapide [17]. Le cloud computing a la capacité de sauvegarder des données sur un fournisseur de serveur

Internet facilement accessible via un accès à distance. Cette technologie, lorsqu'elle est appliquée à la maintenance relie les gestionnaires, les équipements et les techniciens sur une plate-forme unique pour accélérer le travail et permettre un suivi et un contrôle des coûts en temps réel.

1.3 La maintenance industrielle

1.3.1 Définition de la maintenance industrielle

La maintenance industrielle peut être définie comme « l'ensemble de toutes les actions techniques, administratives et de management durant le cycle de vie d'un bien, destinées à le maintenir ou à le rétablir dans un état dans lequel il peut accomplir la fonction requise » [1]. Dans la vision traditionnelle, la maintenance vise à garantir l'intégrité opérationnelle des équipements, particulièrement leur disponibilité, et tend de manière générale à discerner les phénomènes défaillants et à y remédier, de manière à garantir que le système remplisse convenablement sa fonction conçue. Néanmoins, les fonctions de maintenance ne se bornent plus seulement à mettre en place les modalités permettant d'assurer le "service des bien", des impératifs de qualité, de sécurité et de rentabilité sont apparues.

1.3.2 Evolution de la Maintenance Industrielle

L'évolution de la maintenance a été marquée par une progression d'approche corrective à l'approche préventive, puis prédictive, à la Maintenance 4.0 voir (1.2). Chacune de ces phases a surmonté les limites des approches précédentes pour fournir une maintenance plus efficace et intelligente [7].

Maintenance 1.0 :

Étant donné que la demande à l'époque n'était pas aussi élevée qu'aujourd'hui, ce facteur n'était pas encore un problème. Cette phase se caractérise par des pratiques de maintenance manuelles et réactives, axées sur la réparation des équipements en cas de panne.

Maintenance 2.0:

En réponse à un marché de plus en plus compétitif et à une augmentation de la production industrielle. La surutilisation des machines a entraîné une augmentation des pannes et des coûts de réparation. Afin de faire face à ces problèmes, la maintenance préventive a été mise en place, consistant à maintenir les équipements en bon état avant qu'ils ne tombent en panne, ce qui permettait de prolonger leur durée de vie et de réduire les coûts de réparation.

Maintenance 3.0 :

L'accent a été mis sur la prévention des pannes grâce à l'utilisation des technologies de l'information et de la communication, notamment l'intégration de capteurs et de systèmes de surveillance, pour permettre la maintenance prédictive.

Maintenance 4.0 :

Cette phase implique l'intégration des technologies de l'industrie 4.0, telles que l'IoT, l'analyse de big data et l'IA, pour permettre la maintenance prédictive, améliorer la prise de décision en matière de maintenance et optimiser le processus de maintenance [19].

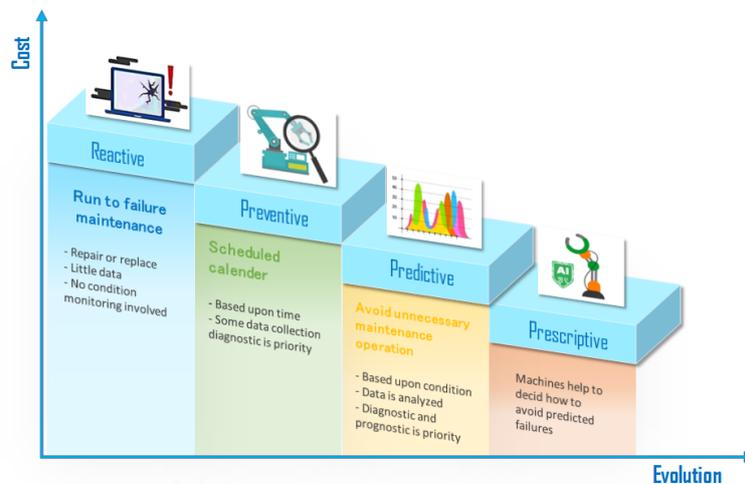


FIG. 1.2 : L'évolution des approches de maintenance industrielle.

1.3.3 Les types de maintenance

Il existe deux principales familles de maintenance : la maintenance corrective et la maintenance préventive, Ils peuvent également être divisés en sous-catégories (voir (1.3)). La maintenance corrective est celle que le système subit lorsque la panne est déjà présente et qu'il faut réparer. La maintenance préventive est celle qui permet d'anticiper et de prévenir les défaillances [42].

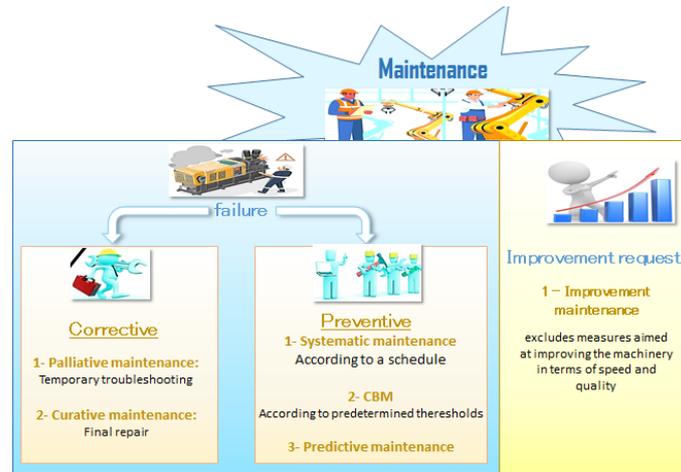


FIG. 1.3 : Les types de maintenance.

Maintenance corrective

Maintenance exécutée après détection d'une panne et destinée à remettre un bien dans un état dans lequel il peut accomplir une fonction requise.» [1]. Habituellement, cette maintenance provoque un arrêt de la production, impliquant une réduction de la production et une augmentation des coûts. La maintenance corrective est utilisée pour les processus où les pannes n'ont pas d'impact critique sur la production [42]. Il existe deux types de maintenance corrective :

- **Maintenance curative** : Ce type de maintenance permet de remettre définitivement en état le système après l'apparition d'une défaillance. Elle se caractérise par la recherche des causes initiales d'une défaillance en vue de réparer l'équipement.
- **Maintenance palliative** : Opération destinée à remettre un équipement dans un état provisoire de fonctionnement de manière à ce qu'il puisse assurer une partie des fonctions requises.

Maintenance préventive

Maintenance exécutée à des intervalles prédéterminés ou selon des critères prescrits et destinée à réduire la probabilité de défaillance ou la dégradation du fonctionnement d'un bien. » [1]. Cette maintenance est généralement appliquée en dehors du temps de production. Trois formes de politique de maintenance préventive peuvent être mises en œuvre :

- **Maintenance systématique** : Lorsque l'intervention de maintenance est exécutée à intervalles fixes et prédéfinis, on parle de la maintenance systématique. Ce type de maintenance est déclenché selon un calendrier et se traduit par le remplacement périodique de pièces, sans contrôle préalable et quel que soit l'état de dégradation des équipements.
- **Maintenance conditionnelle** : « Maintenance préventive basée sur une surveillance du fonctionnement du bien et/ou des paramètres significatifs de ce fonctionnement

intégrant les actions qui en découlent. La surveillance du fonctionnement et des paramètres peut être exécutée selon un calendrier, ou à la demande, ou de façon continue. » [1].

- **Maintenance prédictive** : La PdM est une stratégie de maintenance couramment utilisée et moderne qui permet de réduire les coûts associés à la maintenance préventive en prévoyant de manière précise le moment où les pièces doivent être remplacées, seulement quand cela est vraiment nécessaire [33], c'est-à-dire ni trop tôt (maintenance préventive), ni trop tard (maintenance corrective). Elle se base sur l'analyse de données collectées par des capteurs intelligents positionnés sur l'équipement [43], ce qui permet d'optimiser l'utilisation de la machine avant de procéder à sa réparation ou à son remplacement. Les technologies clés impliquées dans la PdM sont les technologies de collecte et d'analyse de données, telles que l'IoT, le cloud computing, l'analyse prédictive (comme les réseaux de neurones, les algorithmes évolutifs, l'apprentissage automatique, le raisonnement probabiliste) [3].

1.4 La maintenance prédictive

1.4.1 Données essentielles pour la maintenance prédictive

La maintenance prédictive repose sur l'utilisation de big data collectées, stockées et analysées pour identifier des modèles et prévoir les pannes d'équipement.

Les données essentielles pour la maintenance prédictive selon [38], [3] sont :

- **L'historique des pannes** : Il est crucial d'inclure un nombre suffisant d'exemples de fonctionnement normal et échecs dans les données de formation.
- **L'historique de maintenance/réparation** : Cet historique comprend des informations telles que les composants remplacés et tâches de maintenance préventive effectuées.
- **Les conditions de la machine** : Les données utilisées doivent inclure des fonctions qui capturent les schémas de vieillissement et toutes les anomalies pouvant entraîner une baisse des performances.
- **Les données machines** : Désigne les données opérationnelles historiques des actifs, incluant l'état de la machine, les paramètres clés, et les indicateurs de disponibilité, d'arrêt et d'alerte.

1.4.2 Les sources de données pour la maintenance prédictive :

Les sources des données décrites précédemment sont [34] :

- **Les ressources de fabrication** : Données provenant des usines intelligentes, des équipements, des matériaux, des produits et des systèmes de service, ainsi que des données environnementales.
- **La gestion des systèmes d'information de fabrication** : Tels que les ERP et les systèmes assistés par ordinateur.
- **Données externes** : Y compris les conditions météorologiques, les données des fournisseurs, etc.

1.4.3 Démarche d'une approche de maintenance prédictive

Un système de prédiction s'appuie exclusivement sur cette analyse descriptive ; sa construction est déterminée par les étapes suivantes[14] :

- Identification et extraction des données utiles en utilisant diverses sources.
- Prétraitement des données : normalisation, correction, etc.
- Construction d'un modèle prédictif à partir d'algorithmes statistiques et d'apprentissage automatique.
- Evaluation de l'efficacité et de la précision du modèle prédictif.
- Utilisation du modèle pour les tâches de l'aide à la décision.
- Maintien de l'efficacité du modèle prédictif avec une évaluation continue.

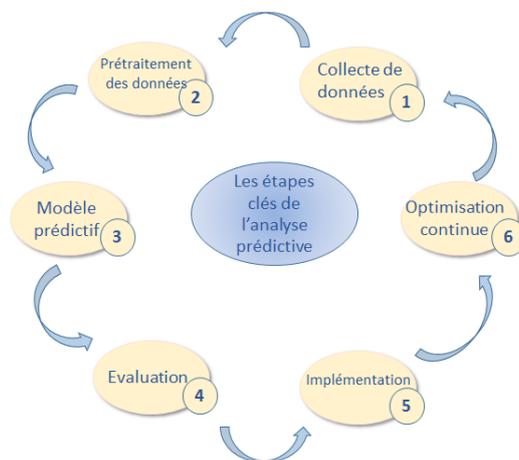


FIG. 1.4 : Les six étapes-clés de l'analyse prédictive [14].

1.4.4 Techniques de surveillance utilisées en maintenance prédictive

La surveillance d'un équipement est assurée en relevant périodiquement un indicateur d'état de dégradation (ou de performance). Il existe différentes techniques d'analyse telles :

L'analyse vibratoire

La méthode la plus connue et la plus largement utilisée, adaptée aux composants mécaniques et aux machines industrielles en fonctionnement, permettant de détecter la majorité des défauts susceptibles d'apparaître dans les machines tournantes [15].

L'analyse d'huile

Elle consiste à prélever un échantillon d'huile et de l'analyser pour déduire l'état de l'équipement, est appliquée à toutes les machines contenant des fluides de lubrification [15].

La thermographie infrarouge

Un nouvel intérêt dans le domaine de la maintenance, initialement réservée au contrôle des installations électriques. Son utilisation pour la surveillance des machines tournantes et la détection des défauts liés à un échauffement anormal est encore limitée. Cette technique permet des mesures à distance et fournit instantanément une image thermique de la zone inspectée [15].

L'analyse acoustique

Détection des bruits anormaux à l'aide de microphones positionnés à distance de l'équipement [15].

L'objectif de ces techniques est de mettre en place des signatures caractéristiques de l'état de fonctionnement d'un système à partir de mesures réalisées sur celui-ci. L'utilisation de ces techniques en maintenance engendre des coûts de mise en œuvre qui ne sont pas toujours négligeables.

1.4.5 Objectif de la maintenance prédictive

La maintenance prédictive vise à optimiser la gestion des équipements et des machines en mettant en place des stratégies et des actions préventives basées sur des données et des modèles prédictifs. Parmi les objectifs clés, nous pouvons citer [23] :

- Réduction des stocks de pièces de rechange.
- Prolongation de la durée de vie des machines et des composants.
- Minimisation des coûts.
- Réduction des temps d'arrêt.
- Amélioration de la disponibilité des équipements.
- Augmentation de la productivité.

- Accroissement de la fiabilité des machines.
- Amélioration de la sécurité des employés.

1.4.6 Avantages de la maintenance prédictive

En comparant avec les stratégies de maintenance corrective et préventive, la maintenance prédictive présente les avantages suivants [7], [4], [3] :

- Allonger la durée de vie des équipements.
- Amélioré la fiabilité des équipements permettant ainsi d'optimiser la production.
- Réduire les coûts liés à la maintenance.
- Diminuer le nombre d'interruptions des machines pour des opérations de maintenance.
- Diminuer le nombre de pannes.
- Mieux planifier les interventions.
- Mieux préparer les équipes d'intervention.
- Mieux échanger entre les professionnels de maintenance et les équipes de production.
- Améliorer la sécurité.
- Mieux anticiper et gérer les besoins de pièces détachées.

1.4.7 Limites de la maintenance prédictive

- Les données peuvent être mal interprétées, entraînant de fausses demandes de maintenance [7].
- Il est coûteux de mettre en place un système IoT complet [3].
- L'analyse prédictive peut ne pas prendre en compte les informations contextuelles, comme l'âge de l'équipement ou les conditions météorologiques.
- Les activités de maintenance préventive peuvent être déclenchées par des échéanciers plutôt que l'état de la machine d'origine [4].

1.5 Conclusion

En conclusion, ce chapitre a mis en évidence l'importance capitale de la PdM dans le contexte de l'Industrie 4.0. Nous avons exploré en détail les objectifs de la maintenance prédictive ainsi que les différentes techniques de surveillance utilisées. La PdM se révèle être un levier majeur pour réduire les coûts, minimiser les temps d'arrêt et atténuer les risques opérationnels. Les prochains chapitres se concentreront sur l'application rigoureuse des techniques de machine learning en matière de PdM. Nous examinerons en profondeur les modèles utilisés pour prédire les défaillances et planifier les interventions de maintenance. Des études de cas concrètes et des exemples pratiques seront présentés pour illustrer de manière éloquente l'efficacité inégalée du ML dans l'optimisation des performances de la maintenance prédictive dans le cadre de l'Industrie 4.0.

Chapitre 2

La Machine Learning

2.1 Introduction

La Machine Learning est un domaine de recherche en pleine expansion qui offre de nombreuses opportunités dans divers domaines, y compris celui de la PdM. Ce chapitre vise à fournir une vision d'ensemble de l'état actuel de ML et de son utilisation spécifique dans le domaine de la maintenance prédictive. Nous mettons en évidence les défis et les lacunes existantes, tout en soulignant l'importance de développer des approches adaptées qui prennent en compte les spécificités des données et des objectifs propres à la PdM.

2.2 L'intelligence artificielle

Au sein de la transition vers l'industrie 4.0, l'intelligence artificielle est une technologie puissante qui pallie les insuffisances et les limites des méthodes et approches industrielles conventionnelles [12]. Elle consiste à simuler l'intelligence humaine dans des machines conçues pour réfléchir et agir comme des êtres humains. L'IA comprend plusieurs domaines tels que la ML et le deep learning, qui se focalisent sur la création de modèles et d'algorithmes permettant aux machines d'exécuter des tâches spécifiques.

Actuellement, de nombreuses entreprises investissent des milliers d'euros dans la création de systèmes d'IA [31], ce qui a conduit à l'émergence de la PdM en tant que nouveau paradigme de maintenance. Grâce à l'IA, cette approche de maintenance ne se déclenche que lorsque des modèles analytiques prédisent des défaillances ou des dégradations spécifiques [24].

La figure (2.1) présente des exemples concrets de l'utilisation de l'intelligence artificielle dans le domaine de la maintenance.

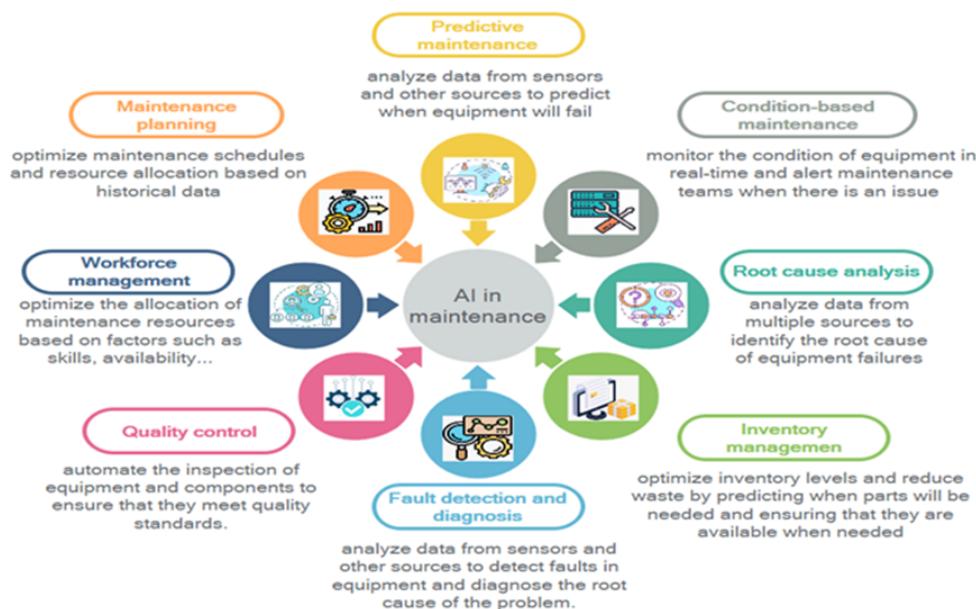


FIG. 2.1 : Exemple d'utilisation de l'AI dans le domaine de PdM.

2.3 Machine learning

2.3.1 Définition de la Machine learning

La ML représente une branche de l'intelligence artificielle qui permet aux ordinateurs de s'instruire sans avoir à être explicitement programmés [28]. Cette branche d'activité repose sur un ensemble d'algorithmes avancés par lesquels les résultats peuvent être prédits selon un modèle établi et entraîné sur la base de données d'entrée historiques et de leurs comportements de sortie. Toutefois, l'efficacité et les avantages du PdM sont variables selon l'approche de ML choisie [29].

2.3.2 Approches de la Machine Learning

Il y a deux grandes approches en ML : l'apprentissage supervisé et l'apprentissage non supervisé, chacune utilisant ses propres méthodes comme illustré dans la figure (2.2).

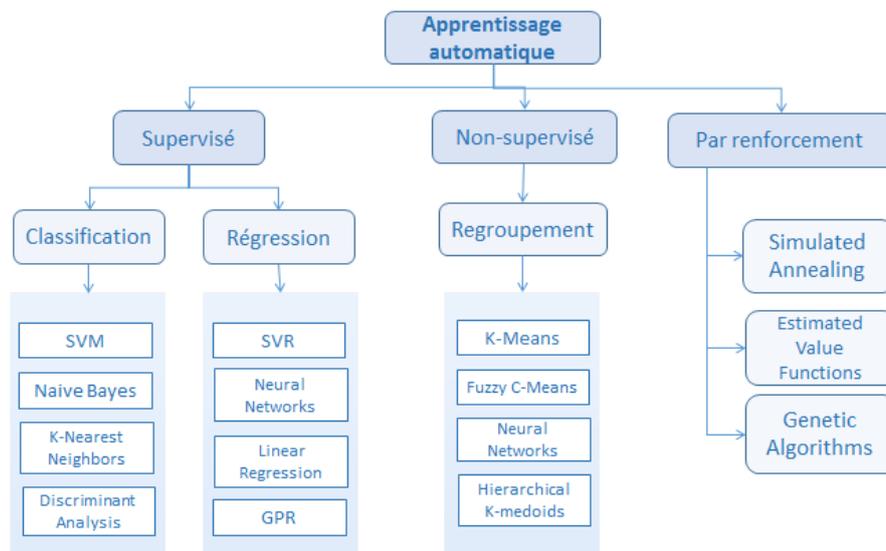


FIG. 2.2 : Taxonomie des différentes approches du machine learning.

