

Compétences nécessaires pour intégrer l'utilisation de
l'apprentissage automatique en techniques de maintenance
industrielle

par

David CÉLESTIN

MÉMOIRE PRÉSENTÉ À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
COMME EXIGENCE PARTIELLE À L'OBTENTION DE
LA MAÎTRISE AVEC MÉMOIRE EN GESTION DE L'INNOVATION
M. Sc. A.

MONTRÉAL, LE 31 MARS 2026

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

©Tous droits réservés, David Célestin, 2026

©Tous droits réservés

Cette licence signifie qu'il est interdit de reproduire, d'enregistrer ou de diffuser en tout ou en partie, le présent document. Le lecteur qui désire imprimer ou conserver sur un autre média une partie importante de ce document, doit obligatoirement en demander l'autorisation à l'auteur.

PRÉSENTATION DU JURY

CE MÉMOIRE A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE :

M. Michel Rioux, directeur de mémoire
Département de génie des systèmes à l'École de technologie supérieure

M. Tony Wong, codirecteur de mémoire
Département de génie des systèmes à l'École de technologie supérieure

M. Mickaël Gardoni, président du jury
Département de génie des systèmes à l'École de technologie supérieure

M. Julio Cesar Montecinos, membre du jury
Département de génie des systèmes à l'École de technologie supérieure

IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC

LE 09 MARS 2026

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

REMERCIEMENTS

Je souhaite tout d'abord remercier la direction et mes collègues au cégep du Vieux Montréal, sans qui ce travail de recherche n'aurait pas été possible. Tout au long de mon parcours, par leur confiance, ils m'ont permis d'aller jusqu'au bout de ce projet.

Je remercie mon directeur de maîtrise, M. Michel Rioux pour son encadrement, ses encouragements et sa confiance tout au long de ce travail de recherche. Par ses interventions, il a su me guider afin de concentrer mes efforts sur les objectifs de ce mémoire.

Je tiens également à remercier mon co-directeur de maîtrise, M. Tony Wong pour son accompagnement, sa patience et la qualité de ses interventions, tant pour la rédaction de mon mémoire ainsi que l'évolution dans mon rôle de chercheur.

Je souhaite aussi remercier tous les intervenants ayant participé à alimenter la réflexion autour de ce projet, Bruno Agard, Catherine Boyer, Magali Marcheschi, Jean-Christophe Voyer, les membres du comité de partage des bonnes pratiques en maintenance BBA, mes collègues au département de génie mécanique et maintenance industrielle ainsi que le bureau de l'environnement et de l'innovation du cégep du Vieux Montréal. Les échanges et les réflexions ont permis d'orienter la problématique de mon mémoire et d'enrichir mon travail.

Enfin, je remercie ma famille, tout particulièrement Marie-Eve et Éloi pour leur patience, les encouragements et le soutien tout au long de ce parcours. Je remercie mes amis et toutes les personnes qui ont contribué à la réalisation de ce mémoire, je vous suis reconnaissant.

Utilisation de l'apprentissage automatique en technologie de maintenance industrielle pour la détection d'anomalies

David CÉLESTIN

RÉSUMÉ

L'utilisation des données et de l'intelligence artificielle occupe une place de plus en plus importante dans le secteur manufacturier au Québec. Pour répondre aux besoins de l'industrie, il est essentiel de mettre de l'avant une culture de collecte et d'analyse des données. Actuellement, la formation des technologues en maintenance industrielle ne comporte pas de compétences liées aux technologies numériques ou à l'analytique des données. Parmi les technologies ayant un impact sur le rôle du technologue, on note l'internet des objets, les données de masse et l'intelligence artificielle, car ces technologies sont au cœur des machines connectées. Dans cette étude, une pompe centrifuge est utilisée afin de déterminer comment mettre en place un premier projet d'analyse de données industrielles. L'objectif de ce projet est de déterminer comment intégrer les compétences en apprentissage automatique dans la formation en maintenance industrielle pour effectuer de la détection d'anomalies. Cette étude consiste à : (1) répertorier quels outils d'apprentissage automatique permettent de s'initier à la détection d'anomalies, (2) cartographier les compétences nécessaires pour exploiter ces outils, (3) proposer une démarche visant à intégrer les nouvelles compétences dans la formation des technologues et (4) établir une grille diagnostique afin d'adapter la formation selon les besoins. Cette étude est basée sur le développement d'un pipeline complet, à partir d'un système mécanique. L'objectif est de collecter des données de fonctionnement pour bâtir un jeu de données, alimenter un algorithme d'apprentissage automatique pour faire de la détection d'anomalies et faire des inférences sur le système, afin de déterminer son état de fonctionnement. Cette étude propose donc, une méthode simplifiée et peu coûteuse pour réaliser un premier projet d'apprentissage automatique en maintenance industrielle. Les résultats de ce travail visent à permettre aux étudiants de s'initier aux concepts liés à l'apprentissage automatique et d'intégrer l'utilisation des nouveaux outils dans leur pratique professionnelle. Ces nouvelles compétences ont le potentiel de moderniser le programme de TMI, de répondre aux besoins de l'industrie en ce qui a trait à l'utilisation des technologies numériques et de pallier partiellement le manque de main-d'œuvre en maintenance. L'étude consiste à collecter un jeu de données indiquant l'état de fonctionnement d'une pompe, pour faire la détection d'anomalies. La méthodologie proposée nécessite de maîtriser de nouvelles compétences pour que le technologue soit en mesure de collaborer avec des experts en analytique. Les recommandations sont regroupées dans un tableau comparatif indiquant les compétences à ajouter à la formation du technologue pour y arriver.

Mots clés : Détection d'anomalies, apprentissage automatique, maintenance industrielle, détection des défaillances, pompe, conditions de fonctionnement, signaux, séries temporelles, algorithme, industrie 4.0, internet des objets industriels, analyse de données.

Implementation of Machine Learning for Anomaly Detection in Industrial Maintenance Technology

David CÉLESTIN

ABSTRACT

Data science and artificial intelligence is becoming increasingly important in Quebec manufacturing industry. In order to address this business requirement, it is necessary to adopt a culture of data science. At the moment, the industrial maintenance program does not include core competencies in the field of data science or digital transformation. Within technologies that can have a significant impact on the role of the industrial maintenance technician, we can note internet of things, big data, and artificial intelligence since they are at the heart of connected machines. In this study, a centrifugal pump is used to build a proof of concept to test an industrial data analysis approach. The main objective of this study is to introduce machine learning core competencies in the industrial maintenance program and apply anomaly detection algorithm to a maintenance problem. This study aims to : (1) list which machine learning tools can be used as an introduction to anomaly detection, (2) map the necessary core competencies needed to use these tools, (3) develop an approach to introduce those new competencies in the industrial maintenance program, (4) develop a diagnostic chart to adapt the training program and add the required competencies to the curriculum. This study is based on the development of a complete pipeline applied to a mechanical system. The objective is to collect data from the pump to build a dataset, feed an anomaly detection algorithm with the data and run inferences on the system to determine its condition by using the data. This study proposes a simple and affordable methodology to test a proof of concept and experiment with anomaly detection for an industrial maintenance problem. The result of his work is intended to give the technicians the ability to use the concepts behind machine learning and use associated tools in their future careers. These new competencies could modernize the current industrial maintenance program and answer the business needs for integration of digital technologies and partially relief the need for maintenance technicians. In this study, a dataset that represent working conditions of a pump is collected to perform anomaly detection. The proposed methodology requires to master new machine learning core competencies for the technician to be able to cooperate with data science experts. Recommendations are grouped in a table indicating the core competencies to add to the industrial maintenance program.

Key words: Anomaly detection, machine learning, industrial maintenance, failure detection, pump, condition monitoring, signal analysis, time series, algorithm, industry 4.0, industrial internet of things, data science

TABLE DES MATIÈRES

INTRODUCTION	1
CHAPITRE 1 REVUE DE LA LITTÉRATURE	7
1.1 Introduction et stratégie de recherche	7
1.2 Pratiques de maintenance et détection des défaillances	8
1.3 Lien entre les défaillances et la détection d'anomalies	12
1.4 Surveillance des conditions de fonctionnement à l'aide de l'apprentissage automatique	15
1.5 Conclusion	23
CHAPITRE 2 MÉTHODOLOGIE	25
2.1 Introduction	25
2.2 Outils d'apprentissage automatique	26
2.3 Contexte, justification et opportunités	28
2.4 Méthodologie	29
2.4.1 Définition du besoin	29
2.4.2 Acquisition des données	30
2.4.3 Prétraitement des données	34
2.4.4 Traitement des données	36
2.4.4.1 Analyse du signal dans le domaine temporel	36
2.4.4.2 Traitement du signal vibratoire	36
2.4.4.3 Sélection des caractéristiques utiles	37
2.4.5 Exploration des données	38
2.4.5.1 Analyse et exploration des données	39
2.4.6 Préparation des données	39
2.4.7 Sélection et entraînement du modèle	40
2.4.7.1 Évaluation du modèle	41
2.4.7.2 Ajustement du modèle	41
2.4.8 Inférences et prise de décisions	41
2.5 Conclusion	41
CHAPITRE 3 CONCEPTION DU BANC DE TESTS	43
3.1 Description du banc de test	43
3.1.1 Fonctionnement de la passerelle	44
3.1.2 Caractéristiques de la pompe	45
3.1.3 Données utilisées pour l'analyse des défaillances	49
3.2 Conclusion	50
CHAPITRE 4 RÉSULTATS, DISCUSSION ET SOLUTION PROPOSÉE	53
4.1 Introduction	53
4.2 Protocole d'expérimentation	53
4.2.1 Paramètres de simulation et configuration des expériences	57
4.2.1.1 Acquisition et prétraitement des données	58

4.2.1.2	Traitement des données.....	59
4.2.1.3	Exploration des données	60
4.2.1.4	Préparation des données.....	64
4.2.1.5	Sélection et entraînement du modèle	65
4.2.1.6	Inférences à partir du modèle.....	73
4.3	Résultats de la détection d'anomalies et discussion	74
CHAPITRE 5 COMPÉTENCES NÉCESSAIRES.....		77
5.1	Introduction.....	77
5.2	Démarche proposée pour intégrer les compétences.....	80
5.3	Grille diagnostique des compétences nécessaires.....	82
CONCLUSION.....		85
ANNEXE I TABLEAU DES COMPÉTENCES EN IA POUR LES DOMAINES TECHNIQUES		89
ANNEXE II TABLEAU DES SOUS-COMPÉTENCES EN FONCTION DU PIPELINE.....		95
ANNEXE III TABLEAU DES COMPÉTENCES EN IA ASSOCIÉES AUX COURS DU PROGRAMME DE TMI AU CVM		107
ANNEXE IV GRILLE DIAGNOSTIQUE DES COMPÉTENCES NÉCESSAIRES.....		111
APPENDICES		125
LISTE DE RÉFÉRENCES BIBLIOGRAPHIQUES		126

LISTE DES TABLEAUX

	Page
Tableau 1.1 - Plan de concepts utilisé pour la revue de littérature	7
Tableau 1.2 - Principaux modes de défaillances des pompes centrifuges	10
Tableau 1.3 - Indicateurs de la condition de fonctionnement.....	11
Tableau 2.1 - Liste des capteurs utilisés	31
Tableau 2.2 - Paramètres de collecte du signal vibratoire	33
Tableau 2.3 - Exemples de données brutes de vibration.....	33
Tableau 2.4 - Exemple de données agrégées	34
Tableau 2.5 - Statistiques descriptives du jeu de données	35
Tableau 2.6 - Extrait du jeu de données utilisées.....	37
Tableau 2.7 - Architecture de l'autoencodeur utilisé	40
Tableau 3.1 - Spécification de la pompe.....	46
Tableau 3.2 - Fréquences de fonctionnement de la pompe.....	48
Tableau 3.3 - Fréquences de défauts.....	48
Tableau 3.4 - Paramètres d'analyse vibratoire.....	49
Tableau 4.1 - Modes de défaillances utilisés pour la simulation	56
Tableau 4.2 - Extrait du jeu de données de procédé	58
Tableau 4.3 - Extrait des données brutes de vibration.....	59
Tableau 4.4 - Extrait des données transformées et fusionnées	60
Tableau 4.5 - Résumé des paramètres de l'autoencodeur retenu	66
Tableau 4.6 - Rapport de classification de l'autoencodeur	68
Tableau 4.7 - Résultats obtenus pour un échantillon	73

LISTE DES FIGURES

	Page
Figure 1.1 - Étapes d'un processus PHM pour le suivi des conditions	13
Figure 1.2 - Étapes d'un processus PHM pour le pronostic	14
Figure 1.3 - Types d'algorithmes d'apprentissage automatique et leur utilisation.....	16
Figure 1.4 - Exemples d'algorithmes utilisés pour le suivi des conditions de fonctionnement Adapté de Ward et al. (2024, p. 8).....	17
Figure 2.1 - Schéma du système utilisé pour la collecte et l'analyse des données	27
Figure 2.2 - Schéma bloc des capteurs.....	31
Figure 2.3 - Exemple de graphique de la vibration.....	38
Figure 3.1 – Schéma du banc de test.....	43
Figure 3.2 - Diagramme de fonctionnement de la passerelle.....	44
Figure 3.3 - Traitement des données.....	45
Figure 3.4 – Pompe centrifuge de 1 hp,	46
Figure 3.5 - Courbe de défaillance potentielle-fonctionnelle (<i>PF curve</i>), (Traduit et adapté de (Knauer, 2024)).....	47
Figure 4.1 - Pipeline proposé pour le projet d'apprentissage automatique.....	54
Figure 4.2 - Exemple de spectre en fréquence pour l'axe z	61
Figure 4.3 - Segmentation des données	62
Figure 4.4 - Segmentation des données	63
Figure 4.5 – Aire sous la courbe précision-rappel	69
Figure 4.6 - Graphique de la perte d'entraînement.....	70
Figure 4.7 - Distribution des erreurs en fonction du seuil	71
Figure 4.8 - Matrice de confusion.....	72
Figure 5.1 - Chronologie des cours et des principales étapes	82

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

API	<i>Application Programming Interface</i> , Interfaces de programmation d'application
AMDEC	Analyse des modes de défaillance, leurs effets et la criticité
AUC	<i>Area under curve</i> , Aire sous la courbe
AutoML	Apprentissage automatique automatisé
CBM	<i>Condition-Based Maintenance</i> , maintenance basée sur la condition
CNN	<i>Convolutional Neural Network</i> , Réseau neuronal convolutif
CSV	Comma Separated Values
CVM	Cégep du Vieux Montréal
DBSCAN	<i>Density-based spatial clustering of applications with noise</i> , regroupement spatial d'applications basé sur la densité contenant du bruit
FFT	<i>Fast Fourier Transform</i> , Transformation rapide de Fourier
Fmax	Fréquence maximale à mesurer
HPF	<i>High pass filter</i> , filtre passe-haut
IDE	<i>Integrated development environment</i> , Environnement de développement
IdO	Internet des objets
IIdO	Internet Industriel des Objets
ISO	Organisation Internationale de Normalisation
LOR	<i>Lines of resolution</i> , lignes de résolution
LPF	<i>Low pass filter</i> , filtre passe-bas
MAD	<i>Median Absolute Deviation</i> , Écart Absolu Médian
MAE	<i>Mean Absolute Error</i> , Erreur Absolue Moyenne
MSE	<i>Mean square error</i> , Erreur Quadratique Moyenne

XVIII

N	Nombre d'échantillons
ODR	<i>Output Data Rate</i> , Fréquence d'échantillonnage
PCA	<i>Principal Component Analysis</i> , Analyse en Composantes Principales
PHM	<i>Prognostics and Health Management</i> , pronostique des défaillances et la surveillance
PM	<i>Predictive Maintenance</i> , maintenance prévisionnelle
RMS	Root mean square, moyenne quadratique
RNN	Recurrent Neural Network, réseau neuronal récurrent
ROC	Receiver Operating Characteristic, Caractéristique de fonctionnement du récepteur
SI	Système international
SVM	<i>Support Vector Machine</i> , machine à vecteurs de supports
t	Temps en secondes
TGM	Technique de génie mécanique
TinyML	Tiny Machine Learning, apprentissage automatique sur microcontrôleur
TMI	Technologie de maintenance industrielle
TFLM	TensorFlow Lite Micro

LISTE DES SYMBOLES ET UNITÉS DE MESURE

UNITÉS ÉLECTRIQUES

A	Ampère
V	Volt

UNITÉS MÉCANIQUES

cpm	Cycles par minute
g	Accélération gravitationnelle
gpm	Gallons par minute
hp	Horse power
NPIPA	<i>Net positive inlet pressure available,</i> Pression positive nette disponible à l'alimentation
Psi	<i>Pound per square inch,</i> livre par pouce carré
rpm	Rotations par minute

UNITÉS DE TEMPS

ms	Millisecondes
Hz	Hertz

INTRODUCTION

Contexte et motivation

La formation en Technologie de maintenance industrielle (TMI) permet de développer des compétences afin de vérifier la conformité des équipements industriels, de mettre en œuvre différents programmes d'entretien, de coordonner les activités de maintenance et de résoudre les problèmes de rendement des équipements (Compétences Québec, 2024). Le technologue issu de cette formation est donc apte à poser des diagnostics poussés et trouver les causes de problèmes sur une grande variété de machines industrielles afin qu'elles puissent fonctionner adéquatement le plus longtemps possible. De ce fait, le technologue intervient sur les machines industrielles afin de rendre l'entreprise plus performante. Au sein des entreprises québécoises, les activités de maintenance nécessitent près de 25% de l'effectif et près de 35 à 40% des coûts d'opération pour l'industrie manufacturière (ITMI, 2021). Toujours selon le rapport de l'ITMI, plusieurs facteurs exercent une pression sur les départements de maintenance des entreprises. Parmi ces facteurs, notons la disponibilité du personnel, la familiarisation avec les machines instrumentées et connectées, ainsi que la pénurie de main-d'œuvre spécialisée.

Le déploiement des technologies associées à l'industrie 4.0 doit permettre aux entreprises d'améliorer leur productivité ainsi que leur compétitivité. Ces nouvelles technologies entraînent une transformation du rôle du technologue en maintenance industrielle. Cette transformation a le potentiel de relâcher une partie de la pression subie par les entreprises, en réduisant la fréquence de certaines tâches de maintenance et en dégageant plus de temps pour la réalisation d'autres tâches par le technologue. « [...] la maintenance du futur a besoin d'une main-d'œuvre qualifiée ouverte aux nouvelles technologies, qui détient un savoir-faire avancé de la maintenance et des nouvelles technologies qui l'accompagnent désormais. » (ITMI, 2021). Parmi les technologies ayant un impact sur le rôle du technologue, on note l'internet des objets, les données massives et l'intelligence artificielle, car ces technologies sont au cœur des machines connectées. Ces technologies permettent de mettre en œuvre des approches telles que la maintenance prescriptive, visant à utiliser des données de fonctionnement afin

d'anticiper les pannes, prévenir les défaillances et améliorer les stratégies de maintenance (Amiral Technologies, 2022).