Apprentissage automatique non supervisés

L'apprentissage automatique non supervisé se réfère essentiellement à toute méthode de ML qui tente d'apprendre la structure sous-jacente d'un ensemble de données en l'absence d'une sortie identifiée (comme dans l'apprentissage automatique supervisé) ou d'une rétroaction directe (comme dans l'apprentissage par renforcement) [28]. Les tâches les plus courantes de l'apprentissage automatique non supervisé sont les méthodes de clustering, car les groupes de données (labels) ne sont pas connus à l'avance [9].

Le clustering consiste à regrouper des enregistrements ou des observations en classes d'objets similaires (cluster). Les algorithmes de clustering cherchent à segmenter l'ensemble des données en groupes relativement homogènes. Ils cherchent à maximiser l'ho-

mogénéité à l'intérieur de chaque groupe et à la minimiser entre les différents groupes [36].

Pour les problèmes de détection de type de défaillance et de maintenance prédictive, l'apprentissage non supervisé peut s'avérer moins efficace car les techniques de clustering et d'estimation de densité des données historiques ne permettent pas une détection précise des types de défaillances ou des besoins de maintenance [39].

L'apprentissage automatique supervisé

L'apprentissage supervisé est un processus de ML qui implique l'utilisation d'un ensemble de données appelé ensemble d'apprentissage constitué de données d'entrée associées à des valeurs de réponse. Ce processus est décrit en détail dans la figure (2.3).

Dans le domaine de la détection de défaillances et de la maintenance prédictive, l'apprentissage supervisé est la méthode d'apprentissage la plus couramment utilisée, car elle permet d'utiliser les données historiques liées à l'état réel du système.

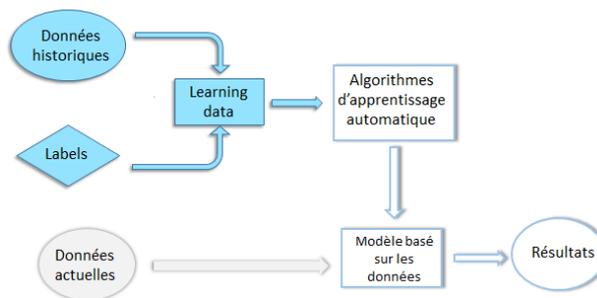


FIG. 2.3 : Apprentissage automatique supervisé.

Du point de vue du machine learning, en considération de la sortie des données, on peut distinguer deux classes de problèmes supervisés [6], [9] :

- **Problème de régression** : Si la sortie consiste en des valeurs continues.
- **Problème de classification** : Si la sortie prend des valeurs catégorielles. Ainsi, on dispose des algorithmes de classification et des algorithmes de régression.

2.4 Apprentissage automatique supervisé

2.4.1 Processus de l'apprentissage automatique supervisé

Les étapes critiques du processus de ML sont l'acquisition des données, le prétraitement des données, la sélection des caractéristiques, la construction du modèle de classification et la validation [24]. Chacune de ces étapes est essentielle et offre un aperçu précieux des performances du processus. La figure (2.4) présente un schéma illustrant les différentes étapes d'un processus de ML.

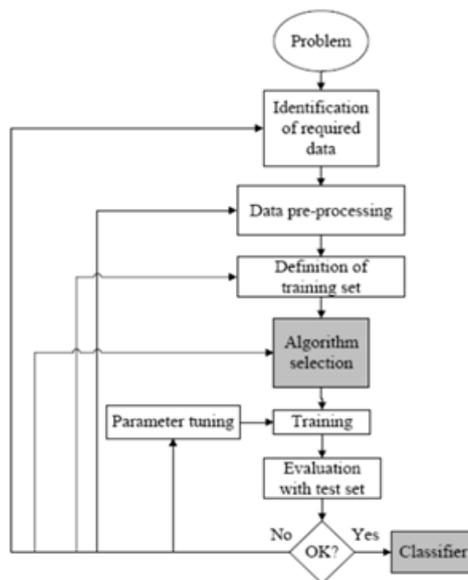


FIG. 2.4 : Le processus de machine learning standard [31]

2.4.2 Algorithmes d'apprentissage automatique supervisé prédominants en maintenance prédictive

Arbre de décision (DT)

Un arbre de décision est une méthode supervisée non paramétrique utilisée pour la classification et la régression [22]. Il est généralement composé d'une racine, de branches, de nœuds de décision et de nœuds feuilles voir(2.5). Chaque nœud interne de l'arbre représente un test sur une caractéristique spécifique, et chaque branche correspond à l'un des résultats possibles de ce test. Les nœuds feuilles représentent le résultat du modèle en fonction des données d'entrée [31]. L'espace d'entrée est divisé de manière récursive en sections plus petites en fonction de la valeur d'une caractéristique ou d'un attribut choisi à chaque nœud, créant ainsi une hiérarchie de modèles [18].

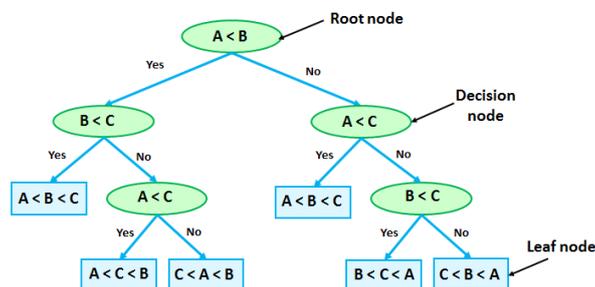


FIG. 2.5 : Structure d'un arbre de décision

Machine à vecteurs de support (SVM)

Les machines à vecteurs de support sont parmi les méthodes de ML les plus récentes [31] utilisées pour la classification et la régression. Les SVM traitent d'un modèle linéaire

généralisé en utilisant l'espace des hypothèses d'une fonction linéaire dans un espace de caractéristiques de grande dimension en créant un hyperplan de partition optimal [27]. L'hyperplan est sélectionné de manière à maximiser la marge entre les deux classes, où la marge est définie comme la distance entre l'hyperplan et les points de données les plus proches de chaque classe voir (2.6).

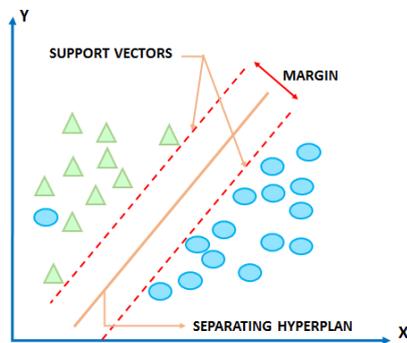


FIG. 2.6 : Le modèle SVM

Random forest (RF)

RF est une technique de ML supervisée utilisée pour les problèmes de classification et de régression. Il s'agit d'une technique d'apprentissage par ensemble qui construit de nombreux arbres de décision, chacun étant entraîné sur un sous-ensemble aléatoire des données et un sous-ensemble aléatoire des caractéristiques, réduisant ainsi le surajustement et renforçant la diversité de l'ensemble voir (2.7). La prédiction finale est ensuite effectuée soit par un vote majoritaire pour les tâches de classification, soit en moyennant les prévisions de tous les arbres de la forêt pour les tâches de régression [24].

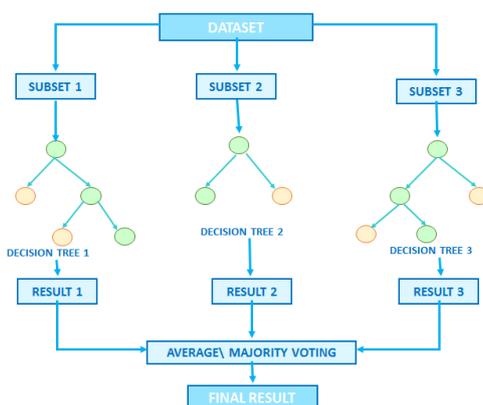


FIG. 2.7 : Structure de RF

K-Nearest Neighbors (KNN)

Le k-NN est un algorithme de classification non paramétrique utilisé en ML. C'est un type d'apprentissage basé sur les instances, où l'algorithme effectue des prédictions pour

un nouveau point de données en se basant sur sa similarité avec les données d'entraînement. Il peut être utilisé pour la classification et la régression, mais est principalement utilisé pour la classification [35]. Le "k" représente le nombre de voisins les plus proches que l'algorithme utilise pour effectuer des prédictions voir(2.8). L'algorithme calcule la distance entre le nouveau point de données et tous les points de données d'entraînement, et sélectionne les K voisins les plus proches en fonction de la distance mesurée (telle que la distance euclidienne ou la distance de Manhattan). L'étiquette de classe du nouveau point de données est ensuite prédite en fonction de la classe majoritaire parmi les K voisins.

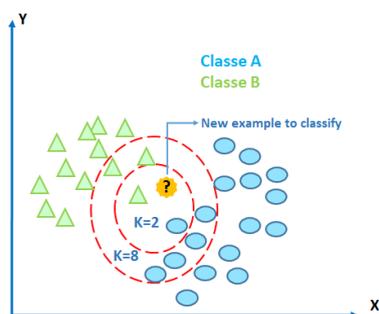


FIG. 2.8 : Le modèle de kNN

Réseaux de neurones artificiels (ANN)

Un ANN est un type de modèle de ML inspiré de la structure et de la fonction du cerveau humain. Il se compose de nœuds interconnectés (neurones) qui travaillent ensemble pour traiter et analyser l'information. Chaque neurone reçoit des entrées d'autres neurones ou de sources externes, traite l'information à l'aide d'une fonction d'activation et envoie des sorties à d'autres neurones ou à la couche de sortie finale voir (2.9). L'architecture du réseau (par exemple, le nombre de neurones dans les couches cachées, les connexions du réseau et les fonctions d'activation) joue un rôle très important dans les performances d'un ANN et dépend généralement du problème à résoudre [35].

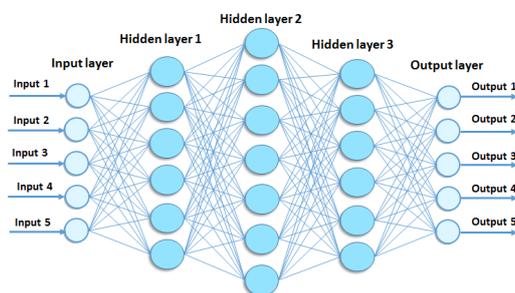


FIG. 2.9 : Structure d'ANN

2.4.3 Analyse des avantages et des limites des algorithmes

Les avantages et les inconvénients des techniques de ML conventionnelles largement utilisées pour la maintenance prédictive sont résumés dans le tableau ci-dessous.

Algorithm	Avantages	Limites
<ul style="list-style-type: none"> • RF 	<ul style="list-style-type: none"> • Gérer des données de grande dimension. • Robuste face aux valeurs aberrantes et au bruit. • Gérer des données déséquilibrées. • Moins sujet au surajustement par rapport aux arbres de décision. 	<ul style="list-style-type: none"> • Surapprentissage avec des données bruitées. • L'interprétation peut poser problème. • Coûteux en termes de calcul, surtout avec de grands ensembles de données.
<ul style="list-style-type: none"> • SVM 	<ul style="list-style-type: none"> • Efficace dans des espaces de grande dimension. • Robuste contre le surajustement. • Gérer des données non linéairement séparables. • Bien fonctionner avec des données structurées ou semi-structurées. 	<ul style="list-style-type: none"> • Nécessite une sélection prudente des paramètres de noyau et de régularisation. • Coûteux en termes de calcul. • Aucune norme pour choisir la fonction de noyau. • Faible efficacité pour les gros volumes de données.
<ul style="list-style-type: none"> • ANN 	<ul style="list-style-type: none"> • Gérer des relations complexes et non linéaires dans les données. • Très adaptable et capable d'apprendre à partir de différents types de données. • Utilisé pour la classification et la régression. • Grande précision de classification et de prédiction. 	<ul style="list-style-type: none"> • Risque de surapprentissage (facilité de surajustement). • De nombreux paramètres de pondération doivent être entraînés. • Nécessite des ressources de calcul plus importantes (coûteux en termes de calcul pour l'entraînement). • Pas de signification physique. • Pas de norme pour décider de la structure du réseau.
<ul style="list-style-type: none"> • KNN 	<ul style="list-style-type: none"> • Simple à mettre en œuvre et à comprendre. • Gérer des relations non linéaires. • Aucun apprentissage requis. • Peu de paramètres à régler. 	<ul style="list-style-type: none"> • Sensible aux valeurs aberrantes et aux données bruitées. • Lent et coûteux en termes de calcul pour les grands ensembles de données. • K doit être déterminé à l'avance.
<ul style="list-style-type: none"> • DT 	<ul style="list-style-type: none"> • Facile à comprendre et à interpréter. • Capable de gérer des données catégorielles et numériques. • Capable de gérer des problèmes multi-classes. • Nécessite peu de préparation des données. • Capable de gérer les valeurs manquantes. • Rapide et efficace pour l'entraînement et la classification. 	<ul style="list-style-type: none"> • Instable. • Facilement sujet au surajustement. • Biaisé en faveur des variables avec de nombreux niveaux. • Ne gère pas bien les données bruitées. • Faible précision de prédiction.

TAB. 2.1 : Avantages et limites des techniques de ML

2.5 La machine Learning pour la maintenance prédictive

2.5.1 Étude de l'existant

Le domaine de ML pour la PdM a suscité un intérêt croissant dans les dernières années, avec de nombreuses études explorant l'utilisation de divers algorithmes et méthodologies pour le diagnostic des défauts, l'évaluation de l'état des outils et la prédiction de durée de vie. Les articles sélectionnés et analysés dans ce domaine sont résumés dans la Mind Map suivante.

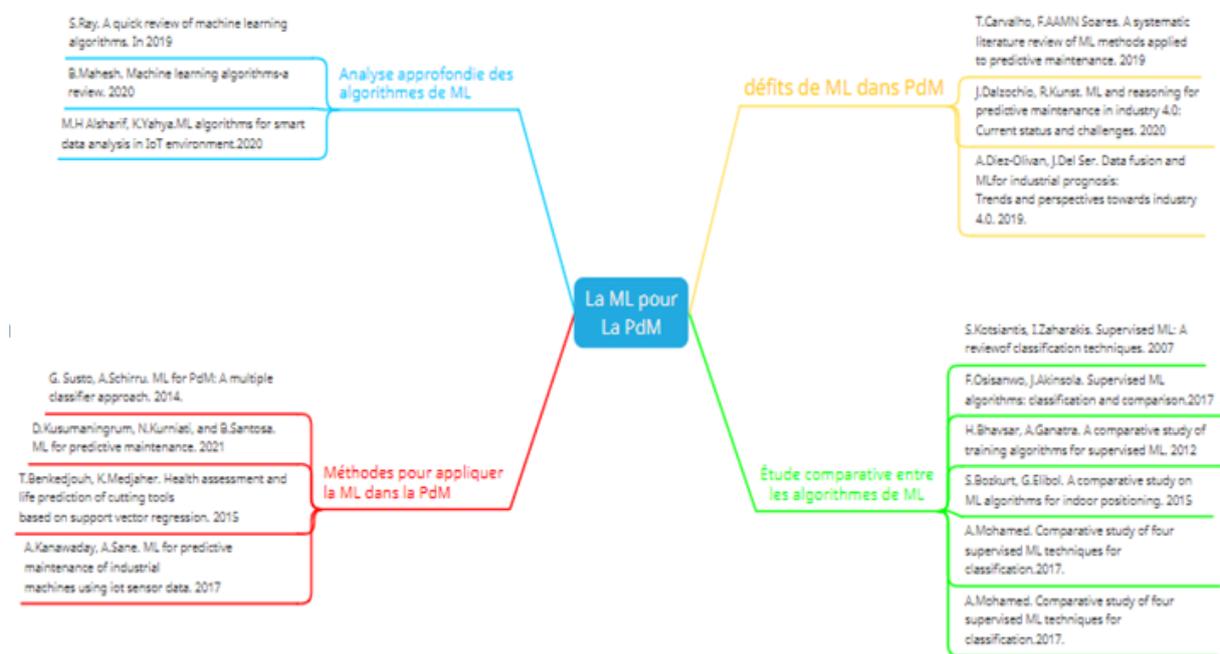


FIG. 2.10 : Mind map

Dans la présentation suivante, nous allons mettre en avant les travaux les plus pertinents dans le domaine de la maintenance prédictive, en décrivant leurs méthodes et en exposant leurs résultats.

Étude d'A.kanawaday et al. (2017) [21]

Contexte, données et algorithmes : Cette étude explore l'utilisation de la prévision de moyenne mobile intégrée autorégressive (ARIMA) sur les données de séries chronologiques collectées à partir de divers capteurs d'une machine à refendre, qui est utilisée pour rembobiner et couper des films d'emballage pour divers clients. Dans le but de prévoir les éventuelles pannes et défauts de qualité en utilisant les algorithmes suivants : Naive Bayes, Support Vector Machine, CART, Deep Neural Network.

L'architecture du système propose : Les modèles supervisés sont formés sur les données historiques tandis que le même ensemble de données est utilisé pour former le

modèle ARIMA. Pour les cycles de production nouveaux et invisibles, le modèle ARIMA prédit les valeurs des paramètres pour le reste du cycle de production et ces valeurs sont transmises au modèle supervisé pour être classifiées.

Résultats : Le modèle de réseau neuronal profond était plus efficace pour modéliser les données. Cependant, les occurrences des cycles de mauvaise qualité sont faibles par rapport aux cycles de bonne qualité. Par conséquent, le modèle continue d'apprendre activement avec les données nouvellement arrivées.

Observations : Cette étude contribue à améliorer les performances globales du processus de production et à assurer des résultats de qualité supérieure par l'utilisation ultérieure de pronostics et de prévisions.

Étude de B Ghotra et al. (2015) [18]

Contexte, algorithmes et données

Une étude sur le corpus de la NASA a influencé une grande quantité de littérature de suivi après sa publication, qui est celle de Lessmann [26], ils ont montré que les performances des modèles de prédiction de défauts formés à l'aide de diverses techniques de classification ne varie pas beaucoup, ce qui implique que l'utilisation de n'importe quelle technique devrait être suffisante pour prédire les défauts. De nombreux chercheurs choisissent de ne pas étudier les performances de diverses techniques de classification sur leurs ensembles de données. C'est la motivation de B Ghotra [18] qui revisitent cette découverte très percutante, en testant différents classificateurs (15 classificateurs) sur différentes bases de données ; la première est la même utilisée par Lessmann [26] , à la fois avant et après le nettoyage, ainsi qu'une autre d'un corpus de systèmes open source.

Construction du modèle et approche d'évaluation.

Le schéma suivant représente l'approche suivie par [18]

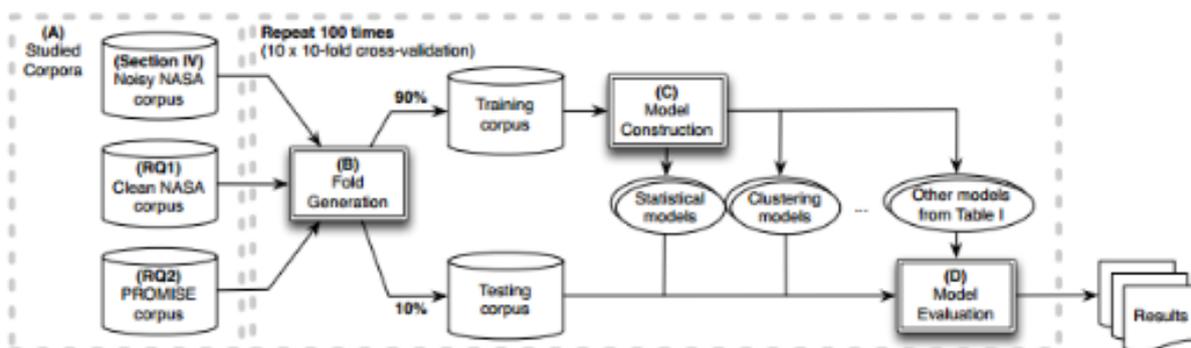


FIG. 2.11 : Construction du modèle et approche d'évaluation

Résultats La réplique de Lessmann et al. donne les mêmes résultats même en utilisant une technique de classement statistique plus avancé (c'est-à-dire le test de Scott-Knott). Cependant, lors de l'utilisation d'une version nettoyée du corpus de la NASA et lors de

l'utilisation d'un autre corpus (l'ensemble de données PROMISE, qui fournit des métriques différentes de celles du corpus de la NASA), les résultats se diffèrent de ceux de Lessmann et al. , c'est-à-dire que les techniques de classification produisent des modèles de prédiction de défauts avec des performances significativement différentes.

Observation : Il existe des ensembles de données où il existe des différences statistiquement significatives entre les performances des modèles formés à l'aide de diverses techniques de classification.

Étude de D Kusumaningrum, et al. (2021) [24]

Objectif de l'étude et résultats obtenus : Contexte, données et algorithmes : Le but de cette étude est de construire un modèle de diagnostic et de pronostic avec le réglage des paramètres d'apprentissage automatique optimaux dans SVM et la RF pour la classification des conditions d'équipement. De plus, la comparaison des paramètres et des méthodes d'apprentissage automatique est effectuée pour déterminer quel modèle a la plus grande précision. Des données réelles d'un moteur EPFAN ont été utilisées pour mener cette étude.

La méthode proposée :

Le schéma suivant représente la méthode proposée par [24].

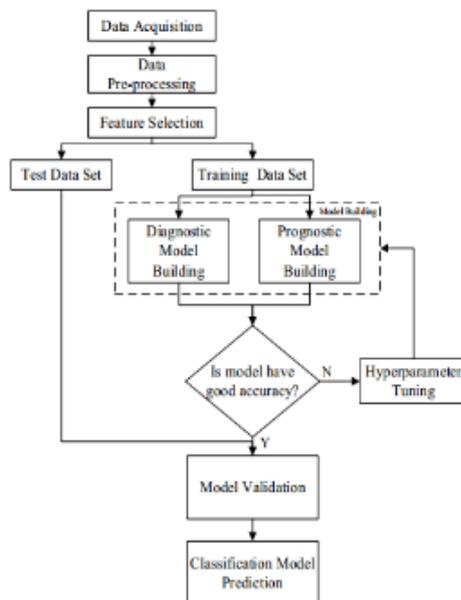


FIG. 2.12 : La méthode proposée

Résultats : Chaque algorithme a une bonne précision pour le modèle diagnostique et pronostique. Mais, la précision et le rappel les plus élevés ont été obtenus avec le RF. Les résultats de la recherche montrent que le paramètre de réglage ne diffère pas de manière significative dans la précision du modèle, en particulier sur l'algorithme RF pour le

pronostic et le diagnostic.

Observation : Les résultats obtenus par cette étude sont spécifiques à la base de données utilisée, ce qui signifie qu'ils ne peuvent pas être généralisés à d'autres contextes ou ensembles de données. L'étude ne prend pas en compte les étapes essentielles de la maintenance prédictive, telles que l'objectif précis de l'étude et la comparaison des résultats en fonction des besoins spécifiques de la maintenance.

2.5.2 Étude comparative et première sélection des algorithmes

Dans cette section, nous présenterons les résultats tirés de notre étude des articles de recherche portant sur les algorithmes d'apprentissage supervisé utilisés en maintenance prédictive (voir 2.2). Au cours de cette analyse, nous avons extrait les critères essentiels permettant d'évaluer ces algorithmes. Nous aborderons également le choix de ces critères de comparaison, en fournissant des arguments justifiant leur pertinence. Enfin, nous présenterons une table comparative qui synthétise les performances des algorithmes évalués en fonction des critères identifiés.

reference	Accuracy in general	Speed of training	Speed of prediction	Tolerance to missing values	Tolerance to irrelevant attributes	Tolerance to redundant attributes	Tolerance to correlated features	Dealing with discrete/continuous	Tolerance to noise	Dealing with danger of overfitting	Attempts for incremental learning	Explanation ability	Support for multi-classification	Statistics Kappa	Type of algorithm	Parametrization	Memory size	Data type	Flexibility	Data size	scalability	Robustness	Implementation
[22]	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
[24]	*		*																				
[31]	*	*	*											*									
[36]		*	*					*		*			*		*	*	*	*	*	*	*	*	*
[28]	*	*	*	*	*	*	*					*								*	*	*	*
[11]	*	*	*	*	*	*	*		*				*							*	*	*	*
[32]	*		*										*										
[21]	*		*																				
[9]	*	*	*	*	*	*	*	*	*	*	*	*	*										
[10]	*	*	*	*	*	*			*	*								*				*	*
[5]		*	*						*					*									
[8]	*		*																	*			
[39]	*	*	*																				
[30]	*	*	*	*	*	*	*		*	*				*	*			*		*	*	*	*
[13]	*	*	*																				
[6]	*	*	*	*	*	*	*								*						*	*	*
[20]		*	*							*					*	*	*						

TAB. 2.2 : A summary of the state of the art in ML applied to PdM

Choix de critères de comparaison : définitions et arguments

Les différents critères de comparaison qui doivent être choisis à l'aide de la recherche scientifique sont maintenant discutés après la description des algorithmes ML, ainsi que de leurs avantages et inconvénients. Le graphique ci-dessous représente les critères sélectionnés :

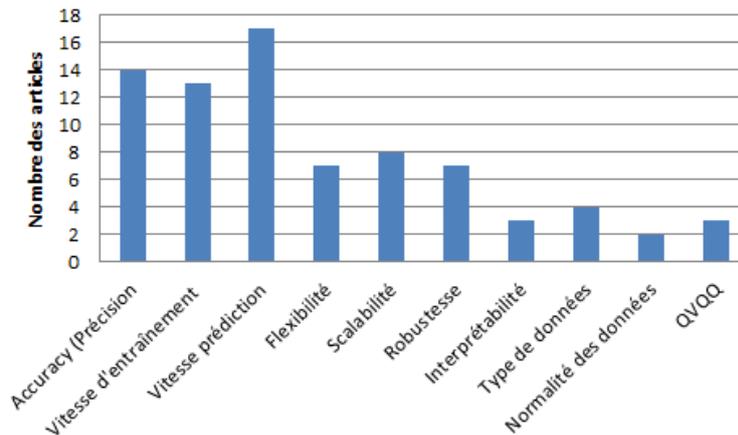


FIG. 2.13 : Nombre de papier concernes par chaque critère

1. Critères liés à l'algorithme :

- **Scalabilité** : C'est la capacité de l'algorithme à maintenir de bonnes performances avec de grandes quantités de données. Dans le contexte de la nouvelle révolution industrielle, où de grandes quantités de données sont constamment générées, la scalabilité devient un avantage clé en faveur de son importance.
- **Flexibilité et adaptabilité** : c'est la capacité à gérer des machines complexes et des schémas de défaillance spécifiques.
- **Robustesse** : C'est la capacité à gérer des conditions inattendues et des données dégradées, telles que la présence de valeurs manquantes causées par des problèmes de défaillance des capteurs.

2. Critères liés aux résultats des algorithmes :

- **Précision** : c'est la capacité de fournir des résultats précis pour éviter toute action inutile ou des pannes imprévisibles.
- **L'interprétabilité** : L'interprétabilité d'un modèle se réfère à sa capacité à être compris et expliqué de manière claire. Cela implique que l'équipe de maintenance peut comprendre les raisons derrière les prédictions générées par le modèle, ce qui leur permet ensuite de développer des stratégies de maintenance appropriées en fonction de cette compréhension.
- **La rapidité** : La rapidité du temps de formation et d'apprentissage est essentielle pour permettre une planification et une exécution rapides des tâches de maintenance.

3. **Critères liés aux caractéristiques des données** : Le choix de l'algorithme repose sur les caractéristiques spécifiques des données, car chaque algorithme atteint des performances optimales dans des conditions particulières. Ainsi, la sélection de l'algorithme le plus adapté est guidée par la nature des données disponibles. Dans ce processus de sélection, il est essentiel de tenir compte des aspects suivants liés aux données :

- **Type de données** : structurées ou non structurées.
- **Quantité de variables qualitatives et quantitatives** : Les variables qualitatives représentent des attributs catégoriques, tandis que les variables quantitatives représentent des mesures numériques.
- **Normalité des données** : La normalité des données fait référence à la distribution des valeurs dans un ensemble de données.

Table comparative

Dans cette section, nous vous présenterons une table comparative qui résume les critères que nous avons choisis pour évaluer différents algorithmes de machine learning utilisés en maintenance prédictive. Nous avons soigneusement sélectionné ces critères en tenant compte de leur pertinence pour la maintenance prédictive.

Critère	RF	SVM	ANN	KNN	DT
Accuracy (Précision)	Excellent (4)	Excellent(4)	Excellent(4)	T.Bien (3)	Bien(2)
Vitesse d'entraînement	T.Bien (3)	Moyenne(1)	Moyenne(1)	T.Bien (3)	T.Bien (3)
Vitesse prédiction	Excellent(4)	Excellent(4)	Excellent(4)	Bien(2)	T.Bien (3)
Flexibilité	T.Bien (3)	T.Bien (3)	Bien(2)	Excellent(4)	Excellent(4)
Scalabilité	Excellent(4)	Bien(2)	Excellent(4)	Bien(2)	Moyenne(1)
Robustesse	Excellent(4)	Excellent(4)	Bien(2)	Bien(2)	Moyenne(1)
Interprétabilité	T.Bien (3)	T.Bien (3)	Moyenne(1)	T.Bien (3)	Excellent(4)
Type de données	Les deux types de données (4)	Structurées (2)	Les deux types de données (4)	Les deux types de données (4)	Les deux types de données (4)
Normalité des données	Non requis (4)	Non requis (4)	requis (2)	Non requis (4)	Non requis (4)
Quantité de variables qualitatives et quantitatives	qualitatives et quantitatives (4)	variables quantitative (2)	qualitatives et quantitatives (4)	qualitatives et quantitatives (4)	variables quantitative (2)
Note	37	29	31	31	28

TAB. 2.3 : Evaluation des algorithmes selon les critères sélectionnés.

Les résultats de la première sélection mettent en évidence que les algorithmes RF, ANN et KNN se démarquent comme les meilleurs candidats parmi ceux que nous avons étudiés.

2.5.3 Discussion

L'analyse approfondie de la revue de littérature a mis en évidence plusieurs problèmes pertinents dans le domaine de ML appliqué à la PdM. Parmi ces problématiques, la méthode de sélection des algorithmes est d'une importance cruciale. Une sélection basée

uniquement sur la table comparative ne tient souvent pas compte des objectifs spécifiques de l'entreprise. Cela signifie qu'il est impossible de considérer ces résultats de manière absolue. De plus, les évaluations de chaque algorithme sont relatives. Par exemple, le KNN peut obtenir d'excellents résultats en termes de temps de formation selon plusieurs études comparatives, mais il peut devenir extrêmement lent avec de grandes bases de données. Par conséquent, se fier uniquement à la table comparative présentée dans la section (2.5.2) pour la sélection est insuffisant.

Il est également essentiel de souligner que les résultats obtenus dans une étude spécifique sont spécifiques à la base de données utilisée et ne peuvent pas être généralisés à d'autres contextes ou ensembles de données. Cela met en évidence l'importance de mener des expérimentations sur différents jeux de données pour obtenir des résultats plus robustes et généralisables. Il convient de mettre l'accent sur la compréhension des conditions contextuelles dans lesquelles une méthode spécifique peut surpasser les autres dans un problème d'application donné.

En outre, il y a un manque de considération du coût de la maintenance. L'analyse de la PdM ne se limite pas seulement à la prédiction des défaillances, mais doit également prendre en compte les coûts associés aux différentes actions de maintenance. Il est donc primordial de développer des approches qui intègrent le coût de la maintenance lors de la planification et de l'exécution des actions prédictives.

Il est évident que la plupart des recherches existantes se sont concentrées sur l'application du processus standard de ML présenté dans la figure (2.4), qui peut ne pas tenir compte des spécificités des problèmes de PdM susceptibles d'influencer la qualité des résultats de prédiction. Par conséquent, il est nécessaire de développer une méthodologie pratique pour l'application de ML dans le domaine de la PdM, en prenant en compte les particularités des données et des objectifs propres à cette discipline.

2.6 Conclusion

Ce chapitre nous a permis d'explorer en profondeur l'état actuel de ML dans le contexte de la PdM. Nous avons examiné les concepts fondamentaux de ML, les différents types d'algorithmes, ainsi que les travaux de recherche existants qui ont abordé l'application de ML à la PdM, en nous concentrant sur les études comparatives des algorithmes. À partir de cette analyse, nous avons identifié plusieurs problématiques importantes dans ce domaine. Notre objectif est de combler ces lacunes en proposant une méthodologie pratique et adaptée, prenant en compte les spécificités des données et des objectifs propres à la PdM. Dans les chapitres suivants, nous détaillerons notre approche proposée, présenterons les résultats obtenus et discuterons des implications pratiques de maintenance prédictive.

Chapitre 3

Modélisation de processus de PdM basé sur la ML

3.1 Introduction

La PdM est cruciale pour réduire les coûts et améliorer la disponibilité des équipements industriels. L'utilisation de la ML est devenue courante dans ce domaine pour prédire les défaillances et planifier les activités de maintenance de manière proactive. Cependant, la plupart des études existantes se concentrent sur le processus standard de ML et négligent les spécificités propres à la PdM.

Ce chapitre se concentre sur la modélisation d'un processus de PdM basé sur la ML en suivant la démarche DMAIC. Nous évaluons les exigences de PdM basées sur la ML, en identifiant les lacunes à améliorer. En analysant les données, nous identifions les mesures clés qui influencent l'efficacité des prédictions en PdM et les avantages potentiels d'un processus spécialisé basé sur la ML. En présentant des méthodes spécifiques et en sélectionnant les algorithmes appropriés, nous visons à améliorer le processus de PdM. Nous soulignons également l'importance de l'évaluation régulière et de l'ajustement continu du processus en fonction des retours d'expérience.

3.2 Démarche DMAIC : Modélisation de notre processus

La méthodologie DMAIC (Define, Measure, Analyze, Improve, Control) est largement reconnue dans le domaine de l'amélioration des processus. Elle offre un cadre structuré pour résoudre les problèmes et optimiser les performances opérationnelles. Dans notre contexte, nous allons appliquer cette approche à l'amélioration du processus standard de ML et l'adapter aux spécifications de la PdM. Le diagramme ci-dessous présente notre démarche DMAIC, spécifiquement conçue pour répondre à notre problématique :

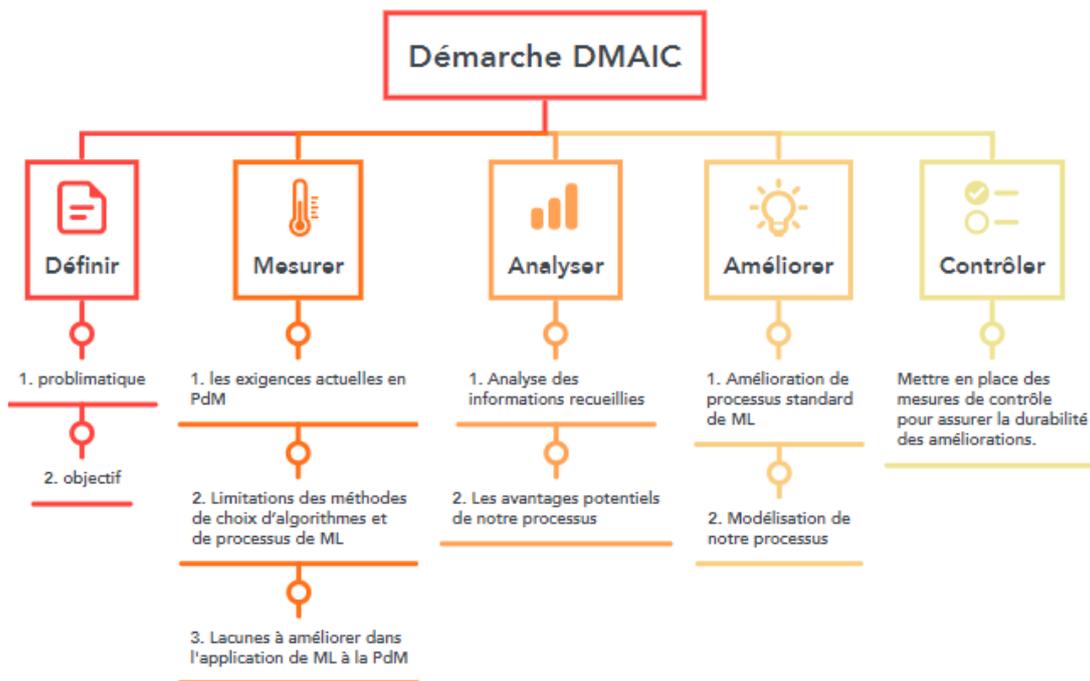


FIG. 3.1 : Les phases de la démarche DMAIC

3.3 Définir

3.3.1 Problématique

L'étude comparative réalisée dans la section (2.5.2) n'est pas suffisante pour prendre une décision sur l'algorithme à utiliser. Par conséquent, il est nécessaire de proposer une autre méthode de sélection dans le but d'évaluer correctement les performances des modèles de ML appliqués à la PdM et obtenir des résultats précis. De plus, la PdM requiert un processus spécifique afin d'appliquer efficacement la ML et obtenir des résultats précis. Le processus standard de ML néglige des aspects critiques tels que la définition précise des objectifs et le choix des mesures d'évaluation appropriées.

3.3.2 Objectif

Les objectifs de cette étude sont les suivants :

- Proposer une méthode efficace permettant de choisir le meilleur algorithme en fonction des spécifications du problème de maintenance.
- Établir un processus pour l'application de la ML dans le domaine de la PdM

3.4 Mesurer

Dans le chapitre précédent, nous avons identifié la nécessité de développer un processus d'application du ML dans la PdM. À cette étape, nous procéderons à une collecte d'informations supplémentaires pour mieux comprendre les lacunes et les besoins en matière de maintenance, en vue de les intégrer à notre processus. Les résultats de cette collecte d'informations sont présentés dans le tableau ci-dessous, mettant en évidence les aspects à prendre en considération lors de l'application de la ML dans la PdM, tels qu'abordés dans les articles étudiés.

Article	Criticité	Objectifs	Stratégie	Algorithmes	Coût	Processus ML	Approche ML
[12]	*			*			*
[40]		*	*		*		*
[11]				*		*	*
[35]		*	*	*	*		
[43]		*	*				
[7]			*				

TAB. 3.1 : Collecte d'informations supplémentaires

Afin de faciliter l'analyse et la compréhension des données présentées dans ce tableau, les résultats sont visualisés de manière graphique à l'aide d'un diagramme en toile d'araignée. Ce type de représentation permet de mettre en évidence les différents aspects

évalués, avec chaque axe du diagramme représentant un aspect spécifique. Les valeurs correspondantes aux différents aspects sont ensuite reliés entre eux pour former une figure en forme de toile d'araignée, ce qui facilite la comparaison des niveaux de considération accordés par les auteurs aux différents aspects étudiés.

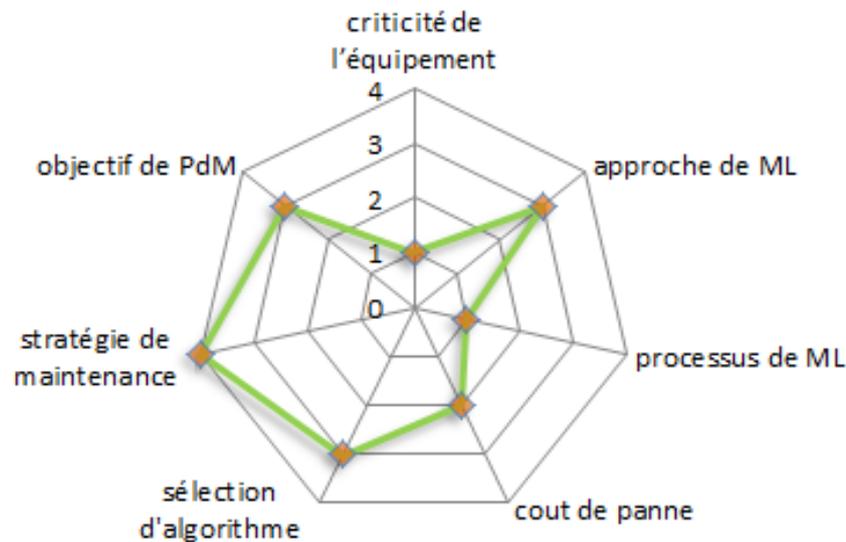


FIG. 3.2 : Diagramme en toile d'araignée

Parmi ces aspects, la stratégie de maintenance et la sélection d'algorithmes de ML ont été largement étudiées et soulignent l'importance accordée à la recherche de méthodes efficaces pour la gestion et l'optimisation de la maintenance des équipements. L'intérêt croissant pour les approches basées sur le ML suggère que ces techniques sont prometteuses pour améliorer les programmes de PdM. Cependant, d'autres aspects tels que la criticité de l'équipement, le coût de panne et le processus de ML ont été moins explorés, mais présentent un potentiel de recherche qui mérite une attention particulière. Dans les sections suivantes, nous examinerons en détail ces différents aspects afin de mieux comprendre leur importance et d'identifier les lacunes à améliorer dans l'application de ML à la PdM.