Problématique

L'utilisation des données et de l'intelligence artificielle occupe une place de plus en plus importante dans le secteur manufacturier au Québec. Pour répondre aux besoins de l'industrie, il est essentiel de mettre de l'avant une culture de collecte et d'analyse des données. Actuellement, la formation des technologues en maintenance industrielle ne comporte pas de compétences liées aux technologies numériques. Parmi les méthodes d'intelligence artificielle actuellement disponible, l'apprentissage automatique fait partie des solutions utilisées en maintenance pour déterminer les conditions de fonctionnement des machines. La détection d'anomalie représente une méthode relativement simple à mettre en œuvre. Elle permet par la suite d'introduire des concepts plus complexes et mettre en place une base solide pour des analyses plus poussées (Biggio et Kastanis, 2020). Par sa connaissance des équipements industriels, le technologue en maintenance se trouve au croisement des équipements et des technologies exploitant les données des machines. Il reste donc un pont à créer pour que le technologue soit en mesure d'utiliser ces technologies et être apte à collaborer avec les experts en analyse des données. Le technologue en maintenance possède l'expertise pour participer à l'identification des données nécessaires, participer à la collecte des données et interpréter les résultats obtenus afin de les traduire en actions concrètes. Dans cette étude, une pompe centrifuge est utilisée afin de déterminer comment mettre en œuvre un premier projet d'analyse de données industrielles. Cet équipement représente un cas d'étude intéressant, car son principe de fonctionnement est relativement simple et il s'agit d'un type d'équipement très fréquent en industrie. De plus, il existe une littérature abondante sur le suivi de la performance, l'analyse des défaillances, les bonnes pratiques de maintenance et l'utilisation de différentes techniques d'apprentissage automatique pour déterminer les conditions de fonctionnement.

Questions de recherche

1. Quels outils d'apprentissage automatique permettent de déployer rapidement la collecte et l'analyse des données, afin de détecter des défaillances?
2. Quelles sont les compétences nécessaires pour utiliser les outils d'apprentissage automatique et faire de la détection d'anomalies?
3. Quelle démarche privilégier afin de développer les compétences d'un technologue pour faire de la détection d'anomalies?
4. Comment établir une grille diagnostique pour identifier la démarche de formation adaptée au contexte du technologue en TMI?

Objectifs

L'objectif particulier de ce projet est de déterminer comment intégrer les compétences en apprentissage automatique dans la formation en maintenance industrielle. Ces nouvelles compétences ont le potentiel de moderniser le programme de TMI, de répondre aux besoins de l'industrie en ce qui a trait à l'utilisation des technologies numériques et de pallier partiellement le manque de main-d'œuvre en maintenance. Pour y parvenir, un banc de test constitué d'une pompe centrifuge fonctionnant en boucle fermée est utilisé. Un pipeline complet est développé, afin de collecter des données de fonctionnement, bâtir un jeu de données à analyser, alimenter un algorithme d'apprentissage automatique pour faire de la détection d'anomalies et faire des inférences sur les données du système, pour déterminer son état de fonctionnement.

Cette approche permet de :

- a. Répertorier quels outils d'apprentissage automatique, permettent de s'initier à la détection d'anomalies.

L'hypothèse est qu'à l'aide de ces outils, le technologue peut développer les compétences nécessaires pour participer à la mise en place d'un système de détection d'anomalies. L'étude se penche sur l'utilisation d'approches fondamentales (sélection et ajustement des modèles, choix des langages de programmation, des bibliothèques nécessaires, etc.). L'atteinte de cet objectif

permettra de répondre à la question de recherche 1. Cette question est adressée au chapitre 2.2 et vise à répertorier les outils disponibles et tester les plus intéressants avec le banc de test.

b. Cartographier les compétences nécessaires pour exploiter les outils d'apprentissage automatique.

L'atteinte de cet objectif permettra de répondre à la question de recherche 2. Cette question est abordée au chapitre 5.2 et vise à identifier les compétences à atteindre pour que le technologue soit en mesure de faire de la détection d'anomalies. Pour chaque étape du processus envisagé, les sous-compétences nécessaires sont documentées et classifiées afin de proposer un parcours d'apprentissage. L'hypothèse est que le technologue dispose de différents instruments pour analyser les défaillances et poser un diagnostic. À l'aide de ces instruments, le technologue est en mesure de choisir les bons indicateurs afin de suivre les conditions de fonctionnement des équipements et alimenter un algorithme d'apprentissage automatique.

c. Proposer une démarche visant à intégrer les nouvelles compétences au programme de maintenance industrielle.

Proposer une démarche d'apprentissage permettant de détecter des anomalies en fonction des données de capteurs. L'atteinte de cet objectif permettra de répondre à la question de recherche 3. Cette question est abordée au chapitre 5.2 et vise à déterminer comment adapter le programme de formation en TMI au Cégep du Vieux Montréal (CVM) pour intégrer des compétences en IA. L'hypothèse est que le technologue est en mesure d'utiliser la détection d'anomalies pour connaître en temps réel l'état de fonctionnement d'une machine industrielle et déterminer si une intervention est nécessaire.

d. Établir une grille diagnostique afin d'adapter la formation aux besoins spécifiques du technologue en maintenance industrielle.

L'atteinte de cet objectif permettra de répondre à la question de recherche 4. Élaborer des recommandations, présentées au chapitre 5.3, afin de proposer un outil permettant de déterminer les étapes et les outils nécessaires pour faire de l'apprentissage automatique en maintenance industrielle, cibler les compétences à ajouter au programme de formation du

technologue et proposer des activités d'apprentissage en fonction des besoins. L'hypothèse est qu'une grille d'analyse permet de cibler les compétences à développer et les outils nécessaires pour adapter la formation et développer les compétences d'un technologue en TMI.

Contributions

Cette étude propose une méthode simplifiée et peu coûteuse pour réaliser un premier projet d'apprentissage automatique en maintenance industrielle. Des données sont collectées à partir d'un équipement réel, car il s'agit d'un aspect important du rôle du technologue en maintenance. Il est ensuite en mesure de comprendre les étapes liées à la collecte, la préparation et le prétraitement des données afin de faciliter le déploiement de projets et de collaborer avec les experts en analyse des données. Grâce à cette démarche, le technologue est en mesure de dégager du temps pour réaliser des tâches à plus grande valeur ajoutée, de proposer des projets d'amélioration et d'améliorer la performance et la disponibilité des équipements dont il est responsable.

Organisation du mémoire

Ce mémoire est organisé de la manière suivante :

Introduction

Cette section présente le contexte et les motivations derrière cette étude. La problématique y est présentée, suivie des questions de recherche, des objectifs et des contributions scientifiques de ce travail de recherche.

Chapitre 1 : Revue de littérature

Le chapitre 1 recense les travaux et les technologies en lien avec la surveillance des machines, les modes de défaillance des pompes centrifuges, les principaux modes de détections des défaillances et l'utilisation de l'apprentissage automatique en maintenance. Ces travaux sont analysés afin de déterminer comment orienter la démarche de cette étude.

Chapitre 2 : Méthodologie

Ce chapitre présente le rôle du technologue en maintenance industrielle ainsi que les opportunités comparativement aux solutions existantes. Ce chapitre aborde le traitement des données, le traitement de signal, les analyses et les algorithmes utilisés en apprentissage automatique et enfin, comment le technologue se positionne comparativement aux experts des données.

Chapitre 3 : Conception du banc de test

Cette section présente l'architecture du banc de test, son fonctionnement, les systèmes utilisés et les paramètres de mesure et d'analyse.

Chapitre 4 : Résultats, discussion et solution proposée

Ce chapitre présente le protocole d'expérimentation, les paramètres et configurations d'expériences, le pipeline et l'étude de cas utilisés pour l'expérimentation.

Chapitre 5 : Compétences nécessaires

Ce chapitre présente les compétences nécessaires pour réaliser un premier projet d'apprentissage automatique. Les compétences y sont regroupées dans un tableau afin de visualiser quelles expertises sont nécessaires en fonction des étapes du pipeline.

Conclusion

La conclusion passe en revue les risques et les enjeux possibles, les avantages de la solution proposée et les pistes à explorer pour des travaux futurs.

CHAPITRE 1

REVUE DE LA LITTÉRATURE

1.1 Introduction et stratégie de recherche

Plusieurs travaux abordent l'analyse des défaillances dans le but d'améliorer la performance d'équipements industriels. Cette revue de littérature s'intéresse principalement à la détection d'anomalies pour détecter les problèmes de fonctionnement des pompes centrifuges dans un contexte de maintenance industrielle. À partir de cet énoncé, le plan de concepts, présenté au Tableau 1.1, a été élaboré pour identifier les principaux thèmes liés à la problématique et aux terminologies utilisées dans les travaux précédents.

Tableau 1.1 - Plan de concepts utilisé pour la revue de littérature

Apprentissage automatique	Détection d'anomalies	Causes des défaillances	Condition monitoring	Maintenance prescriptive	Rotary pump	Capteurs
Machine learning	Anomaly detection	Root cause analysis	Fiabilité	Maintenance prédictive	Pompe	Vibration
		Failure	Reliability	Predictive maintenance	Machine	Sensors
		Défaillance	Condition de fonctionnement		Industrial	Analyse de signaux
		Downtime	Prognostic health management		Mechanical	Signal analysis
		Fault	Condition based monitoring		Rotating machinery	Séries temporelles
		RCA	CBM		Machine tournante	Time series
			PHM			Frequency analysis

À partir du plan de concept, différentes équations de recherche ont été testées et affinées. L'équation utilisée pour l'exploration initiale des bases de données est la suivante :

```
((("machine learning" OR "anomaly detection") AND (("root cause analysis" OR "failure" OR "downtime" OR "fault") OR ("condition monitoring" OR "reliability"))) AND ("maintenance" OR "predictive maintenance") AND ("rotary pump" OR "pump") AND ("signal analysis" OR "vibration" OR "sensor"))
```

Les références ont été trouvées principalement dans les bases de données d'Engineering Village, Proquest, Science Direct et Scopus à l'aide de l'équation de recherche précédente.

L'utilisation de l'apprentissage automatique pour analyser le fonctionnement d'une pompe nécessite la maîtrise de plusieurs techniques d'analyse. Ces techniques permettent de croiser les informations obtenues et d'alimenter des algorithmes afin de prendre des décisions. La surveillance des conditions de fonctionnement est une stratégie de maintenance qui permet de suivre en continu les paramètres d'un équipement et de déterminer s'il se dégrade ou s'il est sur le point de subir une défaillance. L'apprentissage automatique permet de développer des systèmes qui apprennent à partir de données. Ces systèmes peuvent analyser une grande quantité de données et établir des liens dans le but de prendre de meilleures décisions. L'analyse de signaux permet de traiter des données temporelles et de les transformer afin d'extraire des informations et des tendances à partir des caractéristiques d'un signal.

1.2 Pratiques de maintenance et détection des défaillances

Parmi les stratégies de maintenance courantes, la maintenance basée sur la condition (CBM) et la maintenance prédictive (PM) utilisent les données pour planifier les activités de maintenance (Biggio et Kastanis, 2020). Les stratégies de CBM déclenchent un arrêt d'équipement afin de remplacer une pièce, bien que celle-ci pourrait être utilisée plus longtemps. Les stratégies de PM visent à déterminer quand planifier une intervention tout en continuant d'utiliser la pièce en question. Les systèmes de gestion basés sur le pronostic (PHM) nécessitent l'utilisation de données. Il est donc nécessaire de déterminer quels capteurs utiliser, où les installer, choisir la fréquence d'échantillonnage, utiliser une passerelle entre les capteurs

et la base de données utilisée pour la sauvegarde et des analyses basées sur l'historique en plus de visualiser les données en temps réel. Les principales méthodes de surveillance utilisées pour connaître les conditions de fonctionnement sont l'analyse vibratoire, la thermographie, la tribologie, les paramètres de procédé, les inspections visuelles et l'analyse par ultrasons (Mobley, 2002). Parmi ces technologies, l'analyse vibratoire et la thermographie sont particulièrement intéressantes, car elles sont très répandues et les technologues l'utilisent déjà pour l'analyse des machines. Ils sont donc en mesure d'utiliser leur expertise du domaine pour valider les résultats obtenus. Mobley mentionne également que la surveillance des paramètres de procédé permet d'assurer une efficacité et la productivité des équipements. Les effets négatifs associés à la perte d'efficacité peuvent surpasser les pertes associées aux coûts de maintenance (Mobley, 2002). La surveillance de l'efficacité apporte donc des avantages lorsqu'elle est associée à la surveillance des conditions de fonctionnement.

Pour une pompe, le moindre changement apporté aux éléments du système ou aux conditions du fluide aura un impact sur les variables de procédé (Mobley, 2002). Les variables de procédé à suivre sont le débit, la pression, la puissance consommée au moteur. Toujours selon Mobley, les paramètres de fonctionnement à suivre pour les pompes centrifuges, sont les forces axiales, le passage de pales et la vitesse de rotation. Les modes de défaillances des pompes centrifuges affectent directement les paramètres de procédé. Donc, en fonction du problème observé, il est possible d'identifier les différentes causes possibles. Les principaux modes de défaillance ainsi que les causes des problèmes les plus courants sont identifiés dans le Tableau 1.2. Dans ce tableau, un lien est établi pour chacune des causes identifiées et les différents problèmes pour lesquels il est possible de faire le suivi.

Tableau 1.2 - Principaux modes de défaillances des pompes centrifuges
Adapté de Mobley (2002, p. 219)

Cause	Problème												
	Pression de refoulement insuffisante	Fonctionnement intermittent	Capacité insuffisante	Aucun débit	Température élevée des paliers	Faible durée de vie des paliers	Faible durée de vie des joints mécaniques	Niveau de vibrations élevé	Niveau sonore élevé	Demande en puissance excessive	Moteur disjoncte	Température du moteur élevée	Température du fluide élevée
Arbre tordu					•	•	•	•		•			
Corps de pompe déformé par les contraintes de la tuyauterie					•	•	•	•		•		•	
Cavitation	•	•	•	•	•		•	•	•				•
Impulseur bouché	•		•	•				•		•			
Déséquilibre du moteur						•	•	•					
Problèmes électriques (moteur)						•	•	•		•	•	•	
Air entraîné (fuite d'étanchéité ou aspiration)	•	•	•					•	•			•	
Instabilité hydraulique					•	•	•	•	•				
Impulseur monté à l'envers (double aspiration)	•		•							•			
Joint mécanique défectueux						•							
Crépine partiellement obstruée	•		•					•	•				•
Débit insuffisant dans la pompe													•
Pression d'aspiration insuffisante (NPSH)	•	•	•	•				•	•				
Volume d'aspiration insuffisant	•	•	•	•	•			•	•				•
Usure interne	•		•					•		•			
Fuite dans tuyauterie, vannes, réservoirs	•		•	•									
Défauts mécaniques, roulements usés ou rouillés					•		•			•			
Désalignement					•	•	•			•		•	
Désalignement pompe-moteur							•		•	•			•
Mauvaise combinaison de pompes en série	•		•			•		•		•			
Présence de gaz dans le fluide	•	•	•					•	•			•	
Obstruction tuyauterie ou corps de pompe	•		•	•				•				•	•
Déséquilibre du rotor						•	•	•					
Masse volumique trop élevée	•									•		•	
Vitesse trop élevée										•	•		
Vitesse trop faible	•		•	•								•	
Charge totale plus élevée que prévue	•	•	•	•	•		•					•	•
Charge totale plus faible que prévue					•		•	•	•	•			•
Pompes inadaptées en parallèle	•		•	•	•			•	•		•		•
Viscosité trop élevée	•		•							•		•	
Sens de rotation incorrect	•			•						•		•	

L'*Hydraulic Institute* propose une liste d'indicateurs à suivre, afin d'anticiper et d'identifier les défaillances des pompes. Le Tableau 1.3 résume les principaux indicateurs de la condition de fonctionnement et les modes de défaillances associés (Hydraulic Institute, 2013). En résumé, les modes de défaillances peuvent être détectés en mesurant des indicateurs comme la pression, le débit, la consommation de courant, la vibration, la vitesse de rotation et la température. La variation de ces paramètres peut faire l'objet d'un suivi en continu, pour suivre leur évolution et les combiner afin de remonter à la cause fondamentale de la défaillance. Alors que le Tableau 1.2 indique comment l'expertise du technologue lui permet d'établir un lien entre les problèmes observés et leurs causes potentielles, le Tableau 1.3 indique quel paramètre suivre à l'aide de capteurs afin de détecter des défaillances et les causes potentielles.

Tableau 1.3 - Indicateurs de la condition de fonctionnement
Adapté de Hydraulic Institute (2013, p. 23)

Indicateurs	Causes	Mode de défaillance potentiel
Courant élevé	Pression excessive Frottement interne Aspiration d'un corps étranger	Bris d'accouplement Surcharge du moteur Bris d'arbre Pompe saisie
	Vitesse trop élevée	Surcharge du moteur Bris d'arbre Pompe saisie Cavitation
Courant faible	Présence d'air ou de gaz	Débit faible Vibration
	Pression d'alimentation (NPIPA) trop faible Arbre de la pompe ne tourne pas	Débit faible Vibration Cavitation Bris d'accouplement
	Vitesse trop faible	Débit faible
Vibration aux fréquences de roulements	Manque de lubrifiant ou contamination Charge excessive Vitesse excessive Fuite de courant Marques dans les roulements (Brinellage)	Marques des pistes de roulements Défauts de cage des roulements Roulement saisi

Indicateurs	Causes	Mode de défaillance potentiel
Niveaux de vibration élevés	Roulement mal installé Pression différentielle élevée Cavitation Accouplement mal installé Arbre désaxé Désalignement du rotor Résonance	Défauts de roulements Bris de roulement Rupture de joints d'étanchéité Faible débit Bruit excessif Défaut d'accouplement Défaut de roulement Bris d'accouplement Bris mécanique de la pompe
Température de roulement élevée	Charges excessives Roulement mal installé Problème de lubrification Désalignement	Marques des pistes de roulements Glissement des bagues de roulements Roulement saisi
Pression différentielle élevée	Changements dans le système ou le procédé Pompe tourne trop vite Blocage à la sortie Viscosité du fluide trop élevée	Fuites Usure des éléments en rotation Bris de roulement Courant trop élevé
Pression d'alimentation ou NPIPA trop faible	Présence d'air ou de gaz Alimentation bloquée Viscosité du fluide trop élevée Manque de liquide	Cavitation Usure des éléments en rotation Bris de roulement Rupture de joints d'étanchéité
Réduction du débit	Variation de la vitesse Usure interne Cavitation	Défaut électrique Usure des éléments en rotation Vibration
Variation de la vitesse	Défaut électrique Défaut d'accouplement	Variation du débit Courant élevé Courant faible

1.3 Lien entre les défaillances et la détection d'anomalies

L'évolution des stratégies de maintenance pousse les organisations à s'éloigner de la maintenance réactive et tendre vers la maintenance préventive et la maintenance prédictive, dans le but de prolonger la durée de vie des équipements et éviter les arrêts non planifiés en planifiant les interventions avant que les défaillances ne surviennent (Achouch *et al.*, 2022). Biggio et Kastanis présentent un processus de pronostic et de gestion de l'état des actifs (PHM) permettant de déterminer les conditions de fonctionnement à partir des données. Ils utilisent des outils statistiques et la science des données pour détecter les problèmes fonctionnement, faire du diagnostic et estimer la durée de vie résiduelle des équipements (Biggio et Kastanis,

2020). Le processus est séparé en deux phases, la première phase représente le processus de suivi des conditions de fonctionnement (CBM) tel que présenté à la Figure 1.1. Ce processus débute par l'acquisition des données à l'aide de capteurs. Les données sont utilisées afin de bâtir un historique et un tableau de bord afin de suivre le fonctionnement en temps réel. Le prétraitement des données permet de les formater et les adapter à un format compatible avec les algorithmes. L'extraction des caractéristiques est utilisée pour extraire des informations supplémentaires à partir des données brutes. Le suivi des conditions et les données historiques permettent de faire la détection d'anomalies et le diagnostic des défaillances. Les résultats obtenus sont ensuite utilisés pour générer des alertes et adapter les stratégies de maintenance.