3.4.1 Amélioration de la compréhension des exigences actuelles en PdM

Les coûts de pannes

L'analyse des coûts de panne passés et de l'impact financier des arrêts non planifiés permet aux entreprises de prendre des décisions éclairées concernant les investissements visant à améliorer la fiabilité des équipements, optimiser les processus de maintenance et réduire les coûts de panne à long terme. En ce qui concerne les politiques de PdM, les éléments de coûts couramment utilisés dans les systèmes de production sont présentés en détail dans la figure suivante.

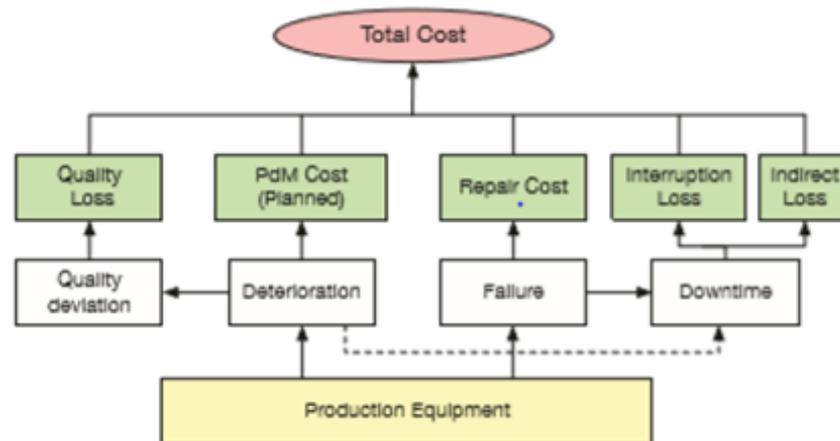


FIG. 3.3 : Les éléments de coûts couramment utilisés dans les systèmes de production pour optimiser les programmes de PdM [35]

La stratégie de maintenance existante

Lors de l'application de la PdM basée sur ML, la nature de la stratégie de maintenance existante joue un rôle crucial [33],[41]. L'apprentissage supervisé est approprié pour les politiques de maintenance corrective (R2F) où les données de maintenance sont disponibles, tandis que les approches d'apprentissage non supervisé sont utilisées pour les politiques de maintenance préventive (PvM) où le cycle complet de maintenance n'est pas toujours visible.

La criticité de l'équipement

L'article [41] a souligné l'importance d'évaluer la criticité de l'équipement dans son contexte opérationnel avant de mettre en œuvre la PdM. Si la défaillance de l'équipement n'a pas d'impact sur le fonctionnement de l'usine, il peut être envisageable de le laisser fonctionner jusqu'à sa panne, puis de le réparer. Cependant, si un équipement est crucial, il peut être nécessaire d'avoir des matériels de secours pour maintenir le fonctionnement de l'usine pendant que l'unité problématique est réparée.

Les métriques d'évaluation

Il est important de choisir les métriques d'évaluation en fonction des objectifs spécifiques de la maintenance prédictive et des contraintes du problème. Certaines métriques peuvent être plus pertinentes que d'autres en fonction du coût des erreurs, du déséquilibre de classes et des priorités du problème de maintenance. La définition détaillée de ces métriques se trouve en annexe A.2.

3.4.2 Limitations des méthodes de choix d'algorithmes et de processus de ML

Les méthodes de sélection d'algorithmes proposées par différents auteurs sont relatives et ne permettent pas une application directe. De plus, les études de cas réalisées par ces auteurs ont utilisé des processus standard qui n'ont pas pris en compte les spécifications de maintenance, notamment l'objectif de l'application de ce type de maintenance, qui permet de faire un choix approprié des métriques d'évaluation pour évaluer équitablement la performance des modèles. Cette négligence peut avoir un impact sur la qualité des résultats obtenus.

3.4.3 Lacunes à améliorer dans l'application de ML à la PdM

1. La minimisation des coûts d'exploitation n'est pas suffisamment prise en compte.
2. La nature de la stratégie de maintenance existante n'est pas toujours considérée de manière adéquate.
3. L'évaluation de la criticité de l'équipement est souvent négligée.
4. Les comparaisons entre les algorithmes de ML sont relatives.
5. Les méthodes de sélection d'algorithmes négligent les spécificités du problème.
6. Les processus utilisés ne sont pas adaptés aux spécifications de maintenance.

3.5 Analyser

3.5.1 Analyse des informations recueillies

Le choix de la stratégie de maintenance dépend de la criticité de l'équipement, où les équipements critiques nécessitent une attention particulière. De plus, le choix de l'approche de ML dépend des données disponibles, qui sont influencées par la stratégie de l'entreprise. En outre, la détermination de l'objectif de la maintenance est essentielle pour faire le bon choix des métriques d'évaluation des algorithmes de ML appropriées.

Il est important de souligner que l'accuracy est utilisée comme une métrique supplémentaire pour évaluer la performance globale du modèle, car elle donne une mesure générale de l'exactitude des prédictions. Cependant, dans le domaine de la maintenance prédictive, il est recommandé de se concentrer sur des métriques plus adaptées à chaque problème spécifique ; le rappel (sensibilité), la précision et la spécificité. En se basant sur ces métriques, on peut prendre en compte les aspects spécifiques du problème de maintenance et obtenir une évaluation plus précise de la performance du modèle.

En outre, pour résoudre le problème de relativité des méthodes d'évaluation, nous proposons une méthode de choix d'algorithmes basée sur le vote pondéré. Cette méthode

permet d'attribuer des poids aux algorithmes en fonction des exigences spécifiques de chaque entreprise, telles que les caractéristiques des données disponibles. Cela garantit une sélection d'algorithmes adaptée aux besoins de l'entreprise.

En intégrant ces idées dans le processus de PdM basé sur le ML, on peut bénéficier d'une meilleure spécialisation, d'une évaluation plus précise, d'une optimisation des coûts et d'une sélection appropriée des algorithmes, ce qui contribue à améliorer l'efficacité et la fiabilité de la maintenance.

3.5.2 Les avantages potentiels de notre processus

1. Amélioration de la précision en faisant le bon choix de classificateur.
2. Réduction des coûts en choisissant la stratégie adéquate de maintenance.
3. Optimisation des ressources en évitant les interventions inutiles.
4. Prise de décisions basée sur les données.
5. Adaptabilité aux changements (processus flexibles)

3.6 Améliorer

3.6.1 Amélioration de processus standard de ML

Nous avons présenté précédemment (voir section 2.4) le processus standard de ML qui est utilisé dans le cadre de la PdM. Cependant, nous avons identifié certaines lacunes de ce processus (section 3.4.3) qui nécessitent d'être abordées. C'est pourquoi nous allons maintenant présenter quatre méthodes pour améliorer ce processus.

Méthode de choix de la stratégie de maintenance

Étant donné que les solutions basées sur la ML pour la mise en œuvre de la PdM sont généralement coûteuses en raison de l'exigence de matériel spécifique pour la collecte et le stockage des big data, il est primordial de prendre des décisions éclairées concernant le type de maintenance à appliquer. En fonction de l'analyse des équipements, en prenant en compte leur criticité ainsi que le coût potentiel engendré par une panne, nous sommes en mesure de choisir la stratégie de maintenance appropriée. Le schéma ci-dessous illustre comment ce choix est effectué.

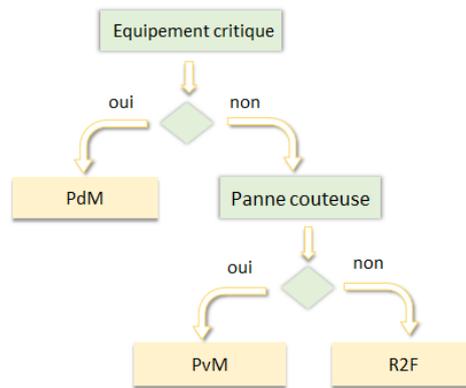


FIG. 3.4 : Processus de sélection de la stratégie de maintenance

Méthode de choix de l'approche de ML

Comme détaillé dans notre analyse 3.5, l'approche de ML à utiliser dépend des données dont nous disposons et le type de données dépend de la stratégie existante. La décision concernant l'approche à adopter est illustrée dans le schéma suivant :

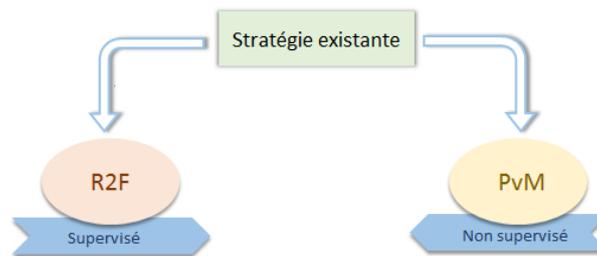


FIG. 3.5 : Choix de l'approche de ML

Méthode de choix des métriques d'évaluation

Selon les besoins de maintenance et selon le coût de l'erreur et la criticité de l'équipement on va faire notre choix de métriques d'évaluation.

Exemple : si l'équipement est critique et que le coût de panne est très élevé par rapport à une réparation suite à une fausse alerte, dans ce cas, notre but est de prédire avec précision les défaillances en priorité et éviter les fausses alertes en deuxième position.

La table suivante va nous aider à faire notre choix de mesure d'évaluation adéquate à notre cas d'étude :

Méthode de vote pondéré pour le choix d'algorithmes (deuxième sélection)

Pour résoudre le problème de relativité des tables comparatives des différents algorithmes, nous proposons d'appliquer la méthode du vote pondéré sur ces tables. L'intérêt de cette méthode réside dans sa capacité à spécialiser la table comparative en fonction

Métriques d'évaluation	Besoins de maintenance
Précision	Fiabiliser les alertes de pannes générées par le modèle. Minimiser le risque de déclencher des interventions inutiles
Sensibilité	Eviter de manquer les alarmes de pannes
Spécificité	Limiter les interventions inutiles en classant correctement les états de bon fonctionnement
F1-score	Indique une performance globale optimale du modèle en termes de prédiction précise des pannes tout en limitant les erreurs de classification (Harmonisation entre précision et rappel)
Courbe ROC	Capacité de discrimination entre les états de pannes et entre les états de bon fonctionnement. Choisir ou ajuster le seuil de classification en fonction de nos préférences en matière de pannes ou de bon fonctionnement en fonction des coûts associés à chaque erreur de classification

TAB. 3.2 : Métriques d'évaluation et besoins de maintenance

des spécificités du problème étudié, en attribuant un poids élevé aux critères ou caractéristiques qui contribuent le plus à la résolution du problème.

Voici un exemple d'application de cette méthode où des poids élevés ont été attribués aux critères de robustesse et de précision.

Poids	9	6	7	5	8	10	4	3	4	4	Note pondérée
Critère	Précision	T.apprentissage	T.prédiction	Flexibilité	Scalabilité	Robustesse	Interpretabilité	Normalité données	Type données	QVQQ	
RF	4	3	4	4	4	4	3	4	4	4	230
SVM	3	2	3	2	2	4	3	4	2	2	166
KNN	3	3	2	2	2	2	3	4	4	4	145
ANN	4	2	2	4	4	2	1	2	4	4	150
DT	2	3	3	3	1	1	4	4	4	2	142

TAB. 3.3 : Exemple d'un tableau pondéré

D'après le tableau 3.3, les trois premiers algorithmes ayant les notes pondérées les plus élevées sont RF, SVM et ANN.

3.6.2 Modélisation de notre processus

Dans le processus que nous proposons, nous commençons tout d'abord par l'identification de la problématique et la définition des objectifs. Nous procédons également à l'évaluation de la criticité de l'équipement. Si l'équipement est considéré comme critique, nous appliquons directement la PdM. Dans le cas contraire, nous estimons le coût de la panne. Si le coût de la panne est élevé, nous mettons en œuvre la PvM. Sinon, nous appliquons le R2F.

Une fois ces étapes effectuées, nous nous appuyons sur la PdM pour poursuivre notre processus. La première étape consiste à sélectionner les métriques d'évaluation en fonction des besoins spécifiques de maintenance, puis à collecter et traiter les données nécessaires. Ensuite, nous procédons à la sélection, à l'entraînement et à l'évaluation de l'algorithme approprié. Enfin, nous sommes en mesure de prédire les résultats escomptés. Il est important de souligner qu'il existe une rétroaction entre les phases d'évaluation, de sélection d'algorithme et de choix des métriques, ce qui permet d'améliorer la qualité des résultats si ces derniers ne sont pas satisfaisants. La figure 3.6 présente la modélisation de notre processus.

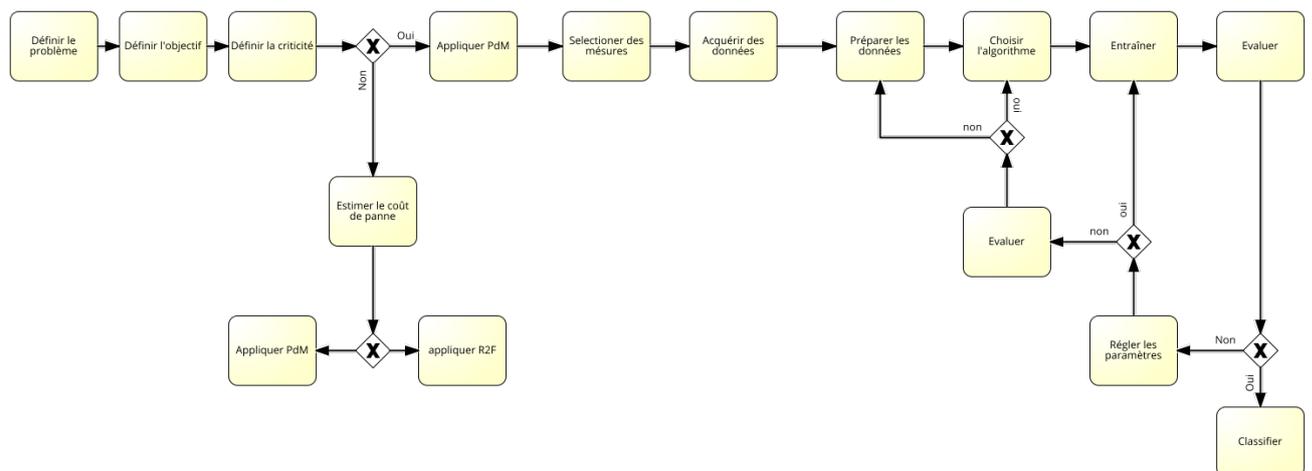


FIG. 3.6 : Processus de ML pour la PdM

3.7 Contrôler

Pour garantir le bon fonctionnement de notre processus, il est essentiel de mettre en place des mécanismes de suivi et de contrôle réguliers. Cela implique d'évaluer régulièrement l'efficacité et l'adéquation du processus en utilisant des métriques pertinentes, telles que la précision des prédictions et la réduction des temps d'arrêt. De plus, il est important d'établir des procédures pour ajuster et améliorer le processus en fonction des retours d'expérience, des évolutions du domaine et des besoins changeants. Ces ajustements peuvent inclure des mises à jour des modèles, des améliorations du prétraitement des données ou des modifications des algorithmes utilisés. En impliquant activement les

parties prenantes tout au long du processus, en recueillant leurs commentaires et en les intégrant dans les décisions prises, on assure une meilleure adhésion et une meilleure acceptation du processus de ML appliqué à la maintenance prédictive.

3.8 Conclusion

En conclusion de ce chapitre, nous avons présenté la modélisation d'un processus de PdM basé sur la ML. En suivant la démarche DMAIC, nous avons défini la problématique de PdM et l'objectif de notre étude. Nous avons mesuré les besoins actuels en PdM, évalué les méthodes de ML existantes, et identifié les lacunes à améliorer dans leur application. En analysant les données recueillies, nous avons identifié les mesures clés influençant l'efficacité des prédictions en PdM, ainsi que les avantages potentiels d'un processus spécialisé de PdM basé sur le ML. En améliorant le processus, nous avons développé des méthodes spécifiques et réalisé une étude comparative pour sélectionner les algorithmes appropriés. Enfin, nous avons souligné l'importance d'évaluer régulièrement l'efficacité du processus et d'ajuster et d'améliorer continuellement en fonction des retours d'expérience. Dans le prochain chapitre, nous démontrerons l'efficacité de notre processus à travers un exemple basé sur des données réelles téléchargées à partir de Kaggle et un problème réel de maintenance d'un compresseur d'air, mettant en évidence les résultats obtenus grâce à notre approche spécialisée de PdM basée sur le ML.

Chapitre 4

Tests et Résultats

4.1 Introduction

Après avoir présenté notre processus de PdM basé sur la ML, ce chapitre se propose de le mettre à l'épreuve en utilisant deux bases de données distinctes. La première a été recueillie lors de notre stage chez Condor, portant sur un compresseur d'air, tandis que la seconde a été téléchargée depuis Kaggle. Dans le but de tester notre processus, nous débutons par une définition approfondie du problème et des objectifs associés, afin de choisir les métriques d'évaluation appropriées.

Ensuite, nous abordons la phase de prétraitement des données, visant à obtenir des données propres et équilibrées. Cette étape est réalisée à l'aide de l'outil Weka. Ensuite, nous procédons à la sélection des algorithmes à l'aide de la méthode de vote pondérée, intégrée à notre processus.

Par la suite, nous nous concentrons sur la construction des modèles, mettant en évidence l'importance de l'optimisation des paramètres. Grâce aux métriques choisies précédemment et à l'utilisation de matrices de confusion, nous démontrons la validité de notre processus. Nous évaluons ainsi les modèles initialement sélectionnés, qui ont été éliminés lors de la deuxième phase de sélection.

4.2 La méthodologie expérimentale

Après avoir établi notre processus dans la phase de modélisation 3.6.2, nous procéderons à son application dans le cadre de deux scénarios ou problèmes de maintenance distincts. Notre processus sera subdivisé en trois catégories principales : tout d'abord, une définition approfondie du problème et des objectifs, suivie de la description, la visualisation, et le traitement des données propres à chaque scénario. Ensuite, nous aborderons l'implémentation des algorithmes en sélectionnant ceux qui répondent aux spécifications du problème, en construisant un modèle, et en optimisant les hyperparamètres. Enfin, nous évaluerons les résultats obtenus et mènerons des discussions pertinentes. Cette approche méthodique permettra de traiter les scénarios de maintenance de manière professionnelle et rigoureuse.

4.3 Description et analyse des scénarios de test

4.3.1 Scénario 1

Description de scénario

Dans ce scénario, nous sommes confrontés à la tâche de résoudre un problème récurrent d'arrêt d'un équipement industriel, entraînant des temps d'arrêt imprévus et des coûts de réparation élevés. Il est important de noter que l'arrêt de l'équipement n'impacte pas directement la chaîne de production, mais sa réparation est généralement coûteuse. Nous disposons de données de surveillance, telles que les vibrations, les températures, les niveaux de bruit et les variations de performance de l'équipement. L'objectif principal

consiste à mettre en place une stratégie de maintenance appropriée pour réduire les temps d'arrêt et les coûts associés.