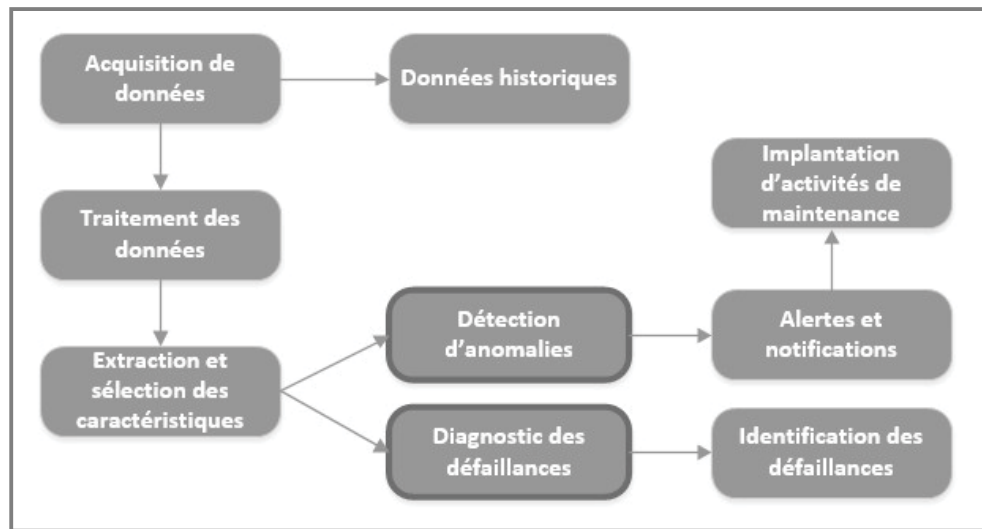


Figure 1.1 - Étapes d'un processus PHM pour le suivi des conditions

Adapté de Biggio et Kastanis (2020, p. 4)

La phase 2 du processus est présentée à la Figure 1.2. À cette étape, les résultats de la détection d'anomalies et de l'analyse des défaillances sont utilisés pour établir un pronostic et anticiper les défaillances. Il est alors possible d'estimer la durée de vie des équipements et d'ajuster les stratégies de maintenance.

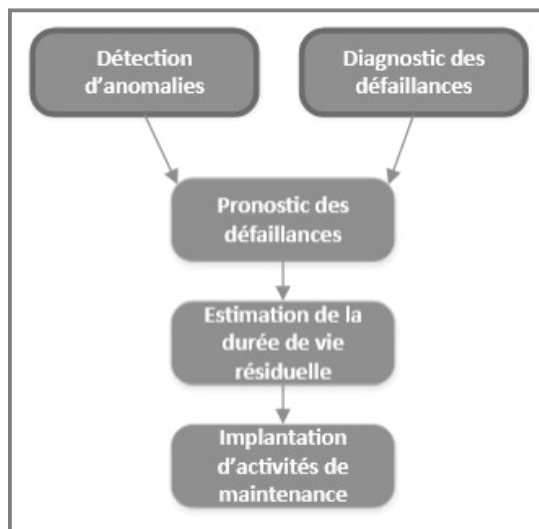


Figure 1.2 - Étapes d'un processus PHM pour le pronostic
Adapté de Biggio et Kastanis (2020, p. 4)

Les algorithmes de détection d'anomalies peuvent être entraînés à reconnaître un patron de fonctionnement normal. Lorsqu'un équipement est en opération, un modèle de détection d'anomalie est utilisé pour comparer si le patron de donnée reçu correspond à celui du fonctionnement normal. Lorsque le modèle est alimenté avec des données représentant des conditions de fonctionnement anormal, le modèle peut déterminer à quelles conditions de fonctionnement correspondent les données. Il est ensuite possible d'aller plus loin et de faire du diagnostic en déterminant si l'anomalie détectée correspond au patron d'un type de faute connu. Le diagnostic des défaillances utilise les données d'historique pour identifier la cause des anomalies détectées. Il est donc nécessaire d'avoir un historique des données liées aux défaillances à détecter afin de pouvoir classifier les données d'opération.

Comme la détection d'anomalies constitue la base des analyses subséquentes, elle permet de s'initier aux concepts d'apprentissage automatique. Ensuite, elle permet de développer des analyses plus poussées avec des algorithmes permettant de réaliser la détection de défaillances. Parmi les algorithmes utilisés pour faire de la détection d'anomalies, l'auteur dénote les réseaux de neurones. Ils permettent d'utiliser par exemple, des données vibratoires pour

déterminer la probabilité que les signaux représentent la condition normale ou une défaillance. Les machines à vecteurs de supports (SVM) à une classe peuvent être entraînées à partir de données vibratoires normales pour faire de la détection d'anomalies. Les modèles d'apprentissage profond tels que les autoencodeurs, les réseaux neuronaux convolutifs (CNN) et les réseaux neuronaux récurrents (RNN) permettent différentes applications de détection d'anomalies.

1.4 Surveillance des conditions de fonctionnement à l'aide de l'apprentissage automatique

Plusieurs travaux utilisent différentes techniques combinant l'internet des objets (IdO) et l'apprentissage automatique pour suivre les conditions de fonctionnement, améliorer la performance des équipements industriels et prédire des pannes. Ward et al. passent en revue les techniques d'apprentissage automatique utilisées pour le suivi des conditions de fonctionnement de 2019 à 2024 en comparant 31 études sur le sujet. Ils relatent les facteurs ayant contribué à l'essor de l'utilisation de l'apprentissage automatique pour le suivi des conditions de fonctionnement :

« ... l'amélioration et la réduction des coûts liés aux dispositifs et capteurs IdO »;

« ... les avancées liées aux capacités de calcul des cartes graphiques et les technologies infonuagiques. »;

« ... la disponibilité d'architectures d'apprentissage automatique à code ouvert et d'outils de traitement pour analyser les conditions de fonctionnement. ».

Ces facteurs permettent de traiter la quantité importante d'information générée lors du suivi des conditions de fonctionnement des machines. (Ward *et al.*, 2024). Ward et al. présentent les types d'algorithmes d'apprentissage automatique et leur utilisation en CBM. La Figure 1.3 montre comment les différents algorithmes sont utilisés en fonction du type de problème à résoudre. L'apprentissage supervisé est utilisé pour déterminer un résultat ou un état lorsque

les données sont identifiées, l'apprentissage non supervisé est utilisé pour catégoriser des données sans connaître préalablement le lien entre elles et l'apprentissage par renforcement est utilisé pour poser des actions et attribuer une récompense en fonction du résultat obtenu.

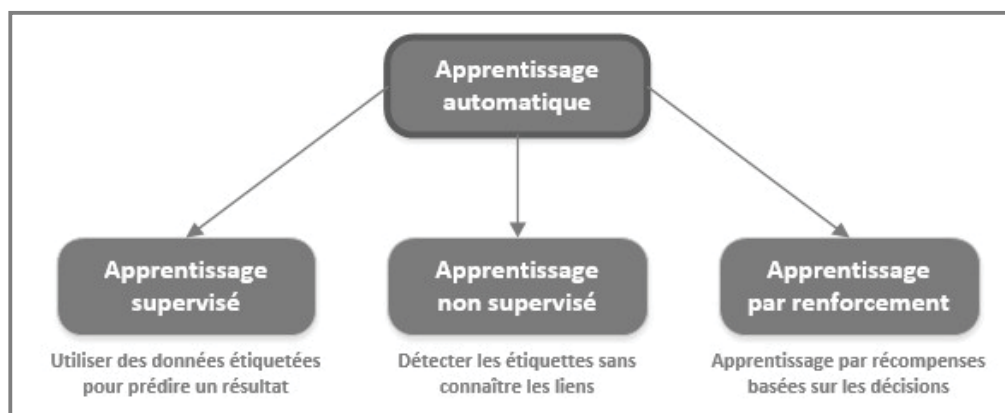


Figure 1.3 - Types d'algorithmes d'apprentissage automatique et leur utilisation
Adapté de (Ward *et al.*, 2024)p. 8)

Dans le cas de cette étude, l'apprentissage supervisé permet de faire le lien entre les données des capteurs et les défaillances préalablement identifiées afin de faire de la détection d'anomalies. L'apprentissage non supervisé utilise les approches de segmentation et de réduction des dimensions pour associer des patrons de fonctionnement aux données et effectuer des regroupements. L'apprentissage par renforcement peut permettre de développer des politiques de maintenance où le modèle pose des actions et prend des décisions en fonction du résultat. L'auteur présente également des exemples d'algorithmes utilisés en CBM. La Figure 1.4 présente les différents algorithmes regroupés par type d'apprentissage.

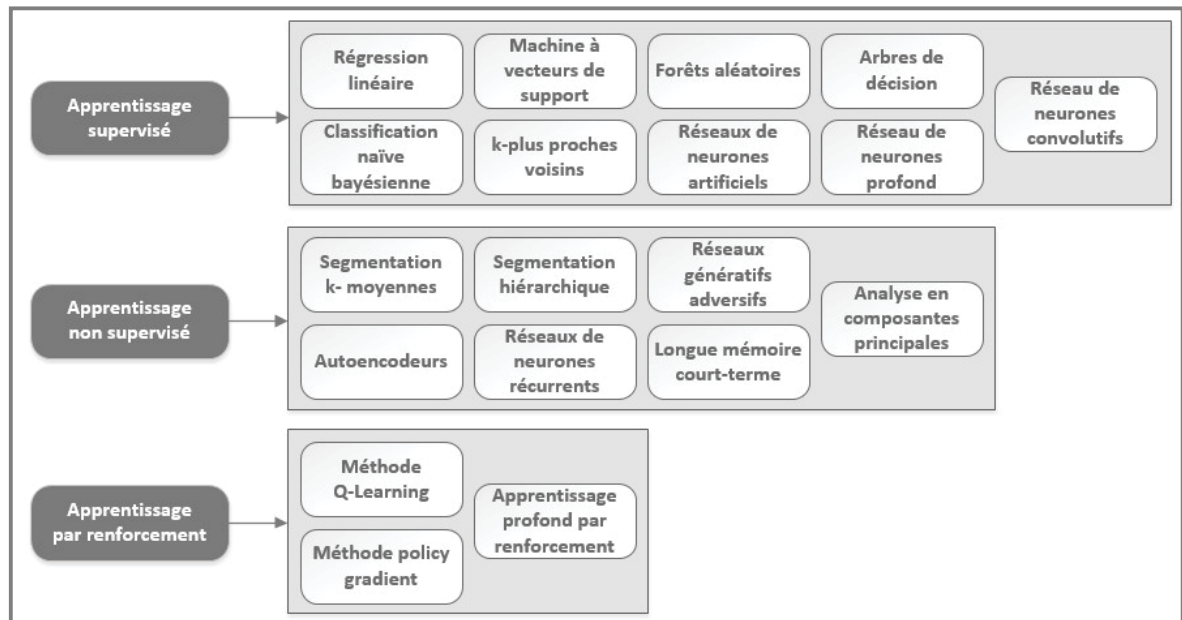


Figure 1.4 - Exemples d'algorithmes utilisés pour le suivi des conditions de fonctionnement
Adapté de Ward et al. (2024, p. 8)

Les travaux suivants ont été consultés afin de déterminer les méthodes à privilégier pour faire de la détection d'anomalie dans un contexte de maintenance et débiter un premier projet.

Mostafavi et Sadighi, Lord et Kaplan, Sai Charan ainsi que Tian et al. utilisent des autoencodeurs, ou des variantes de ce modèle, pour faire de la détection d'anomalies. Ces algorithmes sont intéressants, car ils peuvent être entraînés seulement à partir de données représentant des conditions normales. (Maasoum, Mostafavi et Sadighi, 2020) L'autoencodeur est un réseau de neurones utilisé pour la détection d'anomalie. Il apprend à reconstruire un signal d'entrée non étiqueté. En entraînant le modèle avec des données de fonctionnement normal, l'autoencodeur sera en mesure de reconstruire le signal original, alors que si on lui présente des données de fonctionnement anormal, il ne sera pas en mesure de reconstruire le signal adéquatement. L'erreur quadratique moyenne (MSE) est utilisée comme indicateur pour la détection d'anomalie. (Mostafavi et Sadighi, 2021)

Lord et Kaplan utilisent un autoencodeur et l'apprentissage automatique sur microcontrôleur (TinyML) pour faire de la détection d'anomalies, à partir de données brutes de vibration. Le

TinyML fait référence à l'utilisation de l'analytique des données sur les microcontrôleurs et autres dispositifs aux ressources de calcul et mémoire limitées et aux architectures d'apprentissage automatique automatisées (autoML). Les motivations derrière l'utilisation du TinyML sont les nombreuses applications possibles, le fait de ne pas être dépendant d'une connexion réseau, la sécurité et la confidentialité des données et les faibles coûts associés à ces technologies. L'autoML permet de créer plus facilement des modèles d'apprentissage automatique en automatisant certaines étapes du processus, ce qui permet d'entraîner des modèles sans avoir toute l'expertise nécessaire (Belcic et Stryker, 2025). Lord et Kaplan utilisent l'apprentissage non supervisé, car lorsqu'il y a plus de données normales qu'anormales disponibles, ce type d'apprentissage est très utilisé. Un accéléromètre triaxial collecte les données à un taux d'échantillonnage de 119 Hz pour détecter différentes vitesses de rotation et le débalancement. L'auteur utilise un jeu de données de vibration brute constitué de 10 292 échantillons normaux et 1 070 anormaux. Les données sont normalisées, puis il utilise l'erreur absolue moyenne (*MAE*) comme prédicteur. Il utilise un autoencodeur et calcul l'erreur de reconstruction du modèle. Finalement, le seuil de détection pour considérer une donnée comme étant une anomalie est fixé à 0.5 écart-type, autour de la moyenne des données normales (Lord et Kaplan, 2021). Dans cette étude les auteurs utilisent des données brutes de vibration et un relativement faible taux d'échantillonnage. En contrepartie, ils utilisent un nombre d'échantillons plus important et l'analyse fonctionne sur un microcontrôleur. La solution proposée dans la présente étude utilise des données vibratoires afin de croiser ces données avec les données de fonctionnement (vitesse de rotation, courant, température) et les données de procédé (pression, débit). L'accéléromètre utilisé permettra également une fréquence d'échantillonnage plus élevée.

Sai Charan ainsi que Maasoum et al. utilisent un autoencodeur pour faire de la détection d'anomalies à partir de données temporelles d'un accéléromètre. Sai Charan utilise les données brutes d'un accéléromètre triaxial, installé sur un ventilateur afin de collecter les données à une fréquence de 120 Hz. Un total de 1 800 échantillons d'une durée d'une seconde sont collectés à différentes vitesses de rotation afin de détecter un débalancement. Un autoencodeur est utilisé afin de répliquer le signal d'entrée. L'écart absolu médian (*MAD*) est utilisé pour alimenter

l'autoencodeur. Le MSE est utilisé pour déterminer l'erreur entre l'entrée et la sortie. Si le MSE est au-dessus d'un certain seuil, une anomalie est détectée. Après avoir considéré d'utiliser des caractéristiques du domaine temporel : moyenne quadratique (RMS), l'aplatissement, l'asymétrie, le facteur de crête, et la valeur crête à crêtes, l'auteur a retenu le MAD, car celui-ci performait mieux pour séparer les données normales et anormales (Sai Charan, 2022). Dans son étude, l'utilisation du MAD comme descripteur est intéressante, car les analyses nécessitent beaucoup moins de données comparativement à l'utilisation des données brutes pour différencier les données normales et anormales. Maasoum et al. utilisent un autoencodeur, car il peut être entraîné seulement à partir de données normales, contrairement à l'utilisation d'algorithmes de classification ou clustering. L'auteur utilise un disque rotatif déséquilibré pour collecter des signaux d'accéléromètres à une fréquence de 400 Hz. Il utilise 170 échantillons normaux et 210 anormaux, pour différentes vitesses de rotation et différents débalancements. Il compare les différentes caractéristiques, puis il utilise une distribution de Student pour calculer le niveau de confiance à 99%. Les valeurs au-dessus de ce seuil sont considérées comme des anomalies (Maasoum, Mostafavi et Sadighi, 2020). Dans cette étude, l'auteur utilise plusieurs descripteurs statistiques des domaines temporel et fréquentiel. Les calculs appliqués lors du traitement des données permettent de réduire drastiquement la quantité d'échantillons nécessaires pour les analyses. La fréquence de collecte des données est un peu plus élevée, mais demeure faible.

Mostafavi & Sadighi, utilisent, dans une nouvelle étude, un accéléromètre triaxial pour collecter les données d'une pompe à une fréquence de 1 600 Hz. Ils utilisent dix caractéristiques du domaine temporel, trois du domaine fréquentiel et seize du domaine temps-fréquence pour faire de la détection d'anomalie. Chaque échantillon contient 2 048 points de mesure pour chacun de trois axes (Mostafavi et Sadighi, 2021). L'auteur utilise 1000 échantillons pour les conditions normales, 2 800 pour les conditions anormales, afin d'entraîner un autoencodeur. Le MSE normalisé est utilisé avec un seuil à 3 sigmas pour identifier les anomalies. Le modèle permet de détecter les conditions à différent régime et pour différents types de défaillances. Dans cette étude, l'auteur pousse plus loin l'expérimentation précédente. La fréquence de collecte est beaucoup plus importante et il utilise toujours les

descripteurs statistiques des domaines temporels et fréquentiels. La solution qu'il propose permet de faire l'entraînement directement sur le microcontrôleur. L'architecture de l'autoencodeur proposée est un bon point de départ pour la présente étude, compte tenu des similarités des systèmes.

Costa et al. utilisent 85 capteurs de vibration installés sur une machine, pour collecter les données selon les trois axes. Un axe fictif est déterminé par la résultante des trois axes. Les signaux sont ensuite comparés dans le domaine temporel et fréquentiel. Les signaux sont collectés à 4 500 Hz durant 2 secondes. Il utilise le contenu énergétique, calculé en utilisant la valeur RMS de l'axe fictif, la distribution du contenu énergétique pour le spectre de fréquence, calculé à l'aide d'un filtre appliqué à la bande choisie et en calculant simultanément la valeur RMS pour la plage de fréquence. Pour son étude, la bande de fréquence disponible est de 2 250 Hz et il utilise trois bandes : de 0 à 500 Hz, de 500 à 1 250 et de 1 250 à 2 250 Hz, dans le but de simplifier les calculs (Costa *et al.*, 2023). L'auteur utilise la méthode d'analyse en composantes principales (PCA) comme méthode de segmentation pour différencier les modes opératoires. Il utilise différentes bandes de fréquences, pour déterminer si le moteur est en fonction et calculer l'énergie pour chaque bande d'intérêt. Il utilise le RMS-total avec un seuil pour identifier quand la machine est en fonction et travailler avec seulement ce *cluster*. Une transformation en quantiles est appliquée afin de s'assurer d'avoir une distribution normale. Une distribution normale multivariée est ensuite appliquée pour déterminer la covariance entre les modèles. Pour chaque point, une carte thermique tridimensionnelle permet d'identifier la « normalité » de ce point. Pour un nouveau point, s'il tend à devenir plus foncé avec le temps, une intervention du technologue est nécessaire.

L'auteur utilise un axe fictif pour simplifier la manipulation des données. La séparation du domaine fréquentiel en trois bandes est intéressante pour suivre des défauts se développant à différentes fréquences. Dans l'ensemble, la solution proposée reste plus complexe comparativement aux autres travaux considérés et étant donné que dans la présente étude il s'agit de proposer une méthode d'introduction, la solution retenue doit demeurer relativement simple. Pour la solution proposée dans la présente étude, un registre est utilisé pour identifier

les conditions de fonctionnement de la pompe. Le registre permet d'établir un lien entre les données et les conditions réelles de fonctionnement. À titre de comparatif, la méthode de segmentation DBSCAN est utilisée pour segmenter les modes de fonctionnement de la pompe à l'aide d'une méthode non supervisée.

La méthode DBSCAN permet d'identifier des regroupements de formes variables dans les données. Comme les données généralement disponibles ne sont pas étiquetées, l'utilisation de technique d'apprentissage automatique non supervisé renferme beaucoup de potentiel pour analyser des données réelles (Géron, 2023). En apprentissage supervisé, pour entraîner un modèle à partir des données collectées, il faut indiquer au modèle quelles données sont normales et lesquelles correspondent à des anomalies. Dans le cas de l'apprentissage non supervisé, un modèle de segmentation comme DBSCAN peut effectuer des regroupements à partir de caractéristiques semblables.

L'algorithme DBSCAN ou regroupement spatial d'applications basé sur la densité contenant du bruit vérifie pour chaque instance, combien d'autres instances se trouvent à proximité (distance epsilon). Si cette instance comporte n minimums instances dans son voisinage à distance epsilon, il est considéré faire partie du voisinage, donc appartient à ce regroupement dense ou cœur. Ces cœurs sont donc les clusters et tout ce qui n'appartient pas à un cœur est considéré comme des anomalies ou du bruit. L'algorithme DBSCAN fonctionne bien lorsque les clusters sont séparés par des régions de faible densité (Géron, 2023). Cet algorithme est intéressant, car il est simple à utiliser et performe bien avec des jeux de données relativement petits. Il utilise les paramètres epsilon pour la distance entre les instances d'un cluster et le nombre minimum d'instances nécessaire pour former un cluster.

Saha et al. explorent l'utilisation des microcontrôleurs dans les applications industrielles et les avancements en apprentissage automatique pour le déployer sur les applications en périphéries. Il présente donc les défis et opportunités liés au traitement des données de capteurs directement sur les microcontrôleurs. Selon l'auteur, les réseaux de neurones déployés à la source des données permettent de passer de la simple collecte à l'exécution d'inférence directement sur

les capteurs en faisant abstraction de l'expertise normalement nécessaire pour donner un sens aux données (Saha, Sandha et Srivastava, 2022).

L'auteur a évalué différentes solutions pour développer une solution à l'aide de TinyML. Les architectures sont organisées de façon à présenter les plateformes compatibles, le type de modèle et les bibliothèques supportées, si l'architecture utilise un code ouvert et si l'utilisation est gratuite. Les architectures suivantes sont considérées comme étant les plus intéressantes, car elles sont compatibles avec une variété de microcontrôleurs, dont ESP32 et Arduino pour lesquels beaucoup de documentation et d'exemples sont disponibles. TensorFlow Lite Micro (TFLM) permet de développer sur les microcontrôleurs basés sur ESP32 et Cortex-M. EdgeImpulseEONCompiler permet le développement complet de solutions IoT basées sur TinyML et pouvant fonctionner sur une grande variété de contrôleurs. NanoEdge AI Studio est une architecture d'apprentissage automatique qui permet d'automatiser le processus (autoML) et facilite le développement de différentes applications de ML. Elle est conçue pour fonctionner avec les microcontrôleurs STM32, mais elle est compatible avec d'autres microcontrôleurs. EloquentTinyML répond aux critères de sélection, mais moins de ressources semblent disponibles. Toujours selon l'étude, la détection d'anomalies est une application tout à fait désignée pour TinyML. Elle est en mesure de détecter les déviations d'un procédé à partir d'un flux de données pour indiquer les problèmes de fonctionnement tel que réalisé dans des applications industrielles (Saha, Sandha et Srivastava, 2022). L'utilisation du TinyML est également applicable à la réalité du technologue en maintenance. L'extraction des caractéristiques nécessite une expertise du domaine et elle permet une meilleure performance des modèles sur les microcontrôleurs en réduisant la taille du modèle. Le TinyML permet donc de débiter les premières étapes de résolutions de problèmes en donnant un sens aux données et en informant le technologue d'un changement en cours dans les paramètres de fonctionnement en se basant sur des données. Dans un environnement industriel, il est avantageux de ne pas dépendre d'une connexion réseau pour prendre des décisions ou analyser des données, en plus d'ajouter une couche de protection liée à la sécurité et la confidentialité des données. Finalement, le TinyML permet de conserver de faibles coûts et un déploiement simplifié grâce aux technologies sans fils ayant une bonne autonomie. La présente étude se

concentre sur la sélection d'un modèle simple de détection d'anomalies, car elle permet d'atteindre l'objectif de déployer un premier projet d'analytique et de comprendre toutes les phases nécessaires, ainsi que de tester le modèle et de réaliser des inférences sur des données en temps réel sans passer par le déploiement sur le microcontrôleur.

1.5 Conclusion

La détection d'anomalies est un bon point de départ pour utiliser l'apprentissage automatique dans ses activités de maintenance, car elle est relativement simple à mettre en œuvre. L'émergence de technologies d'IdO à faibles coûts et à code ouvert permet de développer des solutions rapidement pour tester la performance de modèles d'apprentissage automatique en maintenance industrielle. TensorFlow permet de tester des modèles d'apprentissage et il est supporté par de nombreuses ressources disponibles facilement. TinyML et certaines solutions d'autoML permettent de déployer des solutions locales et répondre à certains enjeux, dont la dépendance à une connexion réseau, la sécurité des données et des faibles coûts pour la mise en place. L'apprentissage supervisé permet d'anticiper des défaillances, alors que l'apprentissage non supervisé permet de détecter des défaillances ou le fonctionnement anormal. Plusieurs études utilisent les autoencodeurs pour faire de la détection d'anomalies et utilisent le MSE comme indicateur pour la détection d'anomalies à partir de données de vibration. Ce type de modèle peut être entraîné à partir des données normales exclusivement. Il ne se dégage pas de consensus pour le format des données vibratoires utilisées. Certains travaux utilisent les données brutes d'accélération, d'autres utilisent différentes transformations du domaine temporel et fréquentiel.

CHAPITRE 2

MÉTHODOLOGIE

2.1 Introduction

Ce projet vise à proposer une méthode pour développer les compétences d'un technologue en maintenance industrielle, afin qu'il soit en mesure de déployer un premier projet d'apprentissage automatique. Le pipeline proposé permet donc de faire le suivi des conditions de fonctionnement, afin de bâtir un historique représentant les conditions d'opérations. Les données collectées sont enregistrées afin de permettre au technologue de suivre l'état du système à distance. Le technologue dégage alors plus de temps, car il peut réduire les fréquences d'inspection tant que l'équipement fonctionne normalement. En cas de changement dans les conditions d'opérations, le système informe le technologue à l'aide d'une alerte, afin qu'il puisse aller vérifier le système. L'algorithme d'apprentissage automatique proposé utilise les données d'opération du système afin de caractériser les modes de fonctionnement de l'équipement. Par la suite, l'algorithme retourne la probabilité selon laquelle l'équipement fonctionne normalement ou non. En ajoutant des compétences en apprentissage automatique, le technologue est en mesure d'utiliser de nouveaux outils pour prendre de meilleures décisions, plus rapidement, grâce aux données traitées automatiquement par les algorithmes. Les nouvelles compétences doivent permettre au technologue de comprendre le lien à établir entre les équipements industriels et les algorithmes d'apprentissage automatique afin de communiquer adéquatement les données et faire l'analyse des signaux. Il doit également transposer son expertise pour l'analyse des défaillances afin de déterminer quelle information est pertinente pour alimenter le modèle et comment analyser l'information. La méthodologie proposée doit donc permettre de faire le lien entre les étapes réalisées durant le projet et les cours du programme de TMI. Les liens identifiés, permettront de déterminer comment intégrer les compétences dans les cours du programme et comment palier aux lacunes, afin d'être en mesure de comprendre le processus sans être un expert.

2.2 Outils d'apprentissage automatique

Python est un langage de programmation à code ouvert, très utilisé en science des données. Une énorme quantité d'information et de bibliothèques sont disponibles en plus de nombreux sites et ouvrages de référence pour se familiariser avec les différents concepts selon le type de besoins. Plusieurs ouvrages de référence permettent de se lancer dans l'apprentissage automatique en utilisant Python. Géron propose des exemples, dont la réalisation d'un projet de bout en bout, afin de développer toutes les compétences nécessaires, à partir des premiers pas, jusqu'à la compréhension du fonctionnement des algorithmes (Géron, 2023). Des ouvrages plus spécifiques traitent directement de concepts tels que la détection d'anomalies. Pour s'initier à la détection d'anomalies, Adari et Alla présentent des approches et des concepts importants à maîtriser (Adari et Alla, 2024). Géron utilise les bibliothèques Scikit-Learn, TensorFlow et Keras pour apprendre l'apprentissage automatique. L'auteur couvre de façon très large l'apprentissage automatique et explique le fonctionnement des algorithmes en plus d'expérimenter avec différents types de problèmes (Géron, 2023). Dans Adari et Alla, l'auteur utilise également Jupyter notebook et Scikit-Learn pour développer des exemples d'analyse de données dans un environnement Python. Dans le cas de cet ouvrage, l'auteur aborde les différents outils et approches sous l'angle de la détection d'anomalies. Les références présentées fournissent des jeux de données prêts à être utilisés pour s'initier à l'apprentissage automatique. Par contre, la préparation et le traitement des données sont des étapes très importantes et souvent les plus difficiles à réaliser dans le processus, avant même d'appliquer les techniques d'apprentissage automatique (Nielsen, 2019). Dans cette optique, la solution proposée dans cette étude permet au technologue de collecter et préparer les données d'un équipement réel afin de les utiliser pour alimenter un modèle. Les données peuvent être sauvegardées sous différents formats. Dans un souci de rester cohérent avec les méthodes d'initiation à l'apprentissage automatique et pour simplifier la manipulation des données, cette étude utilise des fichiers au format csv.

Utilisation d'un microcontrôleur

Pour débiter un premier projet à partir de données d'équipements, il est nécessaire d'utiliser une passerelle pour faire le lien entre les capteurs installés et le modèle à entraîner. La Figure 2.1 présente le schéma du système utilisé pour faire la collecte et l'analyse des données. Un microcontrôleur permet de connecter des capteurs et de générer des flux de données en temps réel. Une fois configuré, le microcontrôleur permet de simplifier les étapes de collecte, de prétraitement et de communication des données. Les données peuvent être communiquées à l'aide de différents protocoles câblés ou sans-fils. Un ordinateur est utilisé pour faire le pont avec les services infonuagiques pour accéder aux données à l'aide de l'outil Node-Red. Node-Red est un outil de développement qui permet d'établir un lien entre le flux de données, les applications et les services utilisés dans le fonctionnement des solutions d'IdO. Cet outil est utilisé pour collecter les données, les transformer selon le format souhaité et les communiquer aux différents services utilisés pour la sauvegarde, la visualisation et la communication des données. Le microcontrôleur, l'ordinateur et l'outil Node-Red constituent donc la passerelle utilisée dans cette étude pour alimenter les modèles d'analyse des données programmés en Python, à l'aide de l'outil d'apprentissage automatique TensorFlow.

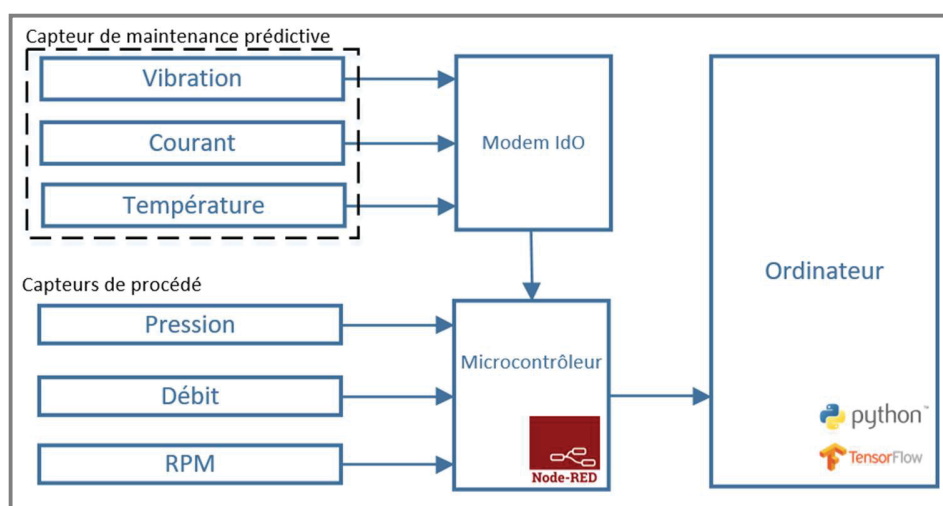


Figure 2.1 - Schéma du système utilisé pour la collecte et l'analyse des données

Cette passerelle est configurée pour enregistrer les fichiers de données brutes, afficher les données dans un tableau de bord en temps réel et sauvegarder les fichiers sur un serveur infonuagique. Cette étude se concentre seulement sur la phase de développement du modèle, car elle permet d'atteindre l'objectif de déployer un premier projet d'analytique et de comprendre toutes les phases nécessaires, ainsi que de tester le modèle dans l'environnement de développement et de réaliser des inférences sur des données en temps réel sans passer par le déploiement sur le microcontrôleur.

2.3 Contexte, justification et opportunités

L'objectif visé avec cette passerelle est d'utiliser une solution peu dispendieuse, permettant de réaliser les étapes nécessaires à la mise en œuvre de la collecte des données. Cette solution permet de valider rapidement les étapes du processus, avant de réaliser des investissements plus importants. Débuter un projet d'analyse des données industrielles nécessite un besoin clairement identifié, d'avoir accès à des données historiques et d'être en mesure de collecter les données manquantes pour les analyses. Il est également nécessaire de détenir des connaissances en analytique, d'utiliser des systèmes informatiques et de communications compatibles avec le système de collecte des données, d'utiliser les services d'une équipe de consultants externe pour le développement et l'accompagnement. Tous ces éléments sont de potentielles barrières à l'entrée pour débiter un premier projet, sans compter les coûts importants associés aux solutions disponibles et les frais mensuels. Le technologue en maintenance industrielle, par sa formation en résolution de problème, maîtrise différents outils. Ces outils lui permettent de se poser les bonnes questions afin de déterminer les actions à poser et s'attaquer à la source des problèmes au lieu de se concentrer sur les symptômes. Comme il est en contact direct avec les équipements, son rôle lui permet plus facilement de faire le lien entre les machines et les données. L'utilisation de données brutes est complexe, mais elle ajoute de la pertinence au rôle du technologue dans un projet d'analyse des données industrielle.

2.4 Méthodologie

Adari et Alla proposent une approche adaptée au niveau du technologue en maintenance industrielle pour faire de la détection d'anomalies. Le cheminement général proposé consiste aux étapes suivantes : Énoncé du besoin, collecte des données, traitement des données, séparation des données, entraînement du modèle, évaluation du modèle, ajustement du modèle (Adari et Alla, 2024).

2.4.1 Définition du besoin

Dans cette étude, la preuve de concept consiste à détecter les problèmes de fonctionnement d'une pompe centrifuge à l'aide de l'apprentissage automatique. Un exemple de cas d'application réel pourrait être de réduire la fréquence des rondes d'inspection de la pompe tant que les données de fonctionnement sont normales. Dans un contexte de pénurie de main-d'œuvre, le technologue peut alors se consacrer à d'autres tâches.

Détection d'anomalies

Durant l'opération de la pompe, on considère qu'elle fonctionne correctement et que cet état normal est l'état attendu la majorité du temps. Les données de fonctionnement de la pompe représentent donc cet état normal. Lorsque les données diffèrent de ce qui est attendu, alors on considère que l'état de fonctionnement diverge et que les données indiquent une anomalie.

Adari et Alla présentent différentes catégories d'anomalies. Les anomalies basées sur les points de mesures sont des valeurs inattendues dans un contexte d'opération normal. Dans ce cas, si la majorité des données collectées représentent le fonctionnement normal, une anomalie correspond à des données représentant un mauvais fonctionnement. Les anomalies basées sur le contexte sont des données qui pourraient être considérées normales, mais qui dans un contexte précis ne font pas de sens (Adari et Alla, 2024). La détection d'anomalies utilise donc des algorithmes basés sur des méthodes statistiques pour déterminer si en fonction des observations récentes, les données correspondent ou non aux valeurs attendues (Adari et Alla, 2024). Différentes tâches peuvent être réalisées à l'aide de la détection d'anomalies. Dans le cas de la pompe, plusieurs techniques peuvent être utilisées :

1. La détection des valeurs aberrantes est réalisée en entraînant un modèle à l'aide de données normales pour ensuite identifier les anomalies lorsqu'elles lui sont présentées.
2. La distribution des probabilités permet d'associer les anomalies aux observations ayant une très faible probabilité d'occurrence.
3. L'apprentissage d'un modèle à reconnaître des anomalies en lui apprenant ce qu'est une donnée normale et ce qu'est une donnée anormale.
4. La détection de nouveauté est similaire, mais le modèle est entraîné à l'aide de données normales et de données anormales, puis le modèle tente de catégoriser de nouvelles anomalies dans les données (Adari et Alla, 2024).

Pour cette étude, la technique 1 est utilisée. Le modèle permet de détecter les changements dans les paramètres d'opérations de la pompe. Le modèle détermine si les données fournies correspondent aux caractéristiques d'un état normal ou non et retourne un pourcentage de confiance dans la réponse fournie. À partir d'un certain seuil, la pompe est considérée dans un état de fonctionnement anormal. Une anomalie est une condition anormale, comme le blocage partiel d'une canalisation, l'aspiration d'air ou de la cavitation, de l'usure ou un déséquilibre ou encore une pièce brisée. Ce type d'analyse correspond à de la segmentation. Le modèle doit déterminer à quel point les paramètres reçus correspondent ou non aux paramètres d'apprentissage.

2.4.2 Acquisition des données

Les données collectées pour la détection d'anomalie sont les données de vibration selon les axes x, y, z, la température, le courant, la pression, le débit et la vitesse de rotation. Le Tableau 2.1 présente la liste des capteurs utilisés incluant le modèle et les principales spécifications techniques.

Tableau 2.1 - Liste des capteurs utilisés

Modèle de capteur	Mesure physique	Type de capteur	Description
NCD Industrial IoT Wireless Predictive Maintenance Sensor V3	Vibration	Accéléromètre	Accéléromètre triaxial +/- 16g, 6 400 Hz
	Courant	Transformateur de courant	Capteur de courant 100 A RMS, résolution 50 mA
	Température	Thermocouple type K	Thermocouple 0-260 °C +/- 2.2 °C
Gravity Water Pressure Sensor SEN0257	Pression	Capteur de pression pour liquide	Capteur de pression 0-1.6 Mpa 0.5%, 0-55 °C
Thermatlantic Helical turbine flow sensor HTFS-G3/4B-HFP	Débit	Capteur de débit à turbine hélicoïdale	Capteur de débit 0.83 - 83 LPM
Banner SM312LVMHS MINI-BEAM: High Speed Retro	RPM	Capteur photoélectrique	Capteur photoélectrique NPN à haute vitesse 500 Hz

La Figure 2.2 présente le schéma bloc des capteurs utilisés. Les données de maintenance (vibration, température, courant) sont collectées à l'aide d'un capteur de maintenance prédictive. Ce capteur sans fils est relié à la passerelle par un modem et est configuré à l'aide de Node-Red. Les données de procédé (pression, débit, vitesse de rotation) sont collectées à l'aide d'un microcontrôleur avant d'être envoyées à la passerelle.