Analyse du scénario

L'analyse est réalisée en suivant les étapes définies dans notre processus décrit en détail dans le troisième chapitre 2.6. Le schéma ci-dessous illustre visuellement notre approche analytique :



FIG. 4.1 : Analyse de premier scénario

4.3.2 Scénario 2

Description de scénario

L'entreprise Condor est confrontée à la tâche de résoudre un problème de panne d'un compresseur d'air GA 75 VSD, qui est un équipement critique dans sa chaîne de production. La panne de ce compresseur entraîne l'arrêt complet de la chaîne de production, ce qui a des conséquences directes sur les délais de livraison, la productivité et les pertes financières. Pour résoudre ce problème, nous disposons de données, telles que la température et la pression, collectées à partir des capteurs installés sur la machine. L'objectif principal de l'entreprise est de mettre en place une stratégie de maintenance visant à réduire les temps d'arrêt imprévus, à éviter les pannes du compresseur et à minimiser les coûts associés. Des informations détaillées sur le compresseur étudié et l'entreprise sont fournies en annexe 4.8.

Analyse du scénario

Le schéma ci-dessous présente notre approche analytique pour le deuxième scénario :

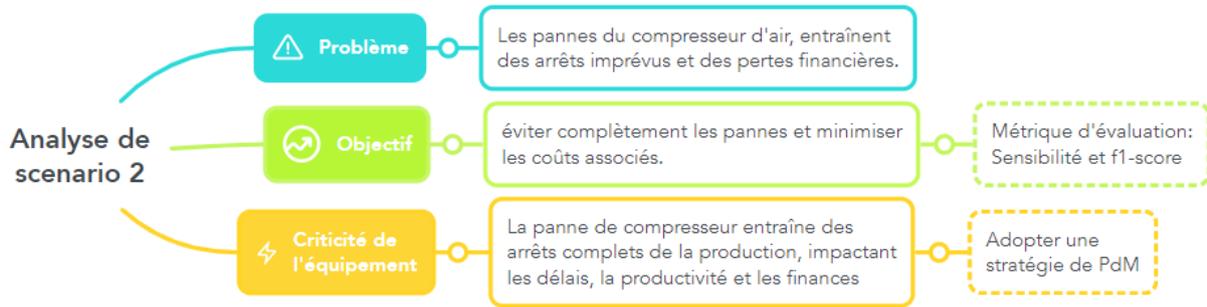


FIG. 4.2 : Analyse du deuxième scénario

4.4 Description des données

Dans cette section, nous allons présenter les données relatives à nos scénarios.

4.4.1 Base de données 1

Les données sont téléchargées à partir du site : <https://www.kaggle.com/datasets/shivamb/machine-predictive-maintenance-classification>, ce jeu de données représente des mesures réelles de maintenance prédictive dans l'industrie, avec des caractéristiques telles que la qualité du produit, les températures de l'air et du processus, la vitesse de rotation, le couple et l'usure de l'outil. La cible de prédiction est la défaillance ou non de l'équipement. Cette base de données comprend 10 000 instances, ce qui signifie qu'il y a 10 000 observations pour chaque caractéristique.

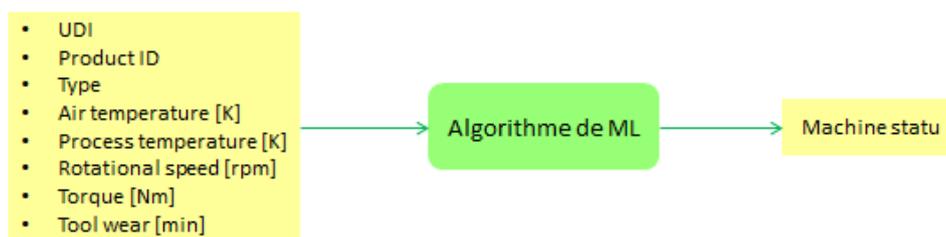


FIG. 4.3 : les données d'entrées et la sortie de premier problème

4.4.2 Base de données 2

Les données utilisées dans cette étude ont été collectées lors du stage effectué à l'entreprise Condor. L'objectif principal était de prédire l'état d'un compresseur d'air, un équipement stratégique dans la chaîne de production. Les données ont été directement collectées à partir de la machine GA 75 VSD, en utilisant des capteurs de température et de pression.

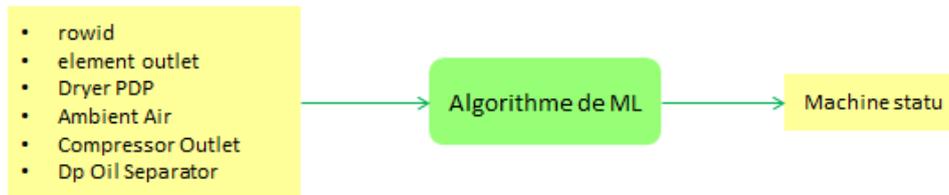


FIG. 4.4 : les données d'entrées et la sortie de deuxième problème

4.5 Préparation des données

La visualisation et la préparation des données sont réalisées à l'aide du logiciel de ML appelé Weka.

4.5.1 Base de données 1

Exploration des données

La fréquence d'apparition dans la première classe est égale à 9652, tandis que dans la deuxième classe, elle est de 348.

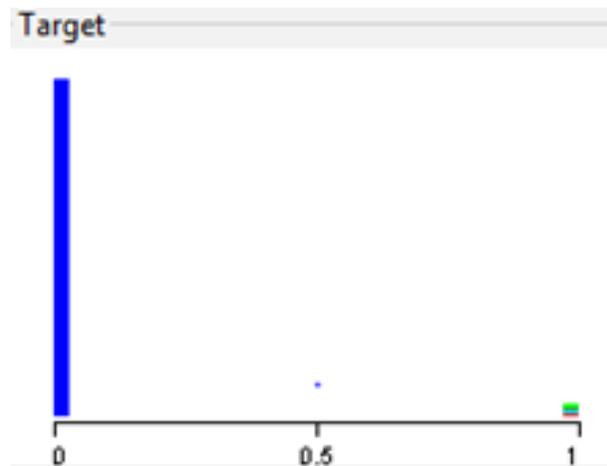


FIG. 4.5 : Visualisation de la répartition des classes de BDD 1

Il existe une corrélation significative entre la variable "Torque" et la variable "Rotational speed", indiquant une forte relation entre ces deux variables. De la même manière, il y a également une corrélation élevée entre la variable "Process temperature" et la variable "Air temperature".

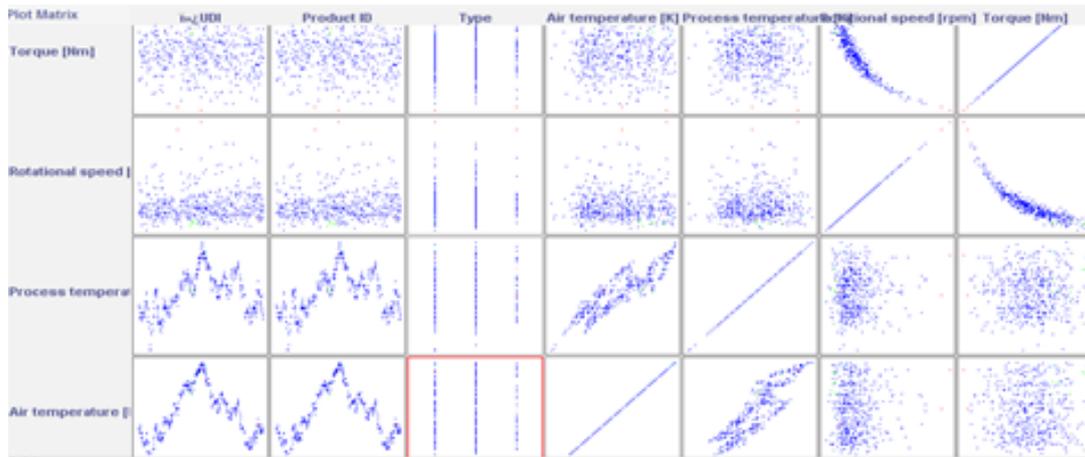


FIG. 4.6 : Analyse de la corrélation entre les variables

Les données ne contiennent pas de valeurs manquantes, ce qui signifie que toutes les observations ont des valeurs complètes pour toutes les variables. Cependant, nous avons identifié la présence d'outliers dans les données. Un graphique représentant l'ensemble des données, composé de 10 000 observations, révèle la présence de 100 outliers. Ces outliers sont des observations qui se situent considérablement en dehors de la plage attendue ou des valeurs typiques pour une variable donnée.

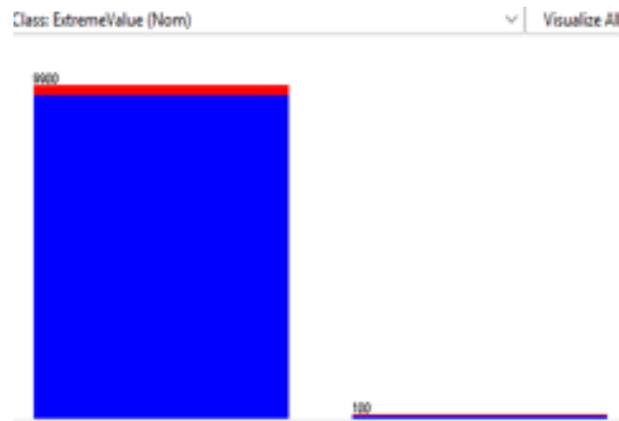


FIG. 4.7 : Visualisation des valeurs aberrantes

Nettoyage des données

Supprimer les valeurs aberrantes ou incohérentes : Après le processus de nettoyage des données, notre jeu de données ne contient plus d'outliers. Sa taille a été réduite à 9900 instances, suite à la suppression de 100 observations considérées comme des outliers.

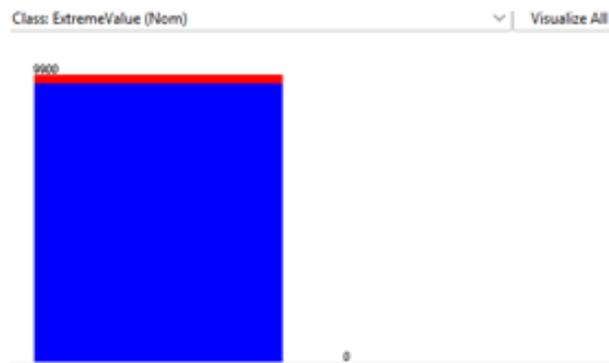


FIG. 4.8 : Suppression des valeurs aberrantes

Sélection des fonctionnalités

Tous les attributs de notre étude sont pertinents, mais ils présentent des degrés d'importance différents.

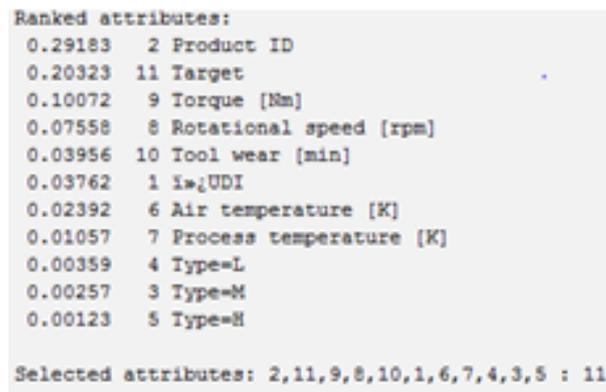


FIG. 4.9 : La sélection des fonctionnalités

Encodage des données

Pour l'attribut "type", qui contient des valeurs catégorielles, nous le transformons en variables binaires afin d'obtenir trois attributs distincts correspondant au nombre de catégories présentes. Pour l'attribut "failure type", nous allons convertir les valeurs catégorielles en valeurs numériques; nous allons attribuer une valeur numérique de 0 à 5.

- 1 i=2UDI
- 2 Product ID
- 3 Type=M
- 4 Type=L
- 5 Type=H
- 6 Air temperature [K]
- 7 Process temperature [K]
- 8 Rotational speed [rpm]
- 9 Torque [Nm]
- 10 Tool wear [min]
- 11 Target

FIG. 4.10 : Encodage des données

Équilibrage des classes

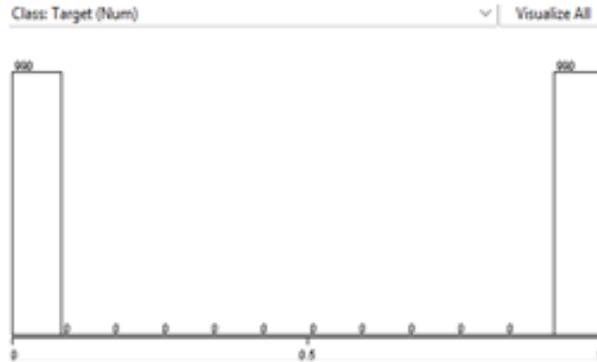


FIG. 4.11 : Équilibrage des classes

4.5.2 Base de données 2

Exploration des données

La fréquence d'apparition dans la première classe est égale à 974, tandis que dans la deuxième classe, elle est de 174.

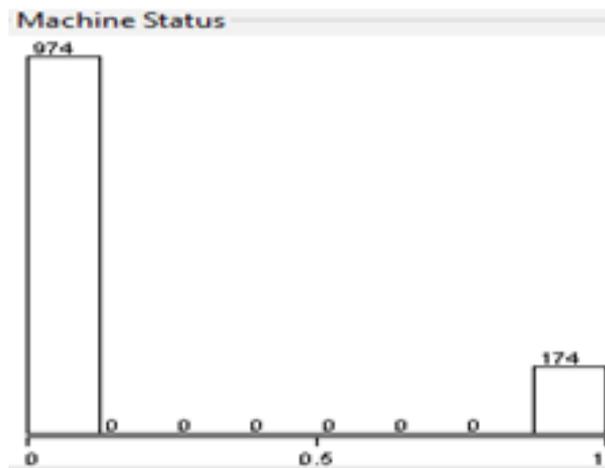


FIG. 4.12 : Visualisation de la répartition des classes de BDD 1

D'après le graphique, on peut observer que les valeurs de panne sont situées à l'extrémité du nuage de points des variables.

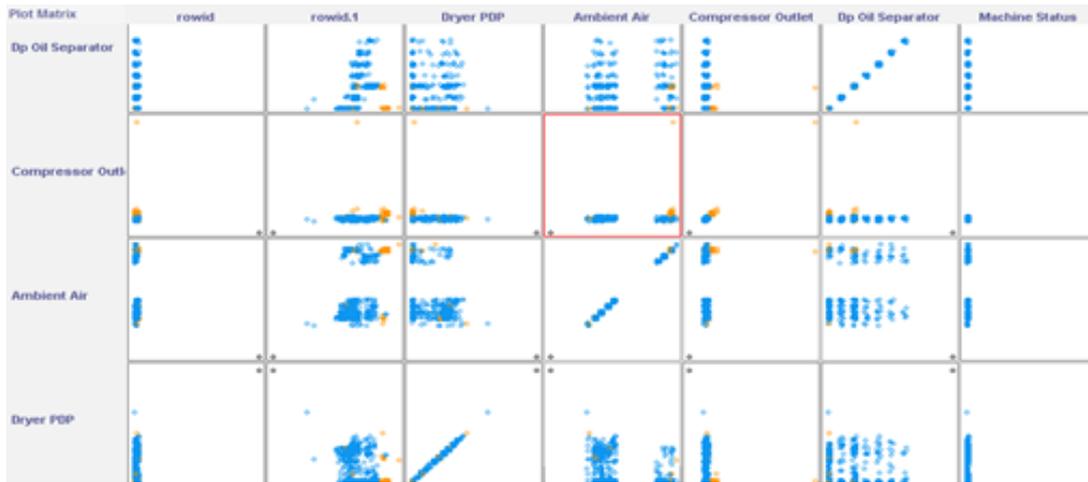


FIG. 4.13 : Visualisation de la corrélation entre les variables

On peut constater qu’il y a cinq valeurs manquantes, dont quatre sont liées à l’attribut ”Dp Oil separator” et une à l’attribut ”Machine status”.

Selected attribute		
Name: Dp Oil Separator	Distinct: 9	Type: Numeric
Missing: 4 (0%)		Unique: 2 (0%)

Selected attribute		
Name: Machine Status	Distinct: 2	Type: Numeric
Missing: 1 (0%)		Unique: 0 (0%)

FIG. 4.14 : Nombre de données manquantes

Nettoyage des données

remplacement des données manquantes

Selected attribute		
Name: rowid	Distinct: 1146	Type: Numeric
Missing: 0 (0%)		Unique: 1145 (100%)

Selected attribute		
Name: rowid	Distinct: 1146	Type: Numeric
Missing: 0 (0%)		Unique: 1145 (100%)

FIG. 4.15 : Remplacement des données manquantes

4.6 Implémentation des algorithmes et résultats

Dans cette partie, nous procéderons à la sélection des algorithmes adaptés à chaque scénario par l’application de la méthode de vote pondérée. Nous présenterons également le code des modèles choisis et, enfin, nous exposerons et analyserons les résultats de classification obtenus.

4.6.1 Sélection d'algorithmes

Compte tenu de l'insuffisance de la première sélection 2.5.2, qui ne tient pas compte des spécificités et de l'objectif du problème, nous allons procéder à une deuxième sélection basée sur la méthode de vote pondéré. Cette approche nous permettra de faire un choix éclairé parmi les différents algorithmes en prenant en compte leurs performances et leur adéquation aux exigences de notre problème.

Algorithmes adaptés au scénario 1

Étant donné les coûts élevés de réparation de l'équipement, nous avons accordé une grande importance à la précision de notre algorithme, ainsi qu'à sa rapidité d'apprentissage et de prédiction. Une meilleure précision nous permet de prendre des décisions plus fiables pour éviter les réparations inutiles. De plus, une vitesse d'apprentissage et de prédiction plus rapide donne à l'équipe de maintenance davantage de temps pour planifier les interventions avant que des pannes ne surviennent.

L'interprétabilité de l'algorithme est également un critère essentiel. Elle permet à l'équipe de maintenance de comprendre les causes profondes des pannes et d'identifier les actions appropriées pour les éviter à l'avenir.

Étant donné que les arrêts de l'équipement sont récurrents et généralement imprévus, la flexibilité de l'algorithme est cruciale. Elle permet à l'algorithme de s'adapter à des scénarios complexes et à des changements dans les données d'entrée, ce qui est essentiel pour anticiper et résoudre les problèmes de manière proactive.

En conclusion, les critères importants dans notre scénario pour éviter les réparations inutiles de l'équipement sont la précision, la rapidité d'apprentissage et de prédiction, l'interprétabilité et la flexibilité.

Le tableau ci-dessous 4.1 présente la sélection effectuée en indiquant les poids attribués à chaque critère

Dans le cas d'étude 1, lors de la deuxième sélection des algorithmes, nous avons sélectionné les algorithmes suivants : RF, KNN et DT
Ces algorithmes ont été choisis en fonction de leurs performances et de leur pertinence pour résoudre le problème spécifique de l'étude.

Algorithmes adaptés au scénario 2

Pour garantir de bonnes performances à mesure que la taille de la base de données augmente, compte tenu du raccordement effectué toutes les 3 secondes, il est essentiel que notre algorithme soit scalable. Étant donné que les données proviennent de capteurs sujets à des défaillances ou des dégradations, la robustesse de l'algorithme est également requise. De plus, en raison de la criticité de l'équipement, une précision élevée est essentielle. En combinant ces trois critères, nous assurons la qualité des résultats de prédiction.

De plus, pour une meilleure planification des opérations de maintenance et une meilleure

Poids	10	10	10	5	1	1	10	1	1	5	Note pondérée
Critère	Précision	T.apprentissage	T.prédiction	Flexibilité	Scalabilité	Robustesse	Interprétabilité	Normalité données	Type données	QVQQ	
RF	4	3	4	4	4	4	3	4	4	4	180
SVM	3	2	3	2	2	4	3	4	2	2	134
KNN	3	3	2	2	2	2	3	4	4	4	136
ANN	4	2	2	4	4	2	1	2	4	4	126
DT	2	3	3	3	1	1	4	4	4	2	147

TAB. 4.1 : Sélection d’algorithmes adaptés au scénario 1

prise de décision, il est recommandé de privilégier des algorithmes qui offrent à la fois une interprétabilité élevée et une rapidité d’exécution.

Il convient de noter que, dans notre cas, les données sont bien structurées et la quantité de variables qualitatives n’est pas très élevée. Par conséquent, tous les algorithmes fonctionneraient bien avec ce type de données.

En conclusion, dans le choix des algorithmes, une importance élevée est accordée à la scalabilité, à la précision, à l’interprétabilité, à la rapidité et à la robustesse.