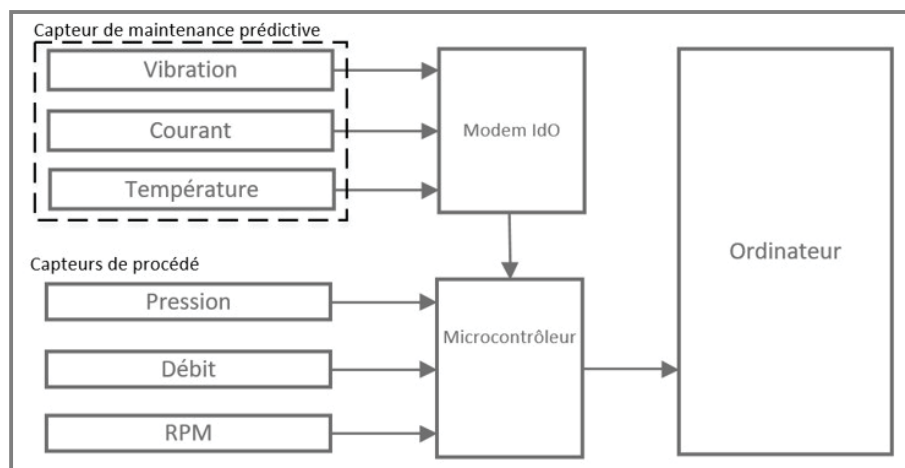


Figure 2.2 - Schéma bloc des capteurs

Données brutes de vibration

Les données brutes de vibration sont constituées d'un signal temporel pouvant être mesuré selon le déplacement, la vitesse ou l'accélération. Dans la littérature, il est généralement accepté que la mesure en vitesse est un bon indicateur pour la surveillance des machines fonctionnant à des régimes entre 600 rpm et 60 000 rpm. Pour les régimes supérieurs, l'accélération est le seul paramètre permettant d'effectuer de bonnes mesures. (Nandi et Ahmed, 2019). Le capteur de maintenance prédictive est un capteur utilisé pour des applications industrielles. Il permet d'enregistrer des mesures à des intervalles comprises entre 5 et 120 minutes. Les données sont collectées toutes les 5 minutes dans le but de bâtir un jeu de données plus rapidement. Le capteur est configuré pour collecter les données de vibration brutes et des données calculées directement par le capteur : valeurs RMS, MAX et MIN. Les données brutes sont collectées en g et les données calculées : en g pour l'accélération, en mm/s pour la vitesse et le déplacement en mm, en plus de donner les fréquences associées aux trois premiers pics pour chacun des axes. Le Tableau 2.2 présente les paramètres de collecte des données vibratoires utilisées. Les données brutes correspondent à un échantillon d'une durée de 1000 ms. Les données sont collectées à intervalle de 5 minutes, car c'est l'intervalle minimum du capteur. La fréquence de collecte est de 3200 Hz, car pour les données brutes, ce capteur permet de collecter un maximum de 3400 données par axe, pour chaque échantillon. Compte tenu de la fréquence d'échantillonnage et du théorème de Nyquist, il est possible d'analyser des fréquences allant jusqu'à 1 600 Hz. Un filtre passe-bas est appliqué à 800 Hz pour conserver les valeurs en dessous de 48 000 rpm, ce qui correspond à plus de dix fois la vitesse de rotation du moteur et pouvoir capter les fréquences de passage des pales. Un filtre passe-haut est appliqué à une fréquence de 6.25 Hz pour conserver les valeurs au-dessus de 375 rpm et éliminer les basses fréquences. La résolution en amplitude est réglée à +/- 16g ce qui est suffisant pour obtenir une résolution adéquate des mesures.

Tableau 2.2 - Paramètres de collecte du signal vibratoire

Paramètres	Valeurs
Fréquence d'échantillonnage	3200 Hz
Fréquence maximale d'analyses	1600 Hz
Résolution d'amplitude	+/- 16 g
Bande de fréquence de la FFT	10 Hz – 800 Hz
Durée de la mesure	1 s
Intervalle de collecte	5 min
Filtre passe-bas (LPF)	800 Hz
Filtre passe-haut (HPF)	6.25 Hz
Nombre de points par échantillon	3200

Le Tableau 2.3 présente dix enregistrements des données brutes de vibration pour les trois premiers points, selon les axes x, y et z. Le tableau inclut également l'horodatage utilisé comme référence pour l'identification des points de mesures.

Tableau 2.3 - Exemples de données brutes de vibration

Horodatage	ax g 1	ay g 1	az g 1	ax g 2	ay g 2	az g 2	ax g 3	ay g 3	az g 3
2025-02-17 10:09	0.0399	1.5799	1.0501	-1.2201	-1.0001	0.3001	-0.6801	-1.6301	-1.0999
2025-02-17 10:14	0.7301	0.5698	0.9198	-0.0899	-2.4602	-1.1602	0.2001	0.7798	0.0998
2025-02-17 10:19	-0.3390	-1.0697	-0.7993	-0.6490	2.0303	0.8407	-0.6390	0.2603	1.1707
2025-02-17 10:24	-0.6190	1.8397	1.0406	-0.2290	0.3397	-0.1994	-0.9490	-2.0303	-1.9294
2025-02-17 10:29	-0.4193	-0.4599	-0.5400	-0.1993	2.1801	0.6900	-0.5993	0.4201	0.1500
2025-02-17 10:34	0.0202	-0.0199	-0.0098	0.0502	-0.0099	0.0002	0.0302	-0.0099	0.0002
2025-02-17 10:39	-0.0097	0.0004	0.0102	-0.0197	0.0304	0.0202	-0.0297	0.0404	0.0302
2025-02-17 10:44	-0.0096	-0.0198	0.0003	0.0004	-0.0298	-0.0297	0.0404	-0.0098	-0.0597
2025-02-17 10:49	-0.0098	0.0003	0.0101	-0.0298	-0.0098	0.0501	-0.0298	-0.0198	0.0301
2025-02-17 10:54	-0.8888	-0.0200	-0.3794	-0.0188	0.1800	-0.1394	-0.1088	-0.5400	-0.5894

Les données de procédé sont collectées en continu et enregistrées à la demande au moment où le capteur de maintenance transmet des données. Dans ce cas, Node-Red est utilisé pour faire l'agrégation et la communication des données. Node-Red fait la demande des données de procédé et les fusionne avec les données de maintenance. Les données fusionnées sont enregistrées dans un fichier csv, puis synchronisées dans un service infonuagique pour le traitement et l'entraînement des algorithmes. Le Tableau 2.4 présente dix enregistrements des données de procédé agrégées avec les données vibratoires. L'horodatage permet de suivre

l'évolution des données dans le temps, de faire le lien entre les données et les conditions de fonctionnement durant l'opération.

Tableau 2.4 - Exemple de données agrégées

Horodatage	courant	x mm sec	y mm sec	z mm sec	pression	débit	rpm
2025-02-17 10:09	8.774	5.45	2.68	2.5	79.73	11.56	3450
2025-02-17 10:14	8.721	5.84	2.79	3.43	80.15	16.56	3435
2025-02-17 10:19	8.603	6.9	2.9	3.67	79.7	15.99	3420
2025-02-17 10:24	8.548	5.97	2.98	2.95	77.66	15.95	3420
2025-02-17 10:29	8.483	5.58	3.46	2.59	76.81	16.01	3405
2025-02-17 10:34	0	0.5	0.36	0.48	44.49	0	0
2025-02-17 10:39	0	0.37	0.55	0.51	43.65	0	0
2025-02-17 10:44	0	0.56	0.49	0.6	43.63	0	0
2025-02-17 10:49	0	0.36	0.43	0.38	43.23	0	0

Durant le fonctionnement de la pompe, un registre est utilisé pour enregistrer les événements affectant les conditions d'opérations. Un code de fonctionnement permet ensuite d'établir un lien entre les données, l'horodatage et les conditions de fonctionnement. Ces codes serviront d'étiquettes (*label*) pour les jeux de données. Les codes utilisés pour cette étude sont :

- Fonctionnement normal : 0
- Blocage partiel à l'entrée : 1
- Blocage partiel à la sortie : 2
- Aspiration d'air : 3
- Impulseur endommagé ou déséquilibré : 4
- Pièce libre ou aspiration d'un corps étranger : 5
- Impulseur usé : 6
- Pompe arrêtée : 10

2.4.3 Prétraitement des données

Le prétraitement des données consiste à assurer la qualité des données et de retirer les enregistrements contenant des valeurs aberrantes. Durant la collecte des données, plusieurs

facteurs peuvent générer des erreurs de mesures. Un capteur défectueux ou mal installé, la prise de mesure dans des conditions de fonctionnement instable, une mauvaise calibration du capteur ou une erreur dans les fonctions de transfert. À cette étape, les enregistrements contenant des données manquantes sont retirés et on s'assure d'utiliser le type de valeurs numériques approprié pour les analyses.

Les statistiques du jeu de données sont extraites afin de vérifier si des données sont manquantes, identifier les valeurs aberrantes et déterminer quelles manipulations supplémentaires sont nécessaires à la préparation des données. Les informations du jeu de données sont utilisées pour vérifier si le format est compatible pour l'utilisation avec les algorithmes. Le Tableau 2.5 présente les statistiques descriptives pour une portion du jeu de données utilisé. Le nombre de valeurs par colonne permet de vérifier s'il manque des données. La moyenne et l'écart-type permettent d'obtenir une idée générale de la distribution des données. Le min et le max permettent de détecter rapidement si des valeurs aberrantes sont présentes. Les pourcentages indiquent les percentiles et leurs valeurs correspondantes.

Tableau 2.5 - Statistiques descriptives du jeu de données

	courant	x mm_sec	y mm_sec	z mm_sec	pression	débit	rpm
compte	230	230	230	230	230	230	230
moyenne	5.546	4.411	2.966	2.159	41.118	6.678	2383
min	0.000	0.290	0.310	0.340	-1.790	0.000	0
25%	0.000	0.480	0.493	0.523	25.735	0.000	0
50%	7.652	4.800	3.675	2.575	40.985	6.960	3420
75%	8.249	6.533	4.345	2.998	55.848	10.093	3435
max	9.183	12.850	6.230	5.220	105.740	19.730	3495
std	3.723	3.214	1.810	1.199	24.512	5.710	1580

Les données de procédé sont chargées dans l'environnement Python, puis le jeu de données est nettoyé pour retirer les doublons, ajuster le nom de colonnes, convertir les valeurs numériques et supprimer les lignes contenant des valeurs manquantes ou des données erronées. Les données brutes de vibration sont chargées afin de subir un traitement similaire, puis les jeux de données sont fusionnés pour en former un seul.

2.4.4 Traitement des données

Le traitement des données constitue une étape importante à réaliser avant l'entraînement des modèles. D'abord, les descripteurs statistiques du signal vibratoire seront calculés afin d'extraire plus d'informations du signal et être en mesure de les combiner avec les données de procédé. Ensuite les données sont nettoyées, étiquetées et segmentées en fonction des conditions de fonctionnement.

2.4.4.1 Analyse du signal dans le domaine temporel

Les données brutes de vibration sont des signaux temporels collectés à haute fréquence sur une période donnée. Pour être en mesure d'appliquer la détection d'anomalies sur les données vibratoires, en même temps que sur les données de procédé, les descripteurs statistiques sont calculés afin de ramener le signal à une valeur descriptive. Différents descripteurs permettront de faire ressortir différentes caractéristiques du signal pour chaque échantillon à analyser. Nandi et Ahmed ont recensé de nombreuses études utilisant des descripteurs statistiques du domaine temporel pour le suivi des conditions de fonctionnement. Les descripteurs les plus utilisés sont le kurtosis, l'asymétrie, le facteur de forme, le facteur d'impulsion, la variance, le facteur de crête, l'amplitude crête à crête, l'amplitude rms et l'amplitude moyenne (Nandi et Ahmed, 2019).

2.4.4.2 Traitement du signal vibratoire

Jablonski propose une séquence pour le traitement du signal temporel. Ce traitement consiste à d'abord retirer la composante continue du signal afin de se concentrer seulement sur les variations du signal. La moyenne des signaux selon les trois axes est donc ramenée à zéro. Un filtre passe-haut est appliqué afin d'éliminer les plus basses fréquences et une partie du bruit, afin de se concentrer sur les fréquences au-dessus de 600 rpm selon les configurations du capteur. Ce seuil est défini par l'Organisation Internationale de Normalisation (ISO) dans la norme ISO 20816, pour les machines tournant à plus de 600 rpm. Finalement, le signal temporel est converti en vitesse en mm/s pour correspondre aux unités du système international

(SI) utilisées en analyse vibratoires (Jablonski, 2021). La mesure de vibration en vitesse rms est très utilisée comme indicateur d'état, car il mesure l'énergie vibratoire de la machine et c'est l'indicateur qui représente le mieux l'énergie émise par l'équipement (Jablonski, 2021). L'équation (2.1) est utilisée pour calculer la valeur RMS du signal pour l'ensemble des données de vibration.

$$x_{RMS} = \sqrt{\frac{1}{N} \sum_{i=1}^N |x_i|^2} \quad (2.1)$$

2.4.4.3 Sélection des caractéristiques utiles

Le jeu de données est segmenté de manière à identifier les différentes conditions de fonctionnement. À partir de la colonne d'horodatage, il est possible d'étiqueter les données correspondant aux différentes conditions de fonctionnement, selon le registre des conditions de fonctionnement. Pour la détection d'anomalies, une colonne état est ajoutée pour déterminer si l'état est normal ou anormal. Si l'état est normal (state = 0) ; si l'état est anormal (state = 1).

Le Tableau 2.6 présente une portion du jeu de donnée obtenu. Le tableau regroupe les étiquettes correspondant à l'état de fonctionnement (label), l'état pour la détection d'anomalies (state), les colonnes des données de procédé et les descripteurs statistiques obtenus à partir des données de vibration.

Tableau 2.6 - Extrait du jeu de données utilisées

courant	x mm sec	y mm sec	z mm sec	pression	débit	rpm	label	state	cluster
8.774	5.45	2.68	2.5	79.73	11.56	3450	0	0	-1
8.721	5.84	2.79	3.43	80.15	16.56	3435	0	0	-1
8.603	6.9	2.9	3.67	79.7	15.99	3420	0	0	0
8.548	5.97	2.98	2.95	77.66	15.95	3420	0	0	0
8.483	5.58	3.46	2.59	76.81	16.01	3405	0	0	1
8.553	6.22	2.85	2.58	76.16	18.12	3420	0	0	0
8.582	6.85	2.96	2.49	76.01	15.44	3420	0	0	0
8.477	6.79	3	2.91	74.14	15.68	3420	0	0	0
6.565	2.69	4.65	2.93	52.34	4.32	3480	1	1	2
6.568	3.69	3.74	2.39	52.17	4.04	3480	1	1	2

2.4.5 Exploration des données

La Figure 2.3 présente un exemple de graphique produit à partir des données brutes de vibration. Le graphique du haut présente le signal temporel en accélération et le graphique du bas présente le spectre en fréquence pour le même signal vibratoire. Dans le domaine fréquentiel, il est possible d'identifier les pics de fréquences associées aux composantes du banc de test. Cette analyse permet d'extraire plus d'information du signal et de déterminer la source de certains problèmes de fonctionnement (Shreve, 1995). Le spectre montre des amplitudes dominantes au début du spectre, entre 50 et 400 Hz. Les fréquences associées à ces pics correspondent aux fréquences de fonctionnement de l'équipement. Les amplitudes permettent de comparer le niveau associé au fonctionnement normal, comparativement à l'augmentation des amplitudes associées à la présence de défauts ou de changement dans les conditions de fonctionnement.

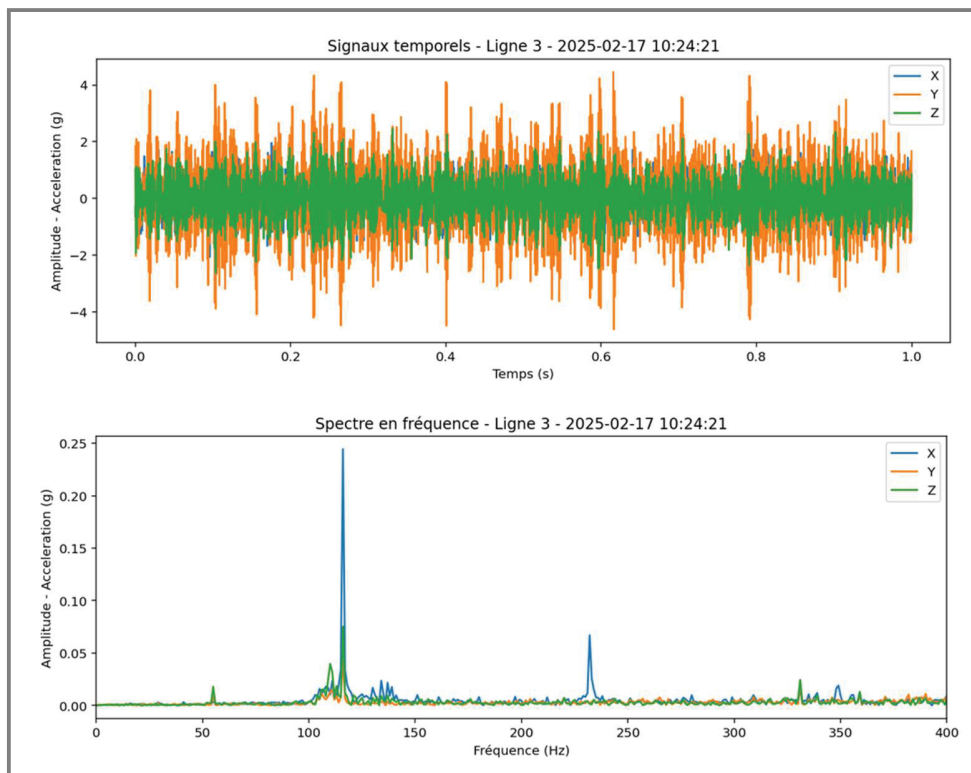


Figure 2.3 - Exemple de graphiques de la vibration dans le domaine temporel et fréquentiel

2.4.5.1 Analyse et exploration des données

L'analyse des données permet d'avoir un aperçu des lectures et de mieux comprendre le jeu de données en utilisant des statistiques et différentes méthodes de visualisation. Elle permet également de déterminer si des données sont manquantes ou si plus de données sont nécessaires. Cette étape inclut également les manipulations nécessaires pour la mise en forme adéquate des données pour alimenter les algorithmes. Comme mentionné au chapitre 2.2, les algorithmes de détection d'anomalies sont entraînés exclusivement à l'aide de données normales. Lors des tests pour déterminer la performance du modèle des données normales et anormales seront utilisées. Les données utilisent un encodage pour déterminer si l'échantillon analysé est normal = 0 ou anormal = 1. Une colonne état est donc ajoutée au jeu de donnée. Pour faire la détection d'anomalies, il est possible d'utiliser les données de procédé et les données de vibration. Les données de procédé peuvent être sélectionnées en fonction des indicateurs présentés au chapitre 1.2

2.4.6 Préparation des données

Étant donné que l'ordre de grandeur des données varie énormément en fonction des unités ou des statistiques utilisées, les données sont mises à l'échelle et normalisées. Cette étape assure que l'algorithme n'accordera pas plus d'importance à certaines données à cause de l'ordre de grandeur utilisé. La mise à l'échelle des caractéristiques permet d'uniformiser l'échelle des données pour éviter que l'algorithme soit biaisé et accorde plus d'importance aux valeurs plus grandes. Une méthode simple à utiliser est la normalisation. Elle consiste à décaler la gamme des valeurs des caractéristiques afin qu'elle se situe dans une gamme donnée, par exemple entre 0 et 1. Une autre méthode appelée standardisation permet d'uniformiser la gamme des données sans qu'elle soit limitée entre 0 et 1. Dans ce cas, la moyenne des données sera centrée à 0 et l'écart type égal à 1. Cette technique est intéressante, car le résultat est moins affecté par les données aberrantes. La mise à l'échelle est calculée pour les données d'entraînement, puis elle est appliquée aux données de validation et de test. À partir du jeu de données obtenu, les outils de science des données de Python sont utilisés pour séparer les données en jeu

d'entraînement et de validation. 80% des données sont utilisées pour l'entraînement et 20% pour la validation. Un test est ensuite effectuée à l'aide de nouvelles données.

2.4.7 Sélection et entraînement du modèle

L'autoencodeur utilisé est une version modifiée du modèle proposé dans les travaux de Maasoum et al. Ils utilisent un autoencodeur pour faire de la détection d'anomalies, car il peut être entraîné seulement à partir de données normales. Cette approche est avantageuse, contrairement à l'utilisation d'algorithmes de classification ou de segmentation qui nécessitent des données normales et anormales pour l'entraînement des modèles. L'intérêt de ce type d'algorithme est qu'il n'est pas nécessaire de connaître l'ensemble des conditions anormales à détecter (Maasoum, Mostafavi et Sadighi, 2020). Le Tableau 2.7 présente l'architecture de l'autoencodeur utilisé par Maasoum et al. La couche d'entrée reçoit les données de manière linéaire, les couches cachées un et trois utilisent des fonctions d'activations hyperboliques tangentes, alors que la deuxième couche cachée et la couche de sortie utilisent des fonctions d'activation sigmoïde.

Tableau 2.7 - Architecture de l'autoencodeur utilisé
Adapté de (Maasoum, Mostafavi et Sadighi, 2020)

	Couche d'entrée	1^{re} couche cachée	2^e couche cachée	3^e couche cachée	Couche de sortie
Nombre de neurones	29	20	15	20	29
Fonction d'activation	Linéaire	Hyperbolique tangente	Sigmoïde	Hyperbolique tangente	Sigmoïde

La détection d'anomalies est réalisée à l'aide d'un seuil de détection fixé à 95%. L'entraînement est fait avec des données normales uniquement et les tests avec des données normales et anormales.

2.4.7.1 Évaluation du modèle

À cette étape la performance du modèle est évaluée à l'aide de différentes mesures de l'erreur de prédiction. La matrice de confusion est utilisée pour déterminer la performance du modèle à effectuer des prédictions.

2.4.7.2 Ajustement du modèle

À cette étape, les hyperparamètres du modèle sont ajustés afin d'améliorer les performances du modèle. Lorsque le modèle est jugé satisfaisant, il est évalué à l'aide des données de test pour déterminer si les performances sont comparables avec de nouvelles données. L'utilisation d'un réseau de neurones offre beaucoup de flexibilité. Cette flexibilité implique que plusieurs hyperparamètres doivent être ajustés afin d'optimiser le modèle et obtenir des résultats satisfaisants pour la tâche à réaliser (Géron, 2023). L'objectif de cette étude n'est pas de développer un modèle, mais de partir d'un modèle existant et l'adapter pour fonctionner avec le jeu de données collecté pour cette étude. Afin de réaliser une tâche de détection d'anomalies, les paramètres proposés par Maasoum et al. seront donc ajustés pour qu'ils soient cohérents avec le format des données utilisés.

2.4.8 Inférences et prise de décisions

À cette dernière étape, le modèle effectue les analyses et retourne le résultat en fonction des données reçues. Le technologue doit être en mesure de décider des interventions à réaliser, mais également être en mesure de détecter des divergences entre les conditions de fonctionnement et les résultats retournés.

2.5 Conclusion

L'objectif du *pipeline* proposé est de faire ressortir les étapes et les compétences nécessaires pour détecter des anomalies à partir des conditions de fonctionnement de la pompe. Pour y parvenir, un autoencodeur est entraîné à l'aide de données normales, puis testé avec des

données normales et anormales. Le modèle doit pouvoir détecter, à partir des données dans quelle condition de fonctionnement la pompe fonctionne. L'entraînement du modèle est effectué avec des données de procédé (pression, débit, vitesse de rotation) et des données de maintenance (vibration, température, courant). Au lieu de simplement utiliser des données brutes pour effectuer l'entraînement du modèle, le croisement d'une grande variété de données est utilisé. De plus, les étapes de prétraitement et de préparation des données permettent de générer un jeu de données plus riche, contenant une grande quantité d'informations sur les conditions de fonctionnement, ceci permettant de réduire de façon importante la quantité de données nécessaire pour les analyses. La méthodologie met donc en lumière plusieurs étapes et compétences nécessaires pour intégrer l'apprentissage automatique dans le programme de TMI :

- Utiliser des algorithmes basés sur les statistiques pour analyser les données;
- Utiliser différents capteurs pour faire l'acquisition de données et caractériser le fonctionnement de la pompe;
- Analyser des signaux, des données de vibration et utiliser des séries temporelles pour structurer les données;
- Manipuler les données afin d'en faire l'agrégation et l'identification;
- Comprendre les statistiques descriptives d'un jeu de données;
- Effectuer le traitement des données pour alimenter les modèles;
- Sélectionner, entraîner et évaluer des modèles;
- Comprendre le résultat retourné par le modèle;
- Utiliser la programmation et les bibliothèques appropriées pour réaliser les différentes étapes.

CHAPITRE 3

CONCEPTION DU BANC DE TESTS

3.1 Description du banc de test

Le banc de test utilisé pour cette étude représente le fonctionnement simplifié d'une pompe industrielle. Les critères ayant mené au choix de ce système sont que la solution doit être peu dispendieuse, représenter un système industriel, être assez compacte pour tenir sur un bureau ou un chariot, avoir la possibilité de créer des défaillances ou perturber le fonctionnement normal facilement et la disponibilité de pièces de remplacement pour les pièces usées ou brisées. La Figure 3.1 présente le schéma du banc de test utilisé. Il est constitué d'une pompe centrifuge fonctionnant en boucle fermée pour faire circuler de l'eau dans son circuit. Les capteurs de pression (P) et de débit (F) permettent de collecter les données de fonctionnement du système. Différentes conditions de fonctionnement peuvent être simulées en actionnant les valves à l'entrée et à la sortie du circuit.

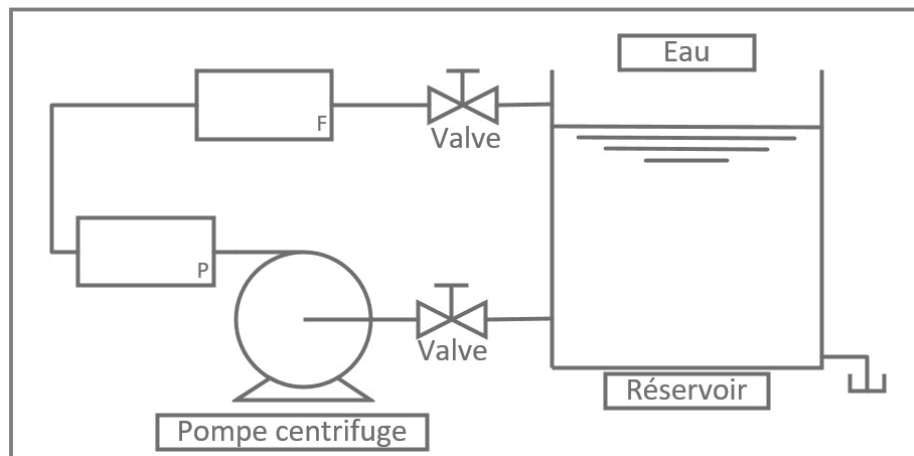


Figure 3.1 – Schéma du banc de test

3.1.1 Fonctionnement de la passerelle

La Figure 3.2 présente le diagramme de fonctionnement de la passerelle. Les données de fonctionnement sont collectées à l'aide des capteurs. La passerelle permet de collecter les données brutes, puis d'effectuer le transfert et la sauvegarde des données. La passerelle fusionne les données avant de les enregistrer dans des fichiers csv (*comma separated values*) et de les afficher dans le tableau de bord d'un service IdO. Un service infonuagique est utilisé pour accéder aux données à partir d'un ordinateur distant.

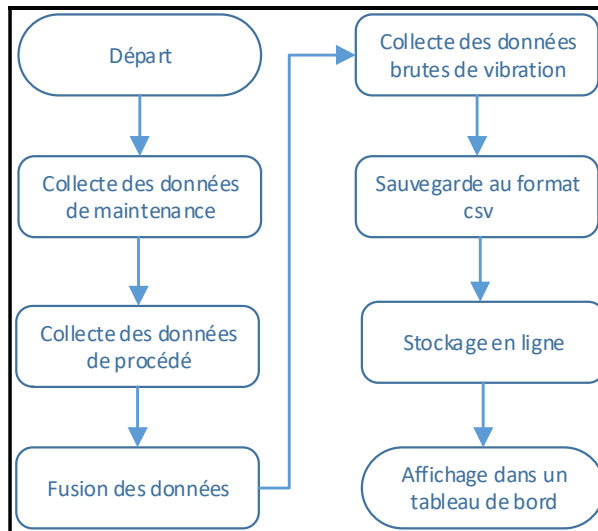


Figure 3.2 - Diagramme de fonctionnement de la passerelle

Un ordinateur est utilisé pour le traitement des données et alimenter les applications d'apprentissage automatique. La Figure 3.3 présente le diagramme de traitement des données. La phase d'entraînement est réalisée hors ligne à l'aide d'un ordinateur qui accède aux données stockées sur un serveur infonuagique. Ensuite différents algorithmes sont utilisés pour la préparation et le traitement des données, la préparation et les analyses, l'entraînement du modèle et la détection d'anomalies. Par la suite, l'ordinateur communique l'état de fonctionnement du système à l'utilisateur.

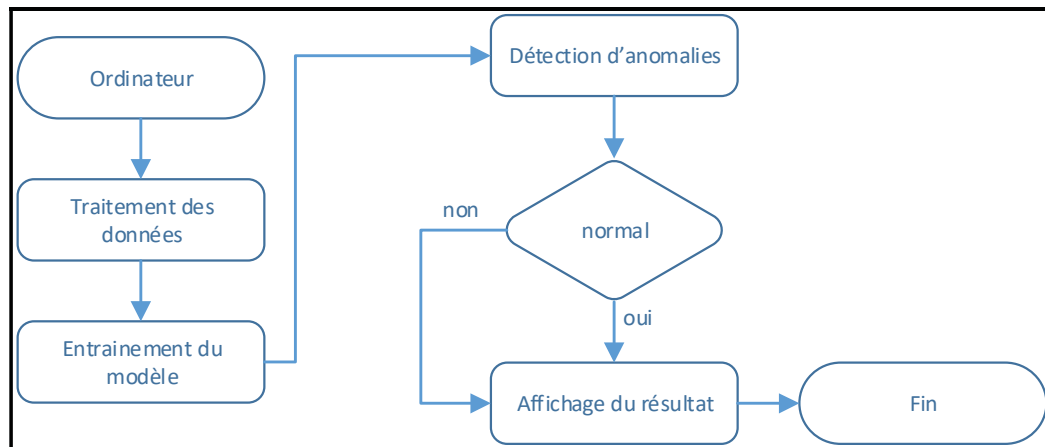


Figure 3.3 - Traitement des données

Les paramètres à mesurer, présentés au chapitre 1.2 sont la vibration, la vitesse de rotation, le courant, la température, le débit et la pression. Ces données sont enregistrées dans des fichiers csv permettant de caractériser l'état de fonctionnement du système. Un algorithme d'apprentissage automatique est ensuite entraîné pour faire la détection d'anomalies.

3.1.2 Caractéristiques de la pompe

Dans cette étude, une pompe centrifuge de 1 hp est utilisée. La Figure 3.4 montre une image de la pompe utilisée. Cette pompe centrifuge de 1 hp a été retenue, car elle représente bien les caractéristiques d'une pompe industrielle standard, c'est-à-dire qu'elle ne comporte pas de capteurs ou modules de collecte et d'analyses des données. De plus, cette pompe utilise un impulseur en plastique, ce qui permet d'accélérer l'usure de cette pièce pour la simulation de défauts. Le choix d'une pompe centrifuge est un cas idéal pour le banc de simulation, car elle fonctionne à un régime stable. Cette stabilité simplifie la collecte et l'analyse des données. Dans un contexte industriel le régime de fonctionnement, les modes et conditions d'opérations peuvent varier et dans ce cas, il serait nécessaire d'en tenir compte lors des relevés. Le Tableau 3.1 présente les spécifications techniques de la pompe. Ces spécifications sont nécessaires pour la sélection des capteurs, la configuration de la passerelle, l'analyse du signal vibratoire et la caractérisation des conditions de fonctionnement.



Figure 3.4 – Pompe centrifuge de 1 hp,
Source Princess Auto

Tableau 3.1 - Spécification de la pompe

Spécification	Unité
Tension d'alimentation	115 V / 60 Hz
Ampérage	9.5 A
Vitesse de rotation	3 475 rpm
Impulseur	6 pales
Débit max.	20 gpm
Pression max.	67 psi

Pour une analyse de la dégradation d'un équipement, les données vibratoires permettent également d'anticiper les défaillances potentielles avant que la défaillance ne se produise. À cet effet, la courbe défaillance potentielle-fonctionnelle (*PF curve*) présentée à la Figure 3.5 montre qu'à partir de l'amorce d'une défaillance, les technologies comme l'analyse vibratoire permettent d'agir dans le domaine prédictif et de détecter les premières manifestations d'une défaillance à venir, longtemps avant l'apparition de la défaillance proprement dite.

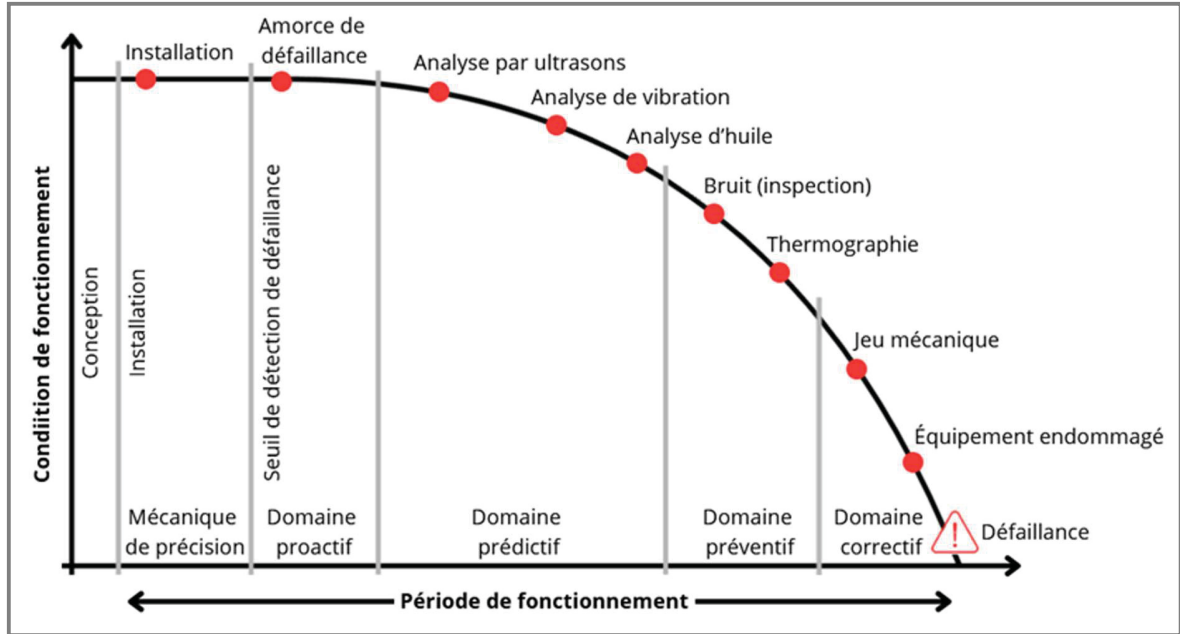


Figure 3.5 - Courbe de défaillance potentielle-fonctionnelle (*PF curve*),
(Traduit et adapté de (Knauer, 2024))

Perez explique comment utiliser l'analyse vibratoire pour suivre les conditions de fonctionnement des pompes centrifuges. Les principaux défauts menants à une augmentation des niveaux de vibration sont le déséquilibre du rotor, les défauts d'alignements entre le moteur et la pompe, les forces produites par la recirculation, le blocage ou les surcharges à l'alimentation et les pulsations liées au passage des pales (Perez, 2023). Le Tableau 3.2 présente les fréquences d'intérêts de la pompe. Ces fréquences sont utilisées pour l'analyse des signaux vibratoires. Comme les machines peuvent fonctionner à différentes vitesses de rotation, il est d'usage de nommer les multiples de la vitesse de référence en termes d'ordres. Le tableau présente donc les principales fréquences de fonctionnement selon dans les unités couramment utilisées en analyse vibratoire. La fréquence de référence est la vitesse de rotation du moteur. La fréquence de la tension électrique correspond à celle du réseau électrique et selon Perez, le moteur électrique servant à entrainer la pompe génère de la vibration à des fréquences de $2x$ la fréquence d'alimentation électrique ou $2x$ la fréquence de

glissement (Perez, 2023). Pour l'impulseur, la fréquence de fonctionnement correspond au nombre de pales multiplié par la vitesse de rotation du rotor.

Tableau 3.2 - Fréquences de fonctionnement de la pompe

Composante	Fréquence ordre	Fréquence cpm	Fréquence Hz
Rotation du moteur	1X	3 475	57.9
Tension électrique	1.036X	3 600	60
Moteur	2.072X	7200	120
Impulseur	6X	20 850	347.5

Pour la détection de défauts, les fréquences de la pompe permettent de déterminer la gamme de fréquences à suivre. Dans le cas d'une étude plus poussée, par exemple pour catégoriser les types de défaillances ou le suivi d'une composante en particulier, ces informations sont d'une grande importance. Il est donc intéressant de tenir compte de ces fréquences pour bâtir un jeu de données qui permettra des analyses plus poussées dans le futur et analyser la dégradation. Ces fréquences seront recherchées dans le spectre des fréquences et la variation de l'amplitude pourra indiquer différentes conditions de fonctionnement. Toujours selon Perez, les défauts d'accouplements se manifestent à une fréquence correspondant à 2 fois la vitesse de rotation de la pompe et est généralement accompagnée de fréquence correspondant à 1 fois la vitesse de rotation. Le débalancement du rotor produira de la vibration à 1x la vitesse de rotation, alors que le jeu mécanique produira des fréquences à 2x et 3x la vitesse de rotation. Lorsqu'un défaut est présent ou qu'il survient un changement dans les conditions de fonctionnement, l'amplitude de la vibration associée à ces défauts varie également. Le Tableau 3.3 présente les fréquences associées aux défauts courants pour les trois premiers ordres de fréquences.