Le tableau ci-dessous présente la sélection effectuée en indiquant les poids attribués à chaque critère

Poids	10	5	5	1	10	10	5	1	1	1	Note pondérée
Critère	Précision	T.apprentissage	T.prédiction	Flexibilité	Scalabilité	Robustesse	Interprétabilité	Normalité données	Type données	QVQQ	
RF	4	3	4	4	4	4	3	4	4	4	186
SVM	3	2	3	2	2	4	3	4	2	2	134
KNN	3	3	2	2	2	2	3	4	4	4	124
ANN	4	2	2	4	4	2	1	2	4	4	139
DT	2	3	3	3	1	1	4	4	4	2	103

TAB. 4.2 : Sélection d’algorithmes adaptés au scénario 2

Dans le cas d’étude 2, lors de la deuxième sélection des algorithmes, nous avons sélectionné les algorithmes suivants : RF, ANN et SVM.

4.6.2 Construction de modèles

Dans cette section, nous présentons le code correspondant à chacun des modèles sélectionnés précédemment, afin de mieux comprendre leur implémentation et de les utiliser dans notre solution.

- Création du modèle.
- Entraînement du modèle :
 - Nous utilisons la méthode ‘fit()’ pour entraîner le modèle sur les données d’entraînement.
 - Le modèle ajuste les données et apprend à associer les caractéristiques (features) aux étiquettes correspondantes (labels).
- Prédications sur les données de test :
 - Nous utilisons la méthode ‘predict()’ pour faire des prédictions sur les données de test.
 - Le modèle utilise les caractéristiques de ces données pour prédire les étiquettes correspondantes.

En exécutant ce code pour chaque modèle sélectionné, nous pourrions évaluer leurs performances et déterminer lequel est le plus adapté à nos scénarios.

RF

```
# Créer les modèles
random_forest_model = RandomForestClassifier()

# Entraîner les modèles sur les données d'entraînement
random_forest_model.fit(X_train,y_train)

# Faire des prédictions sur l'ensemble de test
rf_predictions = random_forest_model.predict(X_test)
```

FIG. 4.16 : Code de RF

DT

```
# Create the Decision Tree model
decision_tree_model = DecisionTreeClassifier()

# Train the model on the training data
decision_tree_model.fit(X_train, y_train)

# Make predictions on the test set
dt_predictions = decision_tree_model.predict(X_test)
```

FIG. 4.17 : Code de DT

SVM

```
# Créer Les modèles|
svm_model = SVC(random_state=42)

# Entraîner Les modèles sur Les données d'entraînement
svm_model.fit(X_train, y_train)

# Faire des prédictions sur l'ensemble de test
svm_predictions = svm_model.predict(X_test)
```

FIG. 4.18 : Code de SVM

KNN

```
# Create the KNN model
knn_model = KNeighborsClassifier()

# Train the model on the training data
knn_model.fit(X_train, y_train)

# Make predictions on the test set
knn_predictions = knn_model.predict(X_test)
```

FIG. 4.19 : Code de KNN

ANN

```
# Créer Les modèles
ann_model = MLPClassifier(hidden_layer_sizes=(64, 64), activation='relu', random_state=42)

# Entraîner les modèles sur Les données d'entraînement
ann_model.fit(X_train, y_train)
|
# Faire des prédictions sur l'ensemble de test
ann_predictions = ann_model.predict(X_test)
```

FIG. 4.20 : Code de ANN

4.6.3 Optimisation des hyperparamètres

L'optimisation des hyperparamètres est une étape cruciale pour améliorer la qualité des prédictions. Dans cette section, nous présenterons le code et les hyperparamètres de chaque algorithme.

Vous trouverez en annexe B.1.7 les hyperparamètres spécifiques à chaque algorithme, ainsi que la description de la méthode d'optimisation utilisée, à savoir la recherche aléatoire (random search)

Code d'optimisation des hyperparamètres pour les différents algorithmes

Le processus suivant décrit les étapes pour optimiser les hyperparamètres d'un modèle à l'aide de la recherche aléatoire :

1. Définition des hyperparamètres à optimiser : Une grille des hyperparamètres est définie dans le dictionnaire 'param grid'.
2. Création du modèle
3. Recherche aléatoire des meilleurs hyperparamètres : Une recherche aléatoire des meilleurs hyperparamètres est effectuée en utilisant la classe 'RandomizedSearchCV' du module 'sklearn.model_selection'.
4. Obtention des meilleurs hyperparamètres : Les meilleurs hyperparamètres sont obtenus en accédant à l'attribut 'best params' de l'objet 'random search'.
5. Création d'un nouveau modèle avec les meilleurs hyperparamètres
6. Entraînement du modèle : Le nouveau modèle est entraîné sur les données d'entraînement à l'aide de la méthode 'fit()'.
7. Prédiction sur l'ensemble de test : Les prédictions sont effectuées sur l'ensemble de test en utilisant la méthode 'predict()' du modèle entraîné.
8. Calcul des métriques : en utilisant les fonctions correspondantes du module 'sklearn.metrics'.

- **RF**

```
# Définir Les hyperparamètres à optimiser
param_dist = {
    'max_depth': [None, 3, 5],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2]
}
# Créer un classifieur Random Forest
rf = RandomForestClassifier()

# Créer un objet RandomizedSearchCV
random_search = RandomizedSearchCV(
    estimator=rf,
    param_distributions=param_dist,
    n_iter=10,
    cv=5,
    scoring='accuracy',
    random_state=42)

# Entraîner Le modèle avec La recherche aléatoire des hyperparamètres
random_search.fit(X_train, y_train)

# Utiliser le meilleur modèle pour faire des prédictions
best_model = random_search.best_estimator_
y_pred = best_model.predict(X_test)

# Calculer La précision et La spécificité
accuracy = accuracy_score(y_test, y_pred)
tn, fp, fn, tp = confusion_matrix(y_test, y_pred).ravel()
specificity = tn / (tn + fp)
```

FIG. 4.21 : Code d'optimisation des hyperparamètres pour RF

- **DT**

```
# Définir Les hyperparamètres à optimiser
hyperparameters = {
    'criterion': ['gini', 'entropy'],
    'max_depth':[None, 3, 5, 10],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2]}

# Créer Le modèle de L'arbre de décision
model = DecisionTreeClassifier()

# Recherche aléatoire des meilleurs hyperparamètres
random_search = RandomizedSearchCV(model, hyperparameters, n_iter=10, cv=5)
random_search.fit(X_train, y_train)

# Afficher Les meilleurs hyperparamètres trouvés
print("Meilleurs hyperparamètres :")
print(random_search.best_params_)

# Utiliser Le modèle avec Les meilleurs hyperparamètres pour prédire Les classes pour Les données de test
y_pred = random_search.predict(X_test)

# Calculer Les mesures de performance
accuracy = accuracy_score(y_test, y_pred)
tn, fp, fn, tp = confusion_matrix(y_test, y_pred).ravel()
specificity = tn / (tn + fp)
print("Précision du modèle :", accuracy)
print("Specificity: ", specificity)
```

FIG. 4.22 : Code d'optimisation des hyperparamètres pour DT

- SVM

```
# Séparer Les caractéristiques (features) et La variable cible (target)
X = df.drop('Machine Status', axis=1)
y = df['Machine Status']

# Diviser Les données en ensembles d'entraînement et de test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Définir Les hyperparamètres à optimiser
param_grid = {
    'C': np.logspace(-3, 3, 7), # Paramètre de régularisation
    'kernel': ['linear', 'rbf', 'sigmoid'], # Type de noyau
    'gamma': ['scale', 'auto'] # Coefficient du noyau
}

# Créer Le modèle SVM
svm_model = SVC()

# Recherche aléatoire des meilleurs hyperparamètres
random_search = RandomizedSearchCV(svm_model, param_distributions=param_grid, scoring='f1_weighted', cv=5)
random_search.fit(X_train, y_train)

# Obtenir Les meilleurs hyperparamètres
best_params = random_search.best_params_

# Créer un nouveau modèle SVM avec Les meilleurs hyperparamètres
best_svm_model = SVC(**best_params)

# Entraîner Le modèle sur Les données d'entraînement
best_svm_model.fit(X_train, y_train)

# Faire des prédictions sur L'ensemble de test
y_pred = best_svm_model.predict(X_test)

# Calculer Les métriques sensibilité (recall) et F1-score
recall = recall_score(y_test, y_pred, average='weighted')
f1 = f1_score(y_test, y_pred, average='weighted')
```

FIG. 4.23 : Code d'optimisation des hyperparamètres pour SVM

- KNN

```
# Définir Les hyperparamètres à optimiser
param_grid = {
    'n_neighbors': np.arange(1, 20), # Nombre de voisins
    'weights': ['uniform', 'distance'], # Méthode de pondération des voisins
    'metric': ['euclidean', 'manhattan'] # Métrique de distance
}

# Créer Le modèle KNN
knn_model = KNeighborsClassifier()

# Recherche aléatoire des meilleurs hyperparamètres
random_search = RandomizedSearchCV(knn_model, param_distributions=param_grid, scoring='f1_weighted', cv=5)
random_search.fit(X_train, y_train)

# Obtenir Les meilleurs hyperparamètres
best_params = random_search.best_params_

# Créer un nouveau modèle KNN avec Les meilleurs hyperparamètres
best_knn_model = KNeighborsClassifier(**best_params)

# Entraîner Le modèle sur Les données d'entraînement
best_knn_model.fit(X_train, y_train)

# Faire des prédictions sur L'ensemble de test
y_pred = best_knn_model.predict(X_test)

# Calculer Les métriques de précision, F1-score et rappel
precision = precision_score(y_test, y_pred)
tn, fp, fn, tp = confusion_matrix(y_test, y_pred).ravel()
specificity = tn / (tn + fp)
```

FIG. 4.24 : Code d'optimisation des hyperparamètres pour KNN

- ANN

```
# Séparer Les caractéristiques (features) et La variable cible (target)
X = df.drop('Machine Status', axis=1)
y = df['Machine Status']

# Diviser Les données en ensembles d'entraînement et de test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Définir Les hyperparamètres à optimiser
param_grid = {
    'hidden_layer_sizes': [(32,), (64,), (32, 32)], # Taille des couches cachées
    'activation': ['relu', 'tanh'] # Fonction d'activation
}

# Créer Le modèle ANN
ann_model = MLPClassifier(random_state=42)

# Recherche aléatoire des meilleurs hyperparamètres
random_search = RandomizedSearchCV(ann_model, param_distributions=param_grid, scoring='f1_weighted', cv=5)
random_search.fit(X_train, y_train)

# Obtenir Les meilleurs hyperparamètres
best_params = random_search.best_params_

# Créer un nouveau modèle ANN avec Les meilleurs hyperparamètres
best_ann_model = MLPClassifier(random_state=42, **best_params, max_iter=500)

# Entraîner Le modèle sur Les données d'entraînement
best_ann_model.fit(X_train, y_train)

# Faire des prédictions sur L'ensemble de test
y_pred = best_ann_model.predict(X_test)

# Calculer Les métriques | F1-score et rappel
recall = recall_score(y_test, y_pred, average='weighted')
f1 = f1_score(y_test, y_pred, average='weighted')
```

FIG. 4.25 : Code d'optimisation des hyperparamètres pour ANN

Résultats d'optimisation des hyperparamètres

Les trois tableaux 4.3 , 4.4 et 4.5 présentent les hyperparamètres des algorithmes du premier scénario : RF, DT et KNN.

Hyperparamètres	Valeur
Min samples split	2
Min samples leaf	1
Max depth	none

TAB. 4.3 : Les meilleurs hyperparamètres de RF

Hyperparamètres	Valeur
Weights	distance
N neighbor	4
Metric	manhattan

TAB. 4.4 : Les meilleurs hyperparamètres de KNN

Hyperparamètres	Valeur
Criterion	Entropy
Max depth	10
Min samples leaf	2
Min samples split	10

TAB. 4.5 : Les meilleurs hyperparamètres de DT

Les trois tableaux 4.6 , 4.7 et 4.8 présentent les hyperparamètres des algorithmes du deuxième scénario : RF, SVM et ANN.

Hyperparamètres	Valeur
Min samples split	2
Min samples leaf	1
Max depth	none

TAB. 4.6 : Les meilleurs hyperparamètres de RF

Hyperparamètres	Valeur
Hidden layer sizes	64
Activation	tanh

TAB. 4.7 : Les meilleurs hyperparamètres de ANN

Hyperparamètres	Valeur
Kernel	Linear
Gamma	auto
C	0.1

TAB. 4.8 : Les meilleurs hyperparamètres de SVM

4.6.4 Résultats des tests

Résultats du premier scénario

Les trois matrices suivantes représentent les résultats des algorithmes RF, KNN et DT sous forme de matrices de confusion :

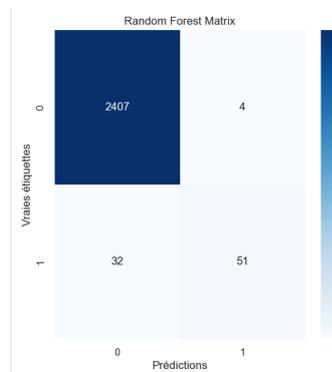


FIG. 4.26 : Matrice de confusion de RF

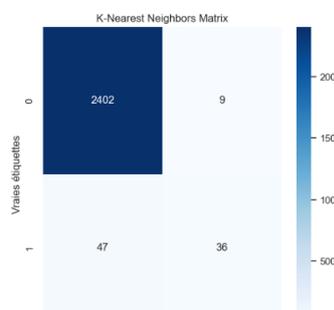


FIG. 4.27 : Matrice de confusion de KNN

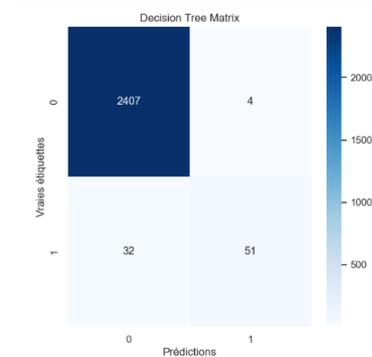


FIG. 4.28 : Matrice de confusion de DT

- Le modèle RF présente de très bonnes performances dans la détection des pannes, avec un nombre élevé de pannes correctement identifiées (2407) et un nombre extrêmement faible de fausses alarmes (4). Ces résultats démontrent la précision du modèle dans la détection des pannes, ce qui permet d'éviter les réparations inutiles.
- Le modèle KNN affiche également de bonnes performances dans la détection des pannes, avec un nombre élevé de pannes correctement identifiées (2402) et un nombre relativement faible de fausses alarmes (9). Cependant, il présente un nombre légèrement plus élevé de pannes non détectées (47) par rapport au modèle RF, ce qui indique qu'il peut manquer certaines pannes lors des prédictions. En revanche, le modèle identifie relativement moins d'états de bon fonctionnement correctement (36), ce qui peut entraîner des réparations inutiles dans certains cas.
- Les performances du modèle DT sont similaires à celles du modèle RF, avec un nombre élevé de pannes correctement identifiées (2407) et un nombre très faible de fausses alarmes (4). Cela démontre que le modèle est capable de détecter avec précision les pannes, ce qui contribue à éviter les réparations inutiles.

En conclusion, les trois modèles (RF, KNN et DT) présentent globalement de bonnes performances dans la détection des pannes, avec des taux élevés de détection des pannes

et des taux relativement faibles de fausses alarmes. Cependant, il est important de noter que le modèle KNN peut présenter un nombre légèrement plus élevé de pannes non détectées, ce qui peut entraîner des réparations inutiles. Il est donc recommandé d'utiliser d'autres mesures de performance, telles que la précision et la spécificité, pour obtenir une évaluation plus complète des performances de chaque algorithme et prendre des décisions de maintenance prédictive plus éclairées.

Le tableau suivant regroupe les mesures de performance de cette classification, en mettant en évidence les performances des modèles avant et après l'optimisation des hyperparamètres. Ces performances sont évaluées à l'aide des mesures de précision et de spécificité.

	Avant l'optimisation		Après l'optimisation	
	Précision	spécificité	Précision	spécificité
RF	0.9852256252	0.997926171712	0.98476343223	0.997926171712
KNN	0.761904761904	0.995852343425	0.982357658380	0.996267109083
DT	0.666666666666	0.986727498963	0.981956696070	0.995022812111

TAB. 4.9 : Précision et spécificité obtenues après l'optimisation des hyperparamètres

- Avant l'optimisation, le modèle RF affiche une précision de 0.985 et une spécificité de 0.998, indiquant sa capacité à classer correctement les échantillons positifs et négatifs. Le modèle KNN présente une précision de 0.762 et une spécificité de 0.996, suggérant une performance inférieure par rapport au modèle RF. Le modèle DT affiche une précision de 0.667 et une spécificité de 0.987, indiquant une performance relativement plus faible.
- Après l'optimisation, la précision du modèle RF reste élevée à 0.985, tout comme la spécificité à 0.998. Cela montre que le modèle RF maintient sa capacité à classer avec précision les échantillons positifs et négatifs après l'optimisation. Le modèle KNN montre une amélioration significative de la précision à 0.982 et de la spécificité à 0.996, ce qui indique une meilleure performance après l'optimisation. Le modèle DT présente également une amélioration, avec une précision de 0.982 et une spécificité de 0.995.

La courbe ROC permet d'ajuster le seuil de classification en fonction des besoins spécifiques du problème. En examinant les courbes ROC des trois algorithmes, il est possible de déterminer le seuil de classification optimal pour maximiser la sensibilité ou la spécificité, en fonction des exigences du projet ou des conséquences associées aux erreurs de classification.

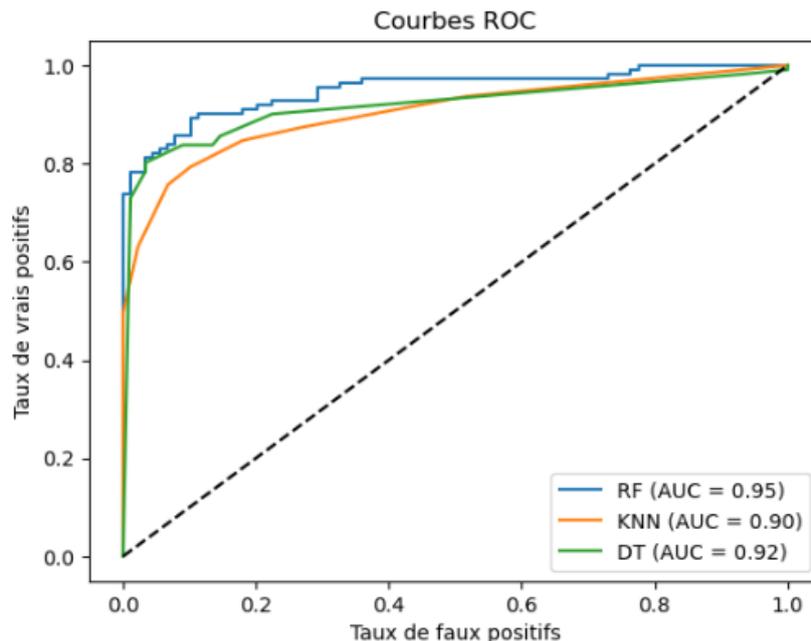


FIG. 4.29 : La courbe ROC pour le premier scénario

- Le modèle RF présente la meilleure performance globale. Sa courbe ROC est située plus haut et plus à gauche par rapport aux autres modèles, ce qui indique une capacité supérieure à distinguer les exemples de pannes des exemples de non pannes. En d'autres termes, le RF est plus sensible à la détection des pannes tout en maintenant un faible taux de fausses alarmes.
- Le modèle DT affiche également une performance respectable. Sa courbe ROC se situe entre celle du RF et du KNN, suggérant qu'il est compétent pour maintenir un équilibre entre la sensibilité (détection des pannes) et la spécificité (détection des non pannes). Le DT peut donc être considéré comme un modèle compétitif pour classer l'état de l'équipement.
- En revanche, le modèle KNN présente la performance la plus faible parmi les trois algorithmes. Sa courbe ROC est positionnée plus bas et plus à droite, ce qui indique une tendance accrue à générer des faux positifs (classification erronée en tant que panne). Cependant, il montre toujours une certaine capacité à distinguer les exemples positifs, bien qu'avec une précision inférieure par rapport aux autres modèles.