Tableau 3.3 - Fréquences de défauts

Défaut	Fréquence ordre	Fréquence cpm	Fréquence Hz
Débalancement	1X	3 475	57.9
Alignement	2X	6 950	115.8
Jeu mécanique	3X	10 425	173.8

À partir des données précédentes, des spécifications des capteurs et de la théorie de l'analyse vibratoire (Shreve, 1995), les paramètres de collecte des données de vibration sont déterminés afin d'effectuer l'analyse du signal, tel que présenté au chapitre 2.4.2. Ces paramètres sont présentés dans le Tableau 3.4. Fmax correspond à la fréquence maximale à mesurer. LPF (*Low-pass filter*) est utilisé pour laisser passer seulement les fréquences en dessous de Fmax. HPF (*High-pass filter*) élimine les fréquences indésirables sous 10 Hz. LOR (*Lines of resolution*) détermine la résolution de la prise de mesures. N est le nombre de points de mesures pour chaque échantillon. ODR (*Output data rate*) est le taux d'échantillonnage du capteur. t est la durée de la prise de mesure pour un échantillon. Le nombre de moyennes représente le nombre de fenêtres utilisées pour le moyennage des spectres. La résolution en amplitude du capteur. Le type de fenêtre utilisé est Hanning. Le pourcentage de recouvrement des fenêtres.

Tableau 3.4 - Paramètres d'analyse vibratoire

Paramètres	Valeurs
Fmax	800 Hz (48 000 cpm)
LPF	800 Hz
HPF	10 Hz (600 cpm)
LOR	800 lignes
N	2048
ODR	2048 Hz
t	1 s
Moyennes	4
Résolution d'amplitude	16 bits
Fenêtre	Hanning
Recouvrement	25 %

3.1.3 Données utilisées pour l'analyse des défaillances

L'utilisation d'une AMDEC (analyse des modes de défaillance, des effets et leurs criticités) permet de structurer l'analyse des défaillances potentielles du système. Elle permet de localiser la source des défauts et la manière dont les défaillances se manifestent. Le tableau complet de l'AMDEC est présenté à l'Appendice A. Le système est décomposé en composantes. Pour chaque composante les modes de défaillances sont énumérés et les causes possibles sont

identifiées. Ensuite, les effets des défaillances sont identifiés, puis le mode de détection de la manifestation de la défaillance. Des facteurs de probabilité d'occurrence, de sévérité et de probabilité de non-détections sont déterminés dans le but de calculer l'indice de criticité. Pour chaque défaillance identifiée, l'indicateur d'état permet de déterminer le message à communiquer au technologue et les actions préventives ou correctives sont également identifiées. En fonction de la criticité, il est possible de déterminer les modes de défaillances à suivre en priorité. Dans cette étude, les principales défaillances à suivre pour la détection d'anomalies sont les modes de défaillances ayant un indice de criticité au-dessus de 100. Ce seuil a été déterminé, car ce sont les défaillances ayant la probabilité d'occurrence la plus élevée (indice $P \geq 4$), en plus d'une combinaison de sévérité et une probabilité de non-détection plus élevée (indices S et D combinés ≥ 24). Les modes de détection associées à ces défaillances indiquent les données à utiliser dans cette étude. Les cas limites ou intermittents ne sont donc pas traités dans cette étude. Ces données sont enregistrées dans des fichiers CSV. Il s'agit du débit, de la vibration et du courant. En plus de ces données, l'horodatage permet de repérer le moment où sont survenues les défaillances dans le jeu de données, la vitesse de rotation du moteur permet de détecter si la pompe est en fonction ou non. La pression et la température du moteur au niveau des roulements sont tout de même collectées pour des analyses futures.

3.2 Conclusion

Le banc de test proposé permet d'instrumenter un système représentatif d'une installation industrielle. La pompe fonctionne en circuit fermée. Elle est utilisée pour simuler différentes conditions de fonctionnement. Cette solution peu dispendieuse permet de faire de la détection d'anomalies et dans une prochaine étude, de la catégorisation de défauts. Une AMDEC est utilisé pour déterminer les données à utiliser pour cette étude. Il s'agit des données de courant, débit et vibration afin de détecter les anomalies causées par les défaillances les plus critiques. Les données collectées sont enregistrées dans des fichiers csv pour les analyses.

Le banc de test utilisé permet donc de réaliser à peu de frais toutes les étapes associées à la détection d'anomalies pour un système industriel. De plus il permet une grande flexibilité au

niveau de la collecte des données afin de bâtir rapidement un jeu de données et le même système peut être configuré pour correspondre aux besoins d'une application industrielle réelle. Les connaissances du technologue en termes d'analyse du signal vibratoire et de l'analyse des modes de défaillance, demeure nécessaire afin d'intégrer les caractéristiques du système dans les analyses et ce, afin d'obtenir rapidement des résultats utilisables pour faire de la détection d'anomalies en maintenance industrielle.

CHAPITRE 4

RÉSULTATS, DISCUSSION ET SOLUTION PROPOSÉE

4.1 Introduction

Ce chapitre présente le protocole d'expérimentation et les résultats obtenus à l'aide de la méthodologie présentée au Chapitre 2.4. Le protocole d'expérimentation proposé permet de simuler, sur une courte période, l'évolution de différentes conditions de fonctionnement d'un système industriel. La pompe fonctionne selon un cycle arrêt / départ, puis des perturbations sont introduites à travers les cycles de fonctionnement. Cette démarche vise à amener le technologue à utiliser la détection d'anomalies pour connaître l'état de fonctionnement de la pompe et déterminer si une intervention est nécessaire. Le protocole nécessite donc d'utiliser des compétences liées aux bonnes pratiques de maintenance, la maîtrise des outils ou instruments nécessaires à la détection des défaillances, mais également de procéder à l'acquisition et au traitement de données dans le but d'en extraire des caractéristiques utiles pour la détection d'anomalies. Parmi les nouvelles compétences pour le technologue en maintenance industrielle, il doit se familiariser avec le langage de programmation Python, les outils d'analyse des données et l'utilisation d'algorithmes de détection d'anomalies.

4.2 Protocole d'expérimentation

Pour cette étude, la passerelle décrite au Chapitre 2.2 est utilisée afin de faciliter la création d'un jeu de données et d'initier le technologue à l'apprentissage automatique. L'utilisation de la passerelle pour faire le lien entre les capteurs et les fichiers des données brutes (fichiers csv) contribue à simplifier de façon importante les étapes de collecte et de prétraitement des données. En effet, la passerelle est préconfigurée pour fonctionner avec des capteurs (vibration, température, courant, rpm) dont la gamme de mesures est adéquate pour une grande variété d'équipements industriels. Une fois les capteurs installés, la passerelle génère automatiquement des fichiers csv accessibles par un service infonuagique.

La Figure 4.1 présente le pipeline proposé pour cette étude. Ce pipeline est adapté de (Géron, 2023). Puisque le besoin consistant à faire de la détection d'anomalies est déjà défini, le cheminement débute avec l'acquisition des données.

- Les données sont collectées à l'aide de la passerelle, puis sont chargées au format csv;
- Le traitement des données comprend les étapes de nettoyage, fusion et augmentation des données, l'analyse de signal, la segmentation et l'étiquetage des données. Cette étape combine donc le prétraitement et le traitement des données;
- À la phase d'exploration, les statistiques descriptives des données sont analysées et le lien est établi entre les données et les paramètres physique des données;
- La préparation des données consiste à sélectionner les descripteurs les plus pertinents pour les analyses, puis séparer les données en jeux d'entraînement, de validation et de test. Les données sont ensuite échantillonnées de façon aléatoire et standardisées;
- Le modèle d'apprentissage automatique (autoencodeur) est élaboré en fonction des travaux précédents, issus de la revue de littérature. Durant l'entraînement, le modèle est ajusté en fonction des caractéristiques des données et des résultats obtenus;
- La phase d'inférence consiste à utiliser le modèle préalablement entraîné et de l'alimenter avec de nouvelles données du système pour indiquer au technologue si les conditions de fonctionnement sont normales ou anormales.



Figure 4.1 - Pipeline proposé pour le projet d'apprentissage automatique

Une pompe centrifuge est utilisée pour pomper de l'eau en boucle fermée afin de bâtir un jeu de donnée correspondant aux paramètres de fonctionnement de la pompe. Les données sont collectées toutes les 5 minutes afin de constituer un jeu de données représentant le fonctionnement normal de la pompe. Durant la collecte, de légères perturbations sont introduites durant le fonctionnement normal, afin d'ajouter du bruit dans les signaux et représenter plus fidèlement le fonctionnement de la pompe dans un contexte réel. Les

perturbations sont introduites en repositionnant manuellement l'accéléromètre, en variant le niveau d'eau dans le réservoir et en variant subtilement la position des valves durant l'opération. Pour les anomalies, des perturbations plus importantes sont introduites dans le but de simuler des problèmes de fonctionnement comparables aux modes de défaillances présentées dans l'AMDEC au Chapitre 3.1.3. Le Tableau 4.1 présente un extrait des cinq premières colonnes de l'AMDEC de l'Appendice A, pour les modes de défaillances associées à un facteur de criticité supérieur à 100, tel qu'expliqué au chapitre 3.1.3. Les modes de détection à utiliser pour détecter les défaillances les plus critiques sont le débit, la vibration et le courant.

La perte de débit causée par l'usure des pièces interne peut être simulée par un blocage partiel à l'aspiration. La vibration excessive causée par un déséquilibre de l'impulseur peut être simulée en retirant une portion de l'impulseur, suite à l'impact avec un corps étranger. Le déséquilibre du rotor du moteur ne sera pas simulé à cause des risques de dysfonctionnement mécaniques et électriques. L'augmentation du courant causé par une surcharge peut être simulée par un blocage partiel au refoulement. La défaillance causée par un roulement ou coussinet endommagé ne sera pas simulé, car elle nécessite une intervention plus importante au niveau du moteur. L'usure de l'impulseur peut être simulée par l'abrasion prématurée de l'impulseur.

Tableau 4.1 - Modes de défaillances utilisés pour la simulation

Composant	Mode de défaillance	Cause	Effet de la défaillance	Mode de détection
Pompe centrifuge	Perte de débit	Usure des pièces internes, blocage partiel à l'aspiration	Réduction du rendement, cavitation	Capteur de débit
Pompe centrifuge	Vibration excessive	Déséquilibre de l'impulseur	Dommages aux autres composantes	Capteur de vibration
Moteur électrique	Déséquilibre rotor	Usure ou défauts de montage	Vibration excessive, usure prématurée	Capteur de vibration
Moteur électrique	Courant élevé	Court-circuit, surcharge, blocage partiel au refoulement	Surchauffe, panne mécanique	Capteur de courant
Pompe centrifuge	Roulement ou coussinet usé	Frottement, lubrification insuffisante, corrosion	Bruit anormal, vibrations, usure rapide	Capteur de vibration
Pompe centrifuge	Impulseur usé	Érosion, cavitation	Réduction du débit, Réduction du rendement	Capteur de débit

Lorsque des perturbations sont introduites dans le système, le moment et la durée de ces perturbations sont notés dans un registre afin d'identifier manuellement dans le jeu de données les données normales et anormales. Lors de la préparation des données, des étiquettes seront appliquées à chaque enregistrement pour déterminer l'état de fonctionnement du système. Les étiquettes utilisées correspondent à un code de fonctionnement / défaillance numérique :

- Fonctionnement normal : 0
- Blocage partiel à l'aspiration : 1
- Blocage partiel au refoulement: 2
- Aspiration d'air (cavitation) : 3
- Impulseur endommagé ou déséquilibré : 4
- Pièce libre ou aspiration d'un corps étranger : 5
- Impulseur usé : 6
- Pompe arrêtée : 10

Comme cette étude s'intéresse uniquement à la détection d'anomalies, les étiquettes associées aux codes de défaillances (codes 1 à 6) sont regroupées dans un seul groupe état. Les données associées à la pompe arrêtée sont retirées, car ils n'apportent aucune information pertinente à l'analyse. Les codes d'états résultants sont donc :

- État normal : 0
- État anormal (anomalie) : 1

L'expérimentation consiste donc à utiliser un autoencodeur basé sur celui présenté au Chapitre 2.4.7 pour identifier correctement l'état normal ou anormal (condition d'anomalie) en analysant les données des capteurs correspondants aux modes de détection des défaillances identifiés au Tableau 4.1. La détection d'anomalies représente un cas d'étude plus simple que la catégorisation de défauts. Donc, l'information retournée par les capteurs utilisés implique qu'il y aura de la redondance dans l'information collectée, mais elle reste pertinente pour des études plus poussées. L'expérimentation doit permettre de dégager une méthode permettant à un technologue en maintenance industrielle d'utiliser l'apprentissage automatique pour détecter les problèmes de fonctionnement de la pompe et ainsi l'aider dans son travail.

4.2.1 Paramètres de simulation et configuration des expériences

L'utilisation d'algorithmes pour faire le suivi des conditions de fonctionnement dans un contexte de maintenance prédictive doit permettre de connaître les conditions d'opération des équipements, diagnostiquer la cause des défaillances et anticiper les bris d'équipements. Investir dans ce type de technologie doit permettre aux équipes de maintenance d'être plus performantes, notamment en rendant possible une meilleure planification des interventions, une meilleure gestion d'inventaire, une réduction des temps d'arrêt et une amélioration de l'efficacité des opérations. La stratégie proposée pour connaître les conditions de fonctionnement de la pompe consiste à faire de la détection d'anomalies pour déterminer si le fonctionnement de la pompe est normal ou anormal. La solution proposée pour initier le

technologie à l'apprentissage automatique est issue de la méthodologie présentée au Chapitre 2.4. et les étapes sont présentées dans les sections ci-dessous.

4.2.1.1 Acquisition et prétraitement des données

La passerelle permet au technologue de collecter des données et de générer des fichiers facilement pour créer un jeu de données. Les fichiers générés à l'aide de la passerelle sont récupérés à l'aide d'un script en Python, puis elles sont chargées dans l'environnement de développement.

Les données de procédé sont nettoyées pour s'assurer de la qualité et l'uniformité des données, les enregistrements contenant des données manquantes sont retirés et on s'assure d'utiliser le type de valeurs numériques approprié pour les analyses. Le Tableau 4.2 présente un extrait du jeu de données de procédé préalablement nettoyé à l'aide d'un script en Python.

Tableau 4.2 - Extrait du jeu de données de procédé

Horodatage	courant	pression	débit	rpm
2025-02-22 14:43	7.608	58.03	8.86	3450
2025-02-22 14:48	7.491	57.39	10.36	3450
2025-02-22 14:53	7.56	57.64	8.06	3435
2025-02-22 14:58	7.503	57.22	8.49	3435
2025-02-22 15:03	7.529	57.47	8.87	3435
2025-02-22 15:08	7.538	56.95	7.6	3450
2025-02-22 15:13	7.562	23.98	8.56	3435

Les données brutes de vibration sont regroupées dans un seul fichier pour être en mesure de faire correspondre l'horodatage avec celles des données de procédés. Le Tableau 4.3 présente un extrait du jeu de données de vibration utilisé.

Tableau 4.3 - Extrait des données brutes de vibration

Horodatage	ax_g 1	ay_g 1	az_g 1	ax_g 2	ay_g 2	az_g 2	ax_g 3	ay_g 3	az_g 3
2025-02-22 14:43	0.9097	0.5097	-0.8007	0.1097	1.6997	1.4493	0.2797	-0.3303	1.3793
2025-02-22 14:48	-0.1489	-0.7695	0.0705	1.2211	1.8805	0.3705	0.2811	1.1405	-0.1095
2025-02-22 14:53	-0.9298	-2.9892	-1.0702	-0.8498	-2.9292	-2.2802	-0.5498	2.5408	0.0598
2025-02-22 14:58	0.0906	1.7693	0.4108	0.3906	2.0093	0.5608	-0.3394	-1.5007	-0.6892
2025-02-22 15:03	0.6194	2.7699	1.2699	-0.6006	0.0399	-0.1501	-0.8506	-1.9501	-0.9401
2025-02-22 15:08	-0.6586	-1.0089	0.0001	-0.8686	-2.4489	-0.7099	-0.1886	0.2211	-0.3199
2025-02-22 15:13	-0.5401	-3.1296	-1.1602	-0.7401	-2.1896	-1.5002	0.7699	2.8404	0.9798

À cette étape, si des lignes contiennent des erreurs, elles sont retirées. Les erreurs peuvent être dues à des erreurs de lectures de capteurs ou des mesures prises dans des conditions transitoires. Finalement, les jeux de données de procédé et de vibration sont sauvegardés séparément pour générer les descripteurs statistiques des signaux vibratoires.

Pour l'étape d'acquisition de données, le technologue doit donc installer les capteurs reliés à la passerelle, récupérer les fichiers du jeu de données et exécuter le script de prétraitement. Des connaissances de base en analyse de signal, en programmation Python et des bibliothèques d'analyses de données sont nécessaires.

4.2.1.2 Traitement des données

Une fois les données chargées dans l'environnement de développement, elles sont organisées selon le format requis par les algorithmes. L'algorithme de traitement du signal vibratoire est appliqué aux données, tel que présenté au Chapitre 2.4.4.2. Cet algorithme permet de convertir les données dans les bonnes unités, calculer les descripteurs statistiques du signal vibratoire puis les jeux de données sont fusionnés. Les enregistrements du jeu de données obtenu sont étiquetés selon les conditions de fonctionnement notées dans le registre lors des opérations et les données sont séparées en jeux de données normales et anormales.

Comme expliqué au Chapitre 2.4.4.2, la valeur rms du signal vibratoire mesuré en vitesse est le descripteur d'état qui représente le mieux l'énergie émise par l'équipement. La valeur rms

du signal vibratoire est donc sélectionnée pour être fusionnée avec les données de procédé et réaliser les analyses. Le Tableau 4.4 présente un extrait du jeu de données suite à l'application de ces étapes.

Tableau 4.4 - Extrait des données transformées et fusionnées

Horodatage	courant	pression	débit	rpm	x_rms	y_rms	z_rms	étiquette
2025-02-22 14:43	7.61	58.03	8.86	3450	4.035	4.758	3.153	0
2025-02-22 14:48	7.49	57.39	10.36	3450	3.847	4.357	2.836	0
2025-02-22 14:53	7.56	57.64	8.06	3435	3.902	4.602	2.948	0
2025-02-22 14:58	7.50	57.22	8.49	3435	3.818	4.244	2.698	0
2025-02-22 15:03	7.53	57.47	8.87	3435	3.724	4.405	2.812	0
2025-02-22 15:08	7.54	56.95	7.60	3450	3.945	4.441	2.775	0
2025-02-22 15:13	7.56	23.98	8.56	3435	3.604	3.970	2.619	0

À cette étape, le technologue doit maîtriser les techniques de traitement et de préparation des données, afin de les structurer adéquatement pour les manipulations subséquentes et reconnaître les caractéristiques et le type de données afin d'appliquer les bons traitements.

4.2.1.3 Exploration des données

Un script en Python est utilisé pour afficher les données dans des graphiques et visualiser les différents enregistrements. Dans le cas des données de vibration, le spectre permet de visualiser rapidement les amplitudes associées aux principales fréquences du système. La Figure 4.2 présente le spectre en fréquence selon l'axe z d'un enregistrement représentant la pompe en fonction. Les quatre premières crêtes correspondent aux fréquences attendues pour la pompe, soit 55 Hz la fréquence associée à la rotation de la pompe, 110 Hz la fréquence associée à 2 fois la fréquence de tension électrique (Perez, 2023), 116 Hz la fréquence associée au fonctionnement du moteur électrique et 331 Hz la fréquence associée au passage des pales de l'impulseur. Le technologue est donc en mesure de vérifier si les données vibratoires sont valides pour entraîner les modèles.