Résultats du deuxième scénario

Les trois matrices suivantes représentent les résultats des algorithmes RF, SVM et ANN sous forme de matrices de confusion :

- Pour l'algorithme RF, la matrice de confusion indique qu'il y a 195 vraies détections de pannes et aucun faux positif. De plus, il y a 1 faux négatif et 34 vrais fonctionnements corrects identifiés. Ces résultats suggèrent que l'algorithme RF possède une

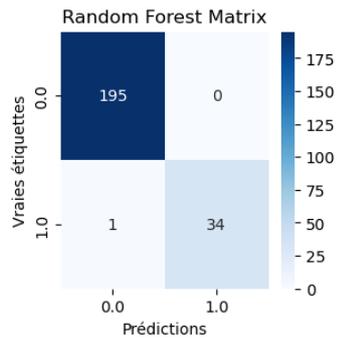


FIG. 4.30 : Matrice de confusion RF

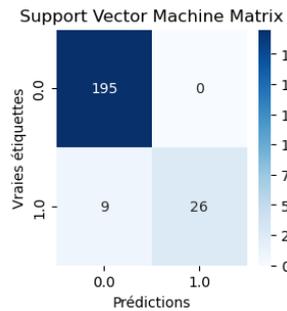


FIG. 4.31 : Matrice de confusion SVM

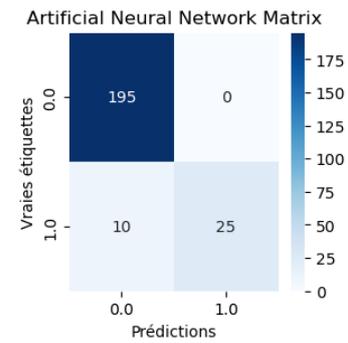


FIG. 4.32 : Matrice de confusion ANN

bonne capacité à prédire avec précision les pannes et les états de bon fonctionnement.

- En ce qui concerne l’algorithme SVM, sa matrice de confusion révèle également 195 vraies détections de pannes et aucun faux positif, ce qui indique une précision parfaite pour cette classe. Cependant, il y a 9 faux négatifs et 26 vrais fonctionnements corrects identifiés. Cela suggère que l’algorithme SVM a une légère tendance à sous-estimer les pannes, ce qui peut entraîner un taux de rappel plus bas.
- L’algorithme ANN, quant à lui, présente également 195 vraies détections de pannes et aucun faux positif, ce qui est encourageant. Cependant, il y a 10 faux négatifs et 25 vrais fonctionnements corrects identifiés, révélant une légère tendance à sous-estimer les pannes, similaire à l’algorithme SVM.

Globalement, les trois algorithmes RF, SVM et ANN montrent une bonne capacité à prédire les vraies détections de pannes, mais ils diffèrent dans leur capacité à prédire les faux négatifs et les vrais fonctionnements corrects. Il serait important d’évaluer d’autres mesures de performance, telles que la sensibilité et le score F1, afin d’obtenir une image plus complète de la performance de chaque algorithme dans le but d’éviter de manquer les alarmes des pannes.

Le tableau suivant présente les performances avant et après l’optimisation des modèles RF, ANN et SVM. Les mesures de performance incluent la sensibilité et le score F1.

	Avant l’optimisation		Après l’optimisation	
	Sensibilité	F1 score	Sensibilité	F1 score
RF	0.995652173913	0.995626227806	0.995652173913	0.995626227806
ANN	0.869565217391	0.876001144164	0.956521739130	0.953442028985
SVM	0.847826086956	0.778005115089	0.960869565217	0.95842421449

TAB. 4.10 : Sensibilité et F1 score obtenues après l’optimisation des hyperparamètres

- Avant l’optimisation, le modèle RF se démarquait avec une sensibilité de 0.995 et un score F1 de 0.995. Cela signifie que le modèle RF avait une capacité élevée à détecter

correctement les cas de panne et à éviter les faux négatifs, ce qui est essentiel pour éviter de manquer les alarmes des pannes. En revanche, les modèles ANN et SVM présentait des performances relativement inférieures, avec une sensibilité de 0.869 et 0.847 respectivement, ainsi que des scores F1 de 0.876 et 0.778. Ces résultats indiquent une tendance plus prononcée à sous-estimer la présence de pannes et à générer des faux négatifs, ce qui peut entraîner un risque accru de manquer des alarmes de pannes.

- Après l'optimisation, les performances des modèles RF, ANN et SVM ont évolué. Le modèle RF a maintenu sa sensibilité élevée à 0.995 et son score F1 élevé à 0.995, confirmant sa capacité à détecter avec précision les équipements en panne et à minimiser les faux négatifs. Les modèles ANN et SVM ont montré une amélioration notable de leurs performances, avec une sensibilité améliorée à 0.956 et 0.96 respectivement, ainsi que des scores F1 améliorés à 0.953 et 0.958. Ces améliorations indiquent une meilleure capacité des modèles ANN et SVM à éviter de manquer les alarmes des pannes après l'optimisation.

La figure ci-dessous illustre la courbe ROC pour les trois algorithmes suivants : SVM, ANN et RF.

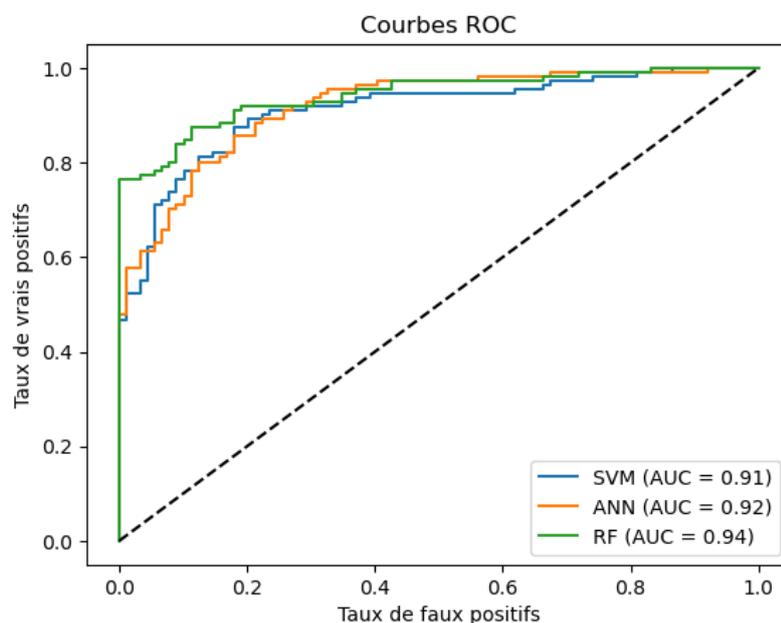


FIG. 4.33 : La courbe ROC pour le 2^{ème} scénario

- La proximité de la courbe ROC de RF avec le coin supérieur droit suggère de bonnes performances pour la détection des pannes de compresseur. Cela signifie que le modèle RF a une capacité élevée à détecter les véritables pannes tout en tolérant un certain niveau de fausses alertes (taux de faux positifs élevé).
- L'analyse comparative des courbes ROC d'ANN et du SVM révèle des informations intéressantes. Initialement, la courbe ROC de SVM se trouve en dessous de celle de l'ANN, indiquant une sensibilité et une spécificité relativement plus faibles pour

SVM dans la détection des pannes de l'équipement par rapport à l'ANN. Cependant, à mesure que l'on progresse sur la courbe ROC, la situation s'inverse légèrement, et la courbe ROC de SVM dépasse celle de l'ANN. Ce changement suggère que, dans certaines zones spécifiques de l'espace des caractéristiques, SVM peut avoir des performances légèrement supérieures en termes de sensibilité et de spécificité pour détecter les pannes de l'équipement par rapport à l'ANN.

4.6.5 Comparaison entre la première et la deuxième sélection

Nous avons réalisé une comparaison entre les résultats obtenus lors de la première sélection d'algorithmes basée sur la somme des évaluations réalisée dans la partie 2.5.2. et ceux obtenus lors de la deuxième sélection effectuée avec notre méthode basée sur le vote pondéré .

la table ci-dessous compare les résultats des algorithmes de deux sélection pour le premier scénario

Sélection	Algorithmes	Performance	
		Précision	Spécificité
1ère	RF	0.98476	0.99792
	KNN	0.98235	0.99626
	ANN	0.95242	0.96848
2ème	RF	0.98476	0.99792
	KNN	0.98235	0.99626
	DT	0.98195	0.99502

TAB. 4.11 : Tableau comparatif des résultats des deux méthodes de sélection (scénario 1)

Après avoir testé les résultats de la première sélection, il est clair que la deuxième sélection a apporté une nette amélioration par rapport à la première. Les performances des algorithmes choisis lors de la deuxième sélection se sont révélées significativement supérieures (la performance de DT est supérieure à celle de ANN), démontrant ainsi les avantages de l'approche basée sur la table pondérée.

la table ci-dessous compare les résultats des algorithmes de deux sélections pour le deuxième scénario

En termes de sensibilité et de score F1, la première sélection a démontré son efficacité accrue. Cependant, dans le contexte spécifique de notre problème où un algorithme robuste est nécessaire, le KNN n'est pas le meilleur choix en raison de sa sensibilité aux données bruitées, qui correspond à notre base de données. Cette limitation a été identifiée lors de la deuxième sélection, ce qui a conduit à la conclusion que la deuxième sélection était préférable.

Sélection	Algorithmes	Performance	
		Sensibilité	F1-score
1ère	RF	0.99565	0.99562
	KNN	0.94837	0.95217
	ANN	0.95652	0.95344
2ème	RF	0.99565	0.9956
	ANN	0.95652	0.95344
	SVM	0.96086	0.95842

TAB. 4.12 : Tableau comparatif des résultats des deux méthodes de sélection (scénario 2)

4.6.6 Discussion

Avant l’optimisation, le modèle RF présente les meilleures performances en termes de précision et de spécificité. Cependant, après l’optimisation, les modèles KNN, ANN et DT ont montré des améliorations significatives, se rapprochant des performances du modèle RF. Ces résultats soulignent l’importance de l’optimisation des modèles pour améliorer leurs performances dans la prédiction précise des exemples positifs et négatifs. En ajustant judicieusement les valeurs des hyperparamètres, les modèles ont été en mesure d’atteindre des niveaux de précision plus élevés et d’améliorer leurs capacités de prédiction.

RF s’est révélé être le meilleur choix pour les tâches de maintenance. Ce modèle a démontré des performances supérieures en termes de précision et de capacité de classification, ce qui en fait une option fiable pour la tâche spécifique de maintenance.

Les modèles DT et RF ont montré des performances similaires dans le cas du premier scénario. Cela peut être expliqué par le fait que le RF est basé sur l’agrégation de multiples DT, et donc les deux modèles partagent le même principe de fonctionnement de l’arbre de décision.

Bien que le modèle ANN affiche une précision élevée, il présente des inconvénients dans notre scénario spécifique. Son temps d’apprentissage et de prédiction est significativement plus long par rapport aux autres modèles, ce qui limite son utilité pratique dans des situations où des délais de réponse rapides sont nécessaires. Cet exemple met en évidence l’importance de la deuxième sélection de modèles.

Le SVM présente des limitations dans le cas des grandes bases de données (scénario 1) en raison de sa complexité de calcul. Bien qu’il puisse encore être utilisé dans ces cas, il est recommandé d’explorer des alternatives plus efficaces, telles que le RF ou le DT, qui offrent de bonnes performances tout en étant plus rapides.

Les deux études de cas ont clairement mis en évidence l’influence de l’alignement stratégique sur le choix des algorithmes et des mesures d’évaluation. Elles ont démontré que l’alignement des objectifs stratégiques de l’entreprise avec les approches analytiques

est essentiel pour obtenir des résultats pertinents.

4.7 Comparaison avec un autre travail

Voici une comparaison entre notre méthode de sélection basée sur le vote pondéré et une autre méthode basée sur le calcul de la somme des évaluations de chaque algorithme :

Méthode de sélection d'algorithme	Avantages	Inconvénients
Notre méthode basée sur le vote pondéré.	- Prend en compte les objectifs spécifiques.	- Nécessite une pondération subjective des critères.
Calcul de la somme des évaluations [2].	- Méthode simple et intuitive.	- Ne permet pas de différencier l'importance des critères

TAB. 4.13 : Comparaison entre la méthode de vote pondéré et la méthode basée sur la somme des évaluations

La méthode basée sur le calcul de la somme des évaluations de chaque algorithme est simple et intuitive. Elle consiste à sommer les évaluations de chaque algorithme sur différents critères pour obtenir un score global. Cependant, cette méthode ne permet pas de différencier l'importance relative des critères, ce qui peut conduire à des résultats moins précis ou adaptés aux objectifs spécifiques de la maintenance prédictive.

La méthode de vote pondéré offre donc une approche plus fine et plus personnalisée en considérant explicitement les objectifs spécifiques. En attribuant des poids aux critères en fonction de leur importance, elle permet de prendre des décisions plus éclairées et mieux adaptées à la résolution du problème de la maintenance prédictive. Cela permet d'optimiser les résultats et de mieux répondre aux besoins spécifiques du domaine.

4.8 Conclusion

En conclusion, ce chapitre nous a permis de tester et d'évaluer l'efficacité de notre processus de PdM basé sur ML sur deux scénarios distincts. Grâce à une approche méthodique, nous avons réalisé une sélection des algorithmes les plus pertinents pour chaque scénario. Les résultats ont montré que le modèle RF a démontré des performances supérieures en termes de précision et de capacité de classification, le positionnant comme le choix privilégié pour les tâches de maintenance. Cependant, l'optimisation des hyperparamètres a permis d'améliorer les performances des modèles KNN et DT et de les rapprocher des performances du RF. Par ailleurs, nous avons identifié les limitations de certains modèles, tels que le temps de calcul élevé de l'ANN et la complexité du SVM pour les grandes bases de données. Enfin, nous avons souligné l'importance de l'alignement stratégique dans le choix des algorithmes et des mesures d'évaluation, mettant en évidence

l'influence des objectifs stratégiques sur les résultats obtenus. Ces résultats confirment la validité de notre processus de PDM basé sur ML et offrent des perspectives prometteuses pour son application dans le domaine industriel.

Conclusion et perspectives

Conclusion générale

L'objectif principal de ce mémoire est de développer un processus méthodique pour l'application de la ML à la PdM, en prenant en compte à la fois les objectifs de la maintenance prédictive et les spécifications du problème. Une attention particulière a été accordée à la sélection des algorithmes afin d'atteindre de manière optimale les objectifs fixés pour ce projet. En effet, compte tenu de la complexité des systèmes industriels et du volume massif de données à gérer, la nécessité de choisir le modèle de ML approprié se fait de plus en plus pressante.

Dans le chapitre 2, nous avons réalisé une revue de la littérature et exploré les concepts fondamentaux de la ML dans le but d'identifier les algorithmes les plus couramment utilisées pour la mise en place d'un système de maintenance prédictive. À partir de cette étude, nous avons élaboré notre méthodologie de recherche, qui est exposée en détail dans le chapitre 3. De plus, nous avons procédé à une étude comparative pour sélectionner trois modèles de ML, à savoir le KNN, le SVM et le RF. Dans la majorité des recherches examinées, le RF semblait offrir les meilleurs résultats. Toutefois, cette constatation a été validée dans le cadre de nos études de cas.

Dans le chapitre 3, nous avons mis en œuvre l'approche DMAIC pour améliorer le processus de la machine learning dans le contexte de la maintenance prédictive. La phase mesurer est basée sur un diagramme en toile d'araignée, ce qui nous a permis d'acquérir une meilleure compréhension des exigences de la maintenance prédictive et de les analyser en détail dans la phase suivante. Ces deux phases ont révélé les lacunes à améliorer ainsi que les avantages potentiels de la mise en place de ce nouveau processus. Parmi ces lacunes, nous avons identifié la nécessité de spécialiser la méthode de choix d'algorithme en utilisant un vote pondéré pour l'adapter aux spécifications du problème. Ce processus, et plus particulièrement cette méthode de choix d'algorithmes, ont démontré leur efficacité dans deux études de cas distinctes présentées dans le chapitre 4.

Dans le chapitre 4, nous avons appliqué notre processus à deux scénarios distincts, notamment un cas réel rencontré lors de notre stage à l'entreprise Condor. Dans cette partie, les résultats obtenus ont révélé des différences entre les algorithmes de ML sélectionnés lors de la première et de la deuxième sélection. Les résultats ont démontré que le DT et le RF ont affiché des mesures de précision et de rappel élevées, ce qui les rend plus performants que le KNN.

Les résultats obtenus dans le cadre de ces 2 scénarios ont confirmé que chaque classificateur possède ses propres paramètres qui jouent un rôle crucial dans l'amélioration de la précision des résultats. Il est donc essentiel de rechercher les paramètres optimaux pour construire un bon classificateur. Parmi les différents algorithmes testés, le SVM se révèle être une technique puissante pour la classification des données. En revanche, malgré ses solides fondements théoriques, sa grande précision de classification et ses succès observés dans des travaux de recherche antérieurs, le SVM n'est pas adapté à la classification de grands ensembles de données, comme cela est le cas dans notre étude de cas 1. En effet, la complexité d'entraînement du SVM dépend fortement de la taille de l'ensemble de données.

Dans les problèmes de maintenance où les données de pannes sont généralement rares, entraînant des bases de données déséquilibrées, la métrique d'évaluation la plus couramment utilisée, l'exactitude (Accuracy), n'est pas suffisamment efficace. Cette métrique peut être trompeuse lorsque les données sont déséquilibrées. Dans de tels cas, d'autres paramètres d'évaluation tels que la matrice de confusion, la précision, le rappel et la spécificité doivent être pris en compte en fonction des objectifs et des spécificités du problème. Ainsi, lorsqu'il s'agit de données déséquilibrées, la décision finale concernant la sélection du modèle devrait envisager une combinaison de différentes mesures plutôt que de se baser sur une seule mesure.

Le RF se révèle être un algorithme idéal pour traiter le déséquilibre extrême des classes, et ce, pour deux raisons principales. Premièrement, la possibilité d'incorporer des pondérations de classe dans le classificateur de forêt aléatoire lui confère une sensibilité aux coûts, pénalisant ainsi les classifications erronées de la classe minoritaire. Deuxièmement, il combine les techniques d'échantillonnage et d'apprentissage en ensemble, en sous-échantillonnant la classe majoritaire et en construisant des arbres sur un ensemble de données plus équilibré.

Perspectives

L'exploration du domaine AutoML (Machine Learning Automatique) dans la PdM offre une perspective de recherche prometteuse. L'AutoML vise à automatiser le processus de développement de modèles de ML, rendant ainsi l'apprentissage automatique plus accessible aux praticiens de la PdM. Cette exploration vise à étudier comment l'AutoML peut améliorer les méthodes de PdM existantes en automatisant la sélection, la configuration et l'optimisation des modèles de ML. Les objectifs incluent l'examen de l'état de l'art de l'AutoML, l'identification des opportunités d'application, l'évaluation des performances de l'AutoML dans des scénarios de PdM spécifiques, et l'analyse des avantages et des limitations de cette approche. Cette perspective de recherche contribuera à l'avancement des connaissances dans le domaine de l'AutoML appliqué à la PdM et fournira des informations précieuses pour les praticiens souhaitant exploiter les avantages de l'AutoML dans leurs stratégies de maintenance prédictive.

Durant la phase de contrôle, nous avons souligné l'importance du feedback et des avis d'experts. Une amélioration potentielle de notre processus consisterait à intégrer une approche basée sur les règles afin de renforcer la performance de nos résultats. Cette approche permettrait de tirer parti des connaissances et de l'expertise des spécialistes du domaine pour guider et améliorer les décisions prises par le modèle. En combinant les données et les règles, nous pourrions obtenir des résultats plus précis et plus fiables, en tirant parti de l'expérience humaine pour compléter les capacités du modèle. L'intégration de cette approche basée sur les règles pourrait constituer une étape essentielle pour optimiser notre processus et obtenir des résultats de meilleure qualité.

Dans le contexte d'un projet de ML pour la PdM, pour rendre les critères de comparaison plus complets, nous pourrions ajouter les critères suivants : l'intégrité du système, le coût d'implémentation et la maintenance de l'algorithme. Il est impératif que les algo-

rithmes de ML puissent s'interfacer avec les systèmes de collecte de données en temps réel et les systèmes de contrôle pour recueillir des données d'entrée pertinentes et fournir des résultats exploitables aux équipes de maintenance. Ce faisant, les coûts et les délais de développement associés à l'implémentation de l'algorithme de ML peuvent être réduits. De plus, il est nécessaire de prendre en compte le coût des logiciels, de l'infrastructure nécessaire et de la formation des employés lors du processus d'implémentation, afin d'assurer le succès global du modèle. Ainsi, pour garantir une précision et une pertinence à long terme, les algorithmes de machine learning doivent être conçus en tenant compte de la facilité de maintenance et de mise à jour, en prenant en compte à la fois la quantité de données disponibles et l'interprétabilité des résultats.

En utilisant la fonctionnalité de sous-processus offerte par Signavio, nous pouvons exploiter davantage cette fonctionnalité en proposant des sous-processus spécifiques à chaque étape de notre processus. Par exemple, nous pouvons concevoir un sous-processus pour la collecte de données, un sous-processus pour la préparation des données, etc. Cela nous permettrait d'affiner notre approche et de mieux organiser les différentes tâches de notre projet, améliorant ainsi l'efficacité et la clarté de notre processus global.

Bibliographie

- [1] 60-000, Norme Française. *Fonction Maintenance*. Association Française de Normalisation, 2002.
- [2] OUADAH ABDELFTAH. “Sélection D’un Algorithme de l’Apprentissage Automatique Supervisé Approprié pour la Maintenance Prédicative”. Mém. de mast. ´ Ecole Nationale Supérieure de Technologie Département de Génie Industriel et Maintenance, 2021.
- [3] Mohamed Gamal ABOELMAGED. “Predicting e-readiness at firm-level : An analysis of technological, organizational and environmental (TOE) effects on e-maintenance readiness in manufacturing firms”. In : *International Journal of Information Management* 34.5 (2014), p. 639-651.
- [4] Mounia ACHOUCHE et al. “On predictive maintenance in industry 4.0: Overview, models, and challenges”. In : *Applied Sciences* 12.16 (2022), p. 8081.
- [5] Pankaj AJIT. “Prediction of employee turnover in organizations using machine learning algorithms”. In : *algorithms* 4.5 (2016), p. C5.
- [6] Mohammed H ALSHARIF et al. “Machine learning algorithms for smart data analysis in internet of things environment : taxonomies and research trends”. In : *Symmetry* 12.1 (2020), p. 88.
- [7] Duc Tran ANH, Katarzyna SKRZYPEK et al. “The predictive maintenance concept in the maintenance department of the “Industry 4.0” production enterprise”. In : *Foundations of Management* 10.1 (2018), p. 283-292.
- [8] BK BHAVITHA, Anisha P RODRIGUES et Niranjana N CHIPLUNKAR. “Comparative study of machine learning techniques in sentimental analysis”. In : *2017 International conference on inventive communication and computational technologies (ICICCT)*. IEEE. 2017, p. 216-221.
- [9] Hetal BHAVSAR et Amit GANATRA. “A comparative study of training algorithms for supervised machine learning”. In : *International Journal of Soft Computing and Engineering (IJSCE)* 2.4 (2012), p. 2231-2307.
- [10] Sinem BOZKURT et al. “A comparative study on machine learning algorithms for indoor positioning”. In : *2015 International Symposium on Innovations in Intelligent SysTems and Applications (INISTA)*. IEEE. 2015, p. 1-8.
- [11] Thyago P CARVALHO et al. “A systematic literature review of machine learning methods applied to predictive maintenance”. In : *Computers & Industrial Engineering* 137 (2019), p. 106024.

- [12] Jovani DALZUCHIO et al. “Machine learning and reasoning for predictive maintenance in Industry 4.0: Current status and challenges”. In : *Computers in Industry* 123 (2020), p. 103298.
- [13] Kajaree DAS et Rabi Narayan BEHERA. “A survey on machine learning : concept, algorithms and applications”. In : *International Journal of Innovative Research in Computer and Communication Engineering* 5.2 (2017), p. 1301-1309.
- [15] BENCHETTOUH Salah EDDINE. “Elaboration d’un système de prédiction des pannes et de planification des maintenances”. Mém. de mast. UNIVERSITE MOHAMED BOUDIAF-M’SILA-FACULTE MATHEMATIQUES ET DE L’INFORMATIQUE, 2019.
- [16] Alejandro Germán FRANK, Lucas Santos DALENOGARE et Néstor Fabián AYALA. “Industry 4.0 technologies : Implementation patterns in manufacturing companies”. In : *International Journal of Production Economics* 210 (2019), p. 15-26.
- [17] Abhijeet GHADGE et al. “The impact of Industry 4.0 implementation on supply chains”. In : *Journal of Manufacturing Technology Management* 31.4 (2020), p. 669-686.
- [18] Baljinder GHOTRA, Shane MCINTOSH et Ahmed E HASSAN. “Revisiting the impact of classification techniques on the performance of defect prediction models”. In : *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*. T. 1. IEEE. 2015, p. 789-800.
- [19] Małgorzata JASIULEWICZ-KACZMAREK et Arkadiusz GOLA. “Maintenance 4.0 Technologies for Sustainable Manufacturing -an Overview”. In : *IFAC-PapersOnLine* 52-10 (nov. 2019), p. 91-96.
- [20] Vansh JATANA. *Machine Learning Algorithms*. Juin 2019.
- [21] Ameeth KANAWADAY et Aditya SANE. “Machine learning for predictive maintenance of industrial machines using IoT sensor data”. In : *2017 8th IEEE international conference on software engineering and service science (ICSESS)*. IEEE. 2017, p. 87-90.
- [22] Sotiris B KOTSIANTIS, Ioannis ZAHARAKIS, P PINTELAS et al. “Supervised machine learning : A review of classification techniques”. In : *Emerging artificial intelligence applications in computer engineering* 160.1 (2007), p. 3-24.
- [23] Christian KRUPITZER et al. “A survey on predictive maintenance for industry 4.0”. In : *arXiv preprint arXiv :2002.08224* (2020).
- [24] Dwi KUSUMANINGRUM, Nani KURNIATI et Budi SANTOSA. “Machine learning for predictive maintenance”. In : *Proceedings of the International Conference on Industrial Engineering and Operations Management. IEOM Society International*. 2021, p. 2348-2356.
- [25] Jay LEE et al. “Industrial big data analytics and cyber-physical systems for future maintenance & service innovation”. In : *Procedia cirp* 38 (2015), p. 3-7.
- [26] Stefan LESSMANN et al. “Benchmarking classification models for software defect prediction : A proposed framework and novel findings”. In : *IEEE transactions on software engineering* 34.4 (2008), p. 485-496.

- [27] Myriam LOPEZ. “Prédiction d’erreurs critiques pour l’aide à la maintenance industrielle”. In : *Rencontres des Jeunes Chercheurs en Intelligence Artificielle (RJ-CIA’21) Plate-Forme Intelligence Artificielle (PFIA’21)*. 2021, p. 82-83.
- [28] Batta MAHESH. “Machine learning algorithms-a review”. In : *International Journal of Science and Research (IJSR).[Internet]* 9 (2020), p. 381-386.
- [29] INNOCENT MATEYAUNGA. “Predictive Maintenance Using Machine Learning”. Mém. de mast. Departement of Electrical et Electronics Engineering Speciality Automation Option : Industrial Computer Science, 2020.
- [30] Amr E MOHAMED. “Comparative study of four supervised machine learning techniques for classification”. In : *International Journal of Applied* 7.2 (2017), p. 1-15.
- [31] FY OSISANWO et al. “Supervised machine learning algorithms : classification and comparison”. In : *International Journal of Computer Trends and Technology (IJCTT)* 48.3 (2017), p. 128-138.
- [32] Marina PAOLANTI et al. “Machine learning approach for predictive maintenance in industry 4.0”. In : *2018 14th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA)*. IEEE. 2018, p. 1-6.
- [33] Martin PECH, Jaroslav VRCHOTA et Jiří BEDNÁŘ. “Predictive maintenance and intelligent sensors in smart factory”. In : *Sensors* 21.4 (2021), p. 1470.
- [34] Qinglin QI et Fei TAO. “Digital twin and big data towards smart manufacturing and industry 4.0: 360 degree comparison”. In : *Ieee Access* 6 (2018), p. 3585-3593.
- [35] Yongyi RAN et al. “A survey of predictive maintenance : Systems, purposes and approaches”. In : *arXiv preprint arXiv :1912.07383* (2019).
- [36] Susmita RAY. “A quick review of machine learning algorithms”. In : *2019 International conference on machine learning, big data, cloud and parallel computing (COMITCon)*. IEEE. 2019, p. 35-39.
- [37] Sufiyan SAJID et al. “Data science applications for predictive maintenance and materials science in context to Industry 4.0”. In : *Materials today : proceedings* 45 (2021), p. 4898-4905.
- [38] Go Muan SANG et al. “Predictive maintenance in industry 4.0”. In : *Proceedings of the 10th International Conference on Information Systems and Technologies*. 2020, p. 1-11.
- [39] Kapil SETHI et al. “Comparative analysis of machine learning algorithms on different datasets”. In : *Circulation in Computer Science International Conference on Innovations in Computing (ICIC 2017)*. T. 87. 2019.
- [40] Gian Antonio SUSTO et al. “Machine learning for predictive maintenance : A multiple classifier approach”. In : *IEEE transactions on industrial informatics* 11.3 (2014), p. 812-820.
- [41] Emiliano TRAINI et al. “Machine learning framework for predictive maintenance in milling”. In : *IFAC-PapersOnLine* 52.13 (2019), p. 177-182.
- [42] Umar M AL-TURKI et al. “Maintenance in manufacturing environment : an overview”. In : *Integrated maintenance planning in manufacturing systems* (2014), p. 5-23.

- [43] Tiago ZONTA et al. “Predictive maintenance in the Industry 4.0: A systematic literature review”. In : *Computers & Industrial Engineering* 150 (2020), p. 106889.

Webographie

- [14] DKT GROUP. *Les six étapes clés de l'analyse prédictive*. <https://dkt-group.com/les-six-etapes-cles-de-lanalyse-predictive/>. février 2023.

Annexes

Annexe A

Présentation de l'entreprise Condor

A.1 Organigramme de l'entreprise

SPA CONDOR ELECTRONICS est une société spécialisée dans la fabrication des équipements électronique et électroménager, informatique, agro-alimentaire, emballage, matériaux de construction et commerce international.) Dépôt à l'INAPI le 30 avril 2003, ce qui leur a permis de brasser un gros volume d'affaire et de projets.

Le succès de Condor repose essentiellement sur le facteur humain (femme et homme), éléments clés de son approche, ces derniers ont été investis d'une autonomie telle qu'ils ne peuvent que s'épanouir et développé leurs compétences professionnelles. L'enthousiasme et l'implication totale de chacun et chacune au service de l'entreprise, dans un même esprit d'équipe a contribué à atteindre les objectifs fixés, à savoir l'élargissement de la gamme à d'autres produits toujours plus innovants, qui apportent plus de confort, de satisfaction, de simplicité d'utilisation et de sécurité Aujourd'hui, le groupe représente un des conglomérats d'entreprises Algériennes les plus puissants et actives sur la sphère économique du pays.

L'organigramme de l'entreprise est affiché dans la figure ci-dessous.

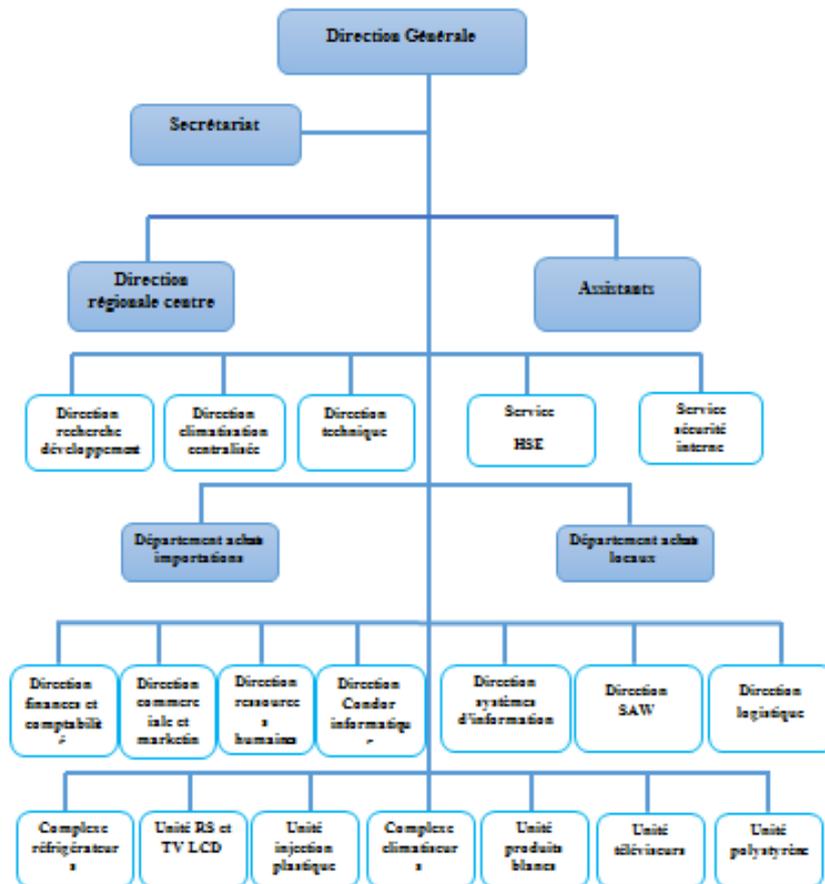


FIG. A.1 : Organigramme de l'entreprise

A.2 Machine étudiée

Le compresseur GA 75 VSD ou GA 90 VSD est un compresseur d'air mono-étagé à vis à injection d'huile, entraîné par un moteur électrique. Il est intégré dans un capotage insonorisant et est disponible en version refroidie par air ou par eau. Le compresseur est contrôlé par le régulateur Elektronikon® Graphic d'Atlas Copco, avec un module de contrôle électronique monté sur la porte. La technologie VSD d'Atlas Copco ajuste automatiquement le régime du moteur en fonction de la demande en air comprimé, assurant ainsi une efficacité énergétique

Annexe B

Métriques d'évaluation

B.1 Métriques de classification

Il existe différentes mesures pour évaluer les performances d'un classificateur. Aucune mesure individuelle ne peut nous donner l'ensemble de l'histoire sur les performances du classificateur [30].

B.1.1 La matrice de confusion

La matrice de confusion est un tableau de taille $n \times n$ utilisé pour visualiser les résultats des modèles prédictifs dans les problèmes de classification, où n est le nombre de classes dans l'ensemble de données. Cette matrice croise les classes cibles réelles avec les classes prédites par le modèle. Elle permet de déterminer le nombre d'instances correctement classées et mal classées.

		Predicted Class	
		Positive	Negative
Actual Class	Positive	True Positive (TP)	False Negative (FN) Type II Error
	Negative	False Positive (FP) Type I Error	True Negative (TN)

FIG. B.1 : Matrice de confusion

- VP : vrais positifs est le nombre d'instances positives correctement classifiées.
- FP : faux positifs est le nombre d'instances négatives et qui sont prédites comme positives.

- FN : faux négatifs est le nombre d'instances positives classifiées comme négatives.
- VN : vrais négatifs est le nombre d'instances négatives correctement classifiées.

À partir de la matrice de confusion on peut calculer plusieurs métriques qu'on va expliquer dans les sections suivantes.

B.1.2 Le taux de succès

Le taux de succès (Accuracy) :C'est le pourcentage de prédictions correctes réalisées par le modèle par rapport à l'ensemble des prédictions effectuées.

$$\text{Accuracy} = \left(\frac{\text{VP} + \text{VN}}{\text{VP} + \text{VN} + \text{FP} + \text{FN}} \right)$$

B.1.3 Sensitivité

La sensivité ou le rappel (recall) est le pourcentage des instances positives correctement identifiées.

$$\text{Rappel} = \left(\frac{\text{VP}}{\text{VP} + \text{FN}} \right)$$

B.1.4 Précision

La précision (precision) est le pourcentage de prédictions positives qui sont correctes.

$$\text{Précision} = \left(\frac{\text{VP}}{\text{VP} + \text{FP}} \right)$$

B.1.5 Spécificité

La spécificité (specificity) est le pourcentage de cas négatifs qui ont été prédits comme négatifs.

$$\text{Spécificité} = \left(\frac{\text{VN}}{\text{VN} + \text{FP}} \right)$$

B.1.6 Le score F1

c'est une mesure qui combine la précision et le rappel d'un modèle de classification pour fournir une évaluation globale de sa performance.

$$\text{Score F1} = \frac{2 * (\text{Recall} * \text{Précision})}{\text{Recall} + \text{Précision}}$$

B.1.7 La courbe ROC

La courbe ROC (Receiver Operating Characteristic) est un graphique utilisé pour évaluer la performance d'un modèle de classification. Elle montre la relation entre le taux de vrais positifs et le taux de faux positifs à différents seuils de classification. Une courbe ROC idéale est proche du coin supérieur gauche, ce qui indique une meilleure capacité du modèle à distinguer les classes

Annexe C

Optimisation des hyperparamètres

C.1 Les hyperparamètres spécifique à chaque algorithme

Le réglage optimal de hyperparamètres dépendra du problème spécifique que nous essayons de résoudre et des caractéristiques de nos données. Ci-dessous, vous trouverez les hyperparamètres spécifiques des algorithmes de machine learning utilisés dans notre étude :

RF

- Nombre d'arbres (n estimators)
- Profondeur maximale de l'arbre (max depth)
- Nombre minimum d'échantillons requis pour diviser un nœud interne (min samples split)
- Nombre minimum d'échantillons requis pour être une feuille (min samples leaf)
- Nombre maximum de variables à considérer pour une division (max features)

SVM

- Paramètre de régularisation (C) : contrôle la pénalité pour les erreurs de classification.
- Paramètre du noyau (kernel) : spécifie le type de fonction noyau à utiliser (linéaire, polynomial, RBF, etc.).
- Gamma (gamma) : un paramètre du noyau RBF qui influence la portée des exemples d'entraînement utilisés pour la classification.
- Tolérance à l'arrêt (tol) : le critère d'arrêt pour l'optimisation.

- Classe de poids (class weight) : attribue des poids différents aux classes pour traiter les déséquilibres.

DT

- Critère de sélection de la division (criterion) : mesure utilisée pour évaluer la qualité de la division d'un nœud (gini, entropie).
- Profondeur maximale de l'arbre (max depth)
- Nombre minimum d'échantillons requis pour diviser un nœud interne (min samples split)
- Nombre minimum d'échantillons requis pour être une feuille (min samples leaf)
- Nombre maximum de variables à considérer pour le choix d'une meilleur division (max features) Artificial

ANN

- Architecture du réseau (nombre de couches, nombre de neurones par couche) :
- Fonction d'activation : la fonction utilisée pour introduire la non-linéarité dans le réseau (relu, sigmoïde, tangente hyperbolique, etc.).
- Nombre d'itérations (epochs) : le nombre de fois que l'algorithme d'apprentissage parcourt l'ensemble de données d'entraînement.
- Taille du lot (batch size) : le nombre d'échantillons utilisés pour mettre à jour les poids après chaque itération

KNN

- k : Le nombre de voisins les plus proches à considérer lors de la prédiction.
- Métrique de distance : La mesure utilisée pour calculer la similarité entre les points (par exemple, la distance euclidienne ou la distance de Manhattan).
- Fonction de pondération : Une fonction optionnelle qui donne plus de poids aux voisins les plus proches lors de la prédiction.
- Stratégie de vote : La méthode utilisée pour déterminer la classe ou la valeur de prédiction en cas d'égalité de voix parmi les voisins les plus proches.
- Prétraitement des données : Les étapes de prétraitement nécessaires avant d'appliquer KNN, telles que la normalisation des caractéristiques.

C.2 Méthode recherche aléatoire pour l'optimisation des hyperparamètres

La méthode de recherche aléatoire pour l'optimisation des hyperparamètres est une approche qui consiste à sélectionner aléatoirement des combinaisons d'hyperparamètres à évaluer pour améliorer les performances d'un modèle d'apprentissage automatique. Elle offre une alternative efficace à la recherche exhaustive en testant un sous-ensemble aléatoire de combinaisons, ce qui permet d'économiser du temps de calcul.