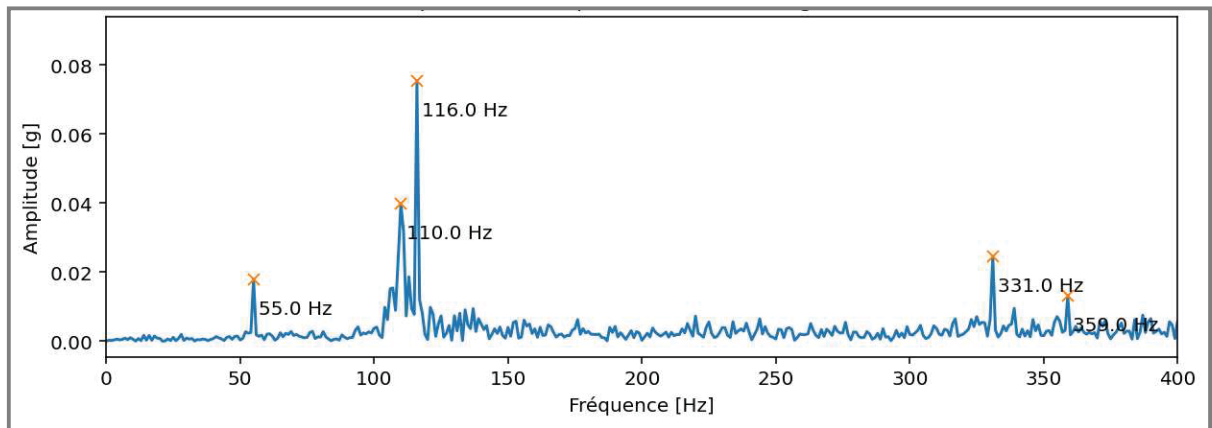


Figure 4.2 - Exemple de spectre en fréquence pour l'axe z

Le script permet également de consulter les caractéristiques des données dans un nuage de points afin de visualiser les données en fonction de la segmentation. La Figure 4.3 présente le nuage de points du jeu de données utilisées. Comme les données utilisées sont étiquetées, nous pouvons utiliser les étiquettes pour visualiser la segmentation des données et identifier les conditions de fonctionnement en fonction des caractéristiques retenues. Cette approche permet de déterminer visuellement si les conditions de fonctionnement sont facilement séparables en fonction des descripteurs sélectionnés. Dans les conditions où les classes sont moins bien séparées, il est possible d'utiliser d'autres méthodes de séparation non-linéaires, comme les machines à vecteurs de support ou SVM.

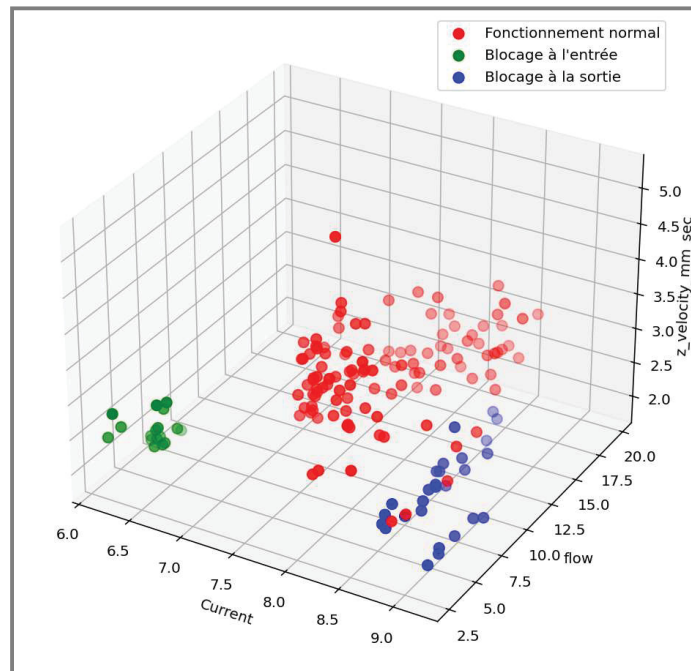


Figure 4.3 - Segmentation des données
en fonction des étiquettes

Dans un contexte réel, la segmentation peut être automatisée à l'aide de méthodes telles que l'analyse en composantes principales (PCA) ou la méthode DBSCAN. La Figure 4.4 présente les résultats obtenus à l'aide de la méthode DBSCAN pour le même jeu de données. La méthode DBSCAN est appliquée sur des données préalablement mises à l'échelle. Ensuite les hyperparamètres sont ajustés en fonction des données. Le jeu de données étant très petit et sachant qu'il y a trois conditions à détecter, les hyperparamètres sont ajustés de manière à obtenir 3 regroupements et limiter le nombre d'enregistrements n'appartenant à aucun regroupement (bruit). Le nombre minimum de points pour constituer un regroupement est établi à 6 points, soit deux fois le nombre de regroupements recherchés (Datacamp, 2024). Pour constituer un regroupement, les points ne doivent pas être espacés d'une distance (pas) de 0.7. Cette valeur est obtenue à l'aide d'un graphique de la distance aux k plus proches voisins. Les mêmes regroupements sont identifiables et on note l'apparition de bruit pour les données qui n'ont pu être catégorisées en fonction de cette méthode de segmentation.

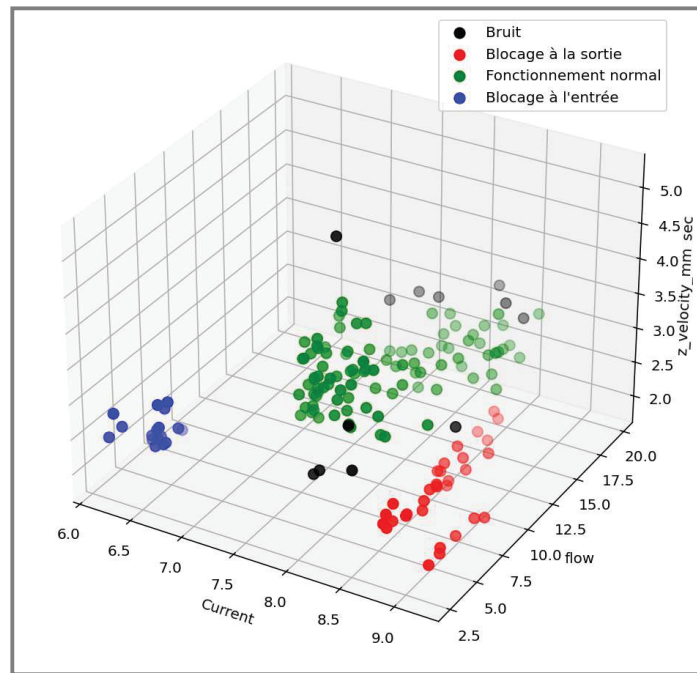


Figure 4.4 - Segmentation des données
à l'aide de DBSCAN

L'exploration des données permet d'analyser individuellement les caractéristiques de chaque enregistrement. Chaque enregistrement représente donc un condensé des informations que le technologue est amené à analyser lors de l'établissement d'un diagnostic. En comparant les données d'un enregistrement par rapport à une référence ou une tendance historique, le technologue est en mesure de déterminer si l'équipement fonctionne adéquatement et si une intervention ou une surveillance est nécessaire. Les bénéfices attendus de la détection d'anomalies sont, d'analyser rapidement une grande quantité d'information (enregistrements) contenant de nombreuses caractéristiques. Il serait difficile pour un technologue de suivre facilement une aussi grande quantité d'information.

Pour cette étape, le technologue doit maîtriser les compétences nécessaires pour utiliser les bons outils de visualisation des données et les descripteurs statistiques appropriés pour explorer les données et analyser les caractéristiques importantes du jeu de données.

4.2.1.4 Préparation des données

L'étape de préparation vise à préparer les caractéristiques des données afin d'alimenter le modèle. Cette étape sert également à déterminer quelles caractéristiques permettent de mieux segmenter les données normales et les données anormales. Les données sont d'abord séparées en jeux d'entraînement (80 % des données normales seulement), de validation (10% des données normales) et de test (10% des données normales et toutes les données d'anomalies). La séparation est réalisée par un échantillonnage aléatoire afin de reconnaître l'information contenue dans les données et non dans la séquence temporelle. La standardisation est réalisée à l'aide des données d'entraînement, puis elle est appliquée aux jeux de données de calibration et de test.

Les caractéristiques à analyser ont été sélectionnées à l'aide de l'AMDEC et à partir des bonnes pratiques d'analyse vibratoires. Ensuite, les données sont séparées en jeu de données normales et données représentant des anomalies. Les données sont standardisées à l'aide de la méthode StandardScaler. Cette technique permet de standardiser les caractéristiques des données en ramenant la moyenne des données à zéro, et l'écart-type à 1. Cette technique est intéressante dans le cas de cette étude, car le résultat est moins affecté par les anomalies.. Pour l'entraînement, seulement les données normales sont utilisées. Un échantillon est prélevé et le reste est conservé pour la validation et le jeu de test. Pour le jeu de test, les données normales sont mélangées avec les données d'anomalies pour tester le modèle après l'entraînement et la validation.

Pour le prétraitement des données, le technologue doit pouvoir différencier le mode de fonctionnement des algorithmes d'apprentissage automatique pour comprendre le traitement approprié à appliquer aux données et la nécessité d'utiliser des jeux d'entraînement, de tests et de validation.

4.2.1.5 Sélection et entraînement du modèle

Pour cette étude, un autoencodeur est utilisé pour faire de la détection d'anomalies. Il s'agit d'un bon type d'analyse pour débiter rapidement un premier projet d'apprentissage automatique en maintenance industrielle. Avec l'expérience acquise et un historique de données pertinent, les fondements acquis peuvent permettre d'aller plus loin et faire de la segmentation de défauts ou des prédictions afin d'anticiper les défaillances à venir.

La sélection d'un modèle approprié, son paramétrage et son ajustement relèvent de compétences au-delà des attentes pour un technologue en maintenance industrielle. Une connaissance de base des modèles utilisés pour des applications en maintenance est pertinente pour la collaboration avec les équipes d'experts en science des données. Aussi, l'évolution rapide des solutions d'autoML comme NanoEdge AI Studio et les solutions sans codage comme EdgeImpulseEONCompiler peuvent contribuer à rendre accessible aux technologues des outils lui permettant de sélectionner des modèles adaptés à différents types d'analyses.

Le modèle d'apprentissage automatique (autoencodeur) est élaboré à partir des travaux de (Maasoum, Mostafavi et Sadighi, 2020), tel que présenté au chapitre 2.4.7. La couche d'entrée comporte 5 neurones afin de correspondre à la dimension des données obtenues des capteurs. Le nombre de neurones des couches cachées est ajusté afin d'obtenir une compression et une reconstruction des données et une régularisation L2 est appliquée aux couches cachées, afin d'éviter un sur-apprentissage des données. Cette fonction ajoute une pénalité égale à la somme des carrés des coefficients à la fonction de perte, afin de stabiliser le modèle rapidement. Le Tableau 4.5 présente l'architecture de l'autoencodeur retenu. Le tableau présente le nombre de neurones et les fonctions d'activations utilisées pour chaque couche. Les couches d'entrée et de sorties utilisent des fonctions d'activation linéaires, car le modèle cherche à reconstruire le signal d'entrée. Pour les couches cachées, la fonction d'activation hyperbolique tangente est utilisée, car elle est compatible au format des données standardisée. Avec seulement 80 paramètres, il s'agit d'un très petit modèle de détection d'anomalies.

Tableau 4.5 - Résumé des paramètres de l'autoencodeur retenu

Couche	Nombre de neurones	Nombre de paramètres
Entrée (caractéristiques)	5	0
Encodeur (hyperbolique tangente)	4	24
Compression (hyperbolique tangente)	3	15
Décodeur (hyperbolique tangente)	4	16
Reconstruction (linéaire)	5	25
Nombre total de paramètres		80

La fonction d'optimisation utilisée est Adam, car elle fonctionne bien et n'a pas besoin d'être ajusté manuellement (Géron, 2023). La fonction de perte utilisée est la moyenne des écarts absolue (*MAE*). Durant l'entraînement, l'optimiseur cherche à minimiser la fonction de perte, donc la moyenne des écarts absolue entre la prédiction et la valeur réelle. Au moment de la validation, la même mesure de performances est utilisée pour suivre l'évolution de la reconstruction de l'autoencodeur. Une fois le modèle compilé, les paramètres d'entraînement utilisés sont *batch_size* de 8 et 200 *epochs*. Un *batch_size* de 8 est sélectionné, car le jeu de données est très petit et 200 *epochs* permet plus de cycles que nécessaire pour que le modèle apprenne des données, car le nombre de cycles optimal est déterminé par la fonction d'EarlyStopping qui est configurée pour restaurer les paramètres optimums après 10 cycles sans amélioration significative (paramètre patience = 10). Par la suite, le modèle est entraîné avec le jeu de données normales seulement.

Le calcul de l'erreur de reconstruction permet de déterminer dans quelle marge l'autoencodeur est en mesure de reconstruire une représentation des données correspondant à celle des données normales. Différents seuils de détection ont été testé entre le 80^e et le 95^e percentile. Le seuil de détection optimal est déterminé au 80^e percentile, car il permet une bonne détection des anomalies tout en minimisant le nombre de faux positifs. Le jeu de données de test comprend des données normales et des données représentant des anomalies. Ce jeu permet d'évaluer les performances du modèle et déterminer s'il est en mesure de différencier les données normales des anomalies en fonction des caractéristiques choisies pour les jeux de données. Si l'erreur

calculée est plus grande que le seuil, alors l'enregistrement est identifié comme une anomalie. Une fois le jeu de test traité, un rapport de classification est généré afin d'évaluer les performances globales du modèle.

Le Tableau 4.6 présente les résultats obtenus avec un seuil de détection au 80^e percentile pour les données test. Le modèle parvient à détecter, parmi les prédictions normales, seulement 64% (*precision*) des données effectivement normales, mais 90% (*recall*) des données normales sont classées correctement. La moyenne harmonique de ces deux indicateurs (*f1-score*) à 75% indique que le modèle manque de précision. Parmi les anomalies détectées, le modèle réussit la détection à 98% et il parvient à classer correctement 89% des anomalies. Donc globalement, la fiabilité du modèle est de 89%. Autrement dit, le modèle détecte bien les situations anormales, mais certaines anomalies sont classées comme étant normales. Quand le modèle prédit une anomalie, il est très robuste et la majorité des anomalies sont détectées. La moyenne macro ou non pondérée correspond à la performance du modèle pour détecter les anomalies parmi les données normales sans tenir compte d'un possible déséquilibre du nombre de données pour chacune des classes. Étant donné qu'il y a plus d'anomalies que de données normales dans le jeu de test utilisé, la moyenne pondérée permet d'en tenir compte dans le calcul de la performance du modèle. Le modèle obtenu n'est pas parfait, mais il permet tout de même de réaliser toutes les étapes du pipeline et d'obtenir une performance correcte, compte tenu du très petit jeu de données utilisé. Bien que très peu de données soient utilisées dans cette étude, dans un contexte réel il est nécessaire de collecter beaucoup plus de données. Comme les données sont sélectionnées en fonction des défaillances à détecter, quelques centaines à quelques milliers d'enregistrements seraient suffisants à améliorer la qualité du modèle et des analyses. Pour un fonctionnement à régime constant, le volume nécessaire de données peut être collecté en une à quatre semaines en fonction du temps de fonctionnement et de la fréquence de la collecte des données. Le modèle pourrait être amélioré en utilisant plus de données et en testant d'autres ajustements au modèle et aux hyperparamètres. Lors de la collecte des données normales, du bruit a volontairement été introduit afin de représenter des variations subtiles, comme on pourrait en retrouver dans un contexte industriel. Ces variations dans le jeu de données normales fait en sorte que le modèle classe des observations normales,

comme des anomalies. Pour améliorer cette situation, il est nécessaire de collecter plus de données normales afin que le modèle soit en mesure de mieux prendre en compte la variabilité dans les données. Aussi, comme les variations ou la dégradation évolue dans le temps, d'autres approches comme l'utilisation de fenêtres temporelles glissantes peuvent aider à mieux classifier les enregistrements. Dans ce cas au lieu de considérer les enregistrements comme des points individuels, ils sont analysés en tenant compte des séquences de données temporelles.

Tableau 4.6 - Rapport de classification de l'autoencodeur

	Précision	Rappel	Score f1	Support
Classe 0 (normal)	0.64	0.90	0.75	10
Classe 1 (anomalie)	0.98	0.89	0.93	44
Fiabilité du modèle			0.89	54
Moyenne macro	0.81	0.89	0.84	54
Moyenne pondérée	0.91	0.89	0.90	54

La courbe précision-rappel permet de visualiser la performance du modèle pour des classes déséquilibrés. La Figure 4.5 présente la courbe précision-rappel obtenue et la valeur son aire sous la courbe (AUC). L'aire sous la courbe peut varier de zéro à un et une valeur élevée indique que le modèle parvient à bien séparer les anomalies et les données normales. Le résultat obtenu est très élevé, ce qui est optimiste compte tenu que le jeu de données utilisé est très petit.

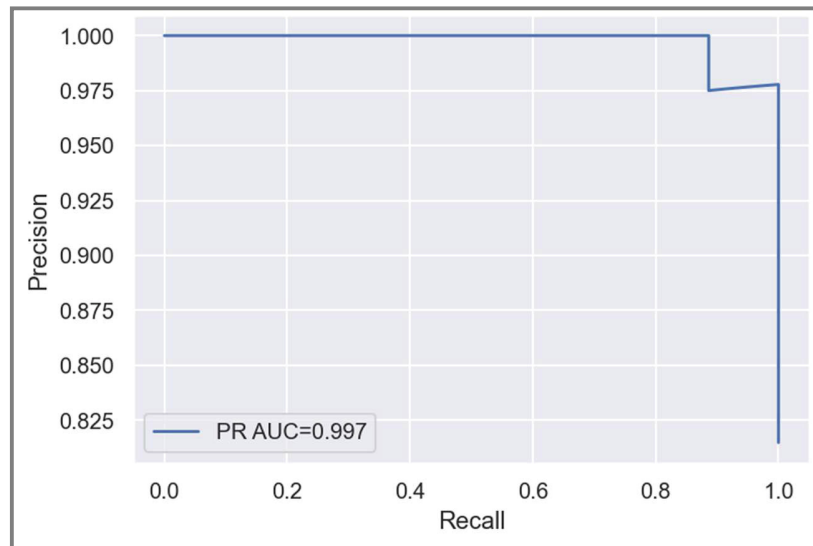


Figure 4.5 – Aire sous la courbe précision-rappel pour le jeu de données

Les courbes de pertes d'entraînement et de validation permettent de déterminer si le modèle apprend bien à reconstruire la représentation des données et si le modèle généralise bien avec de nouvelles données. La Figure 4.6 montre que les deux courbes décroissent régulièrement, puis elles atteignent un plateau. La courbe d'entraînement (*training*) est plus basse que la courbe de validation, ce qui indique que le modèle apprend bien et qu'il n'a pas appris par cœur les relations entre les données. Finalement, l'écart entre les courbes est modéré, mais acceptable.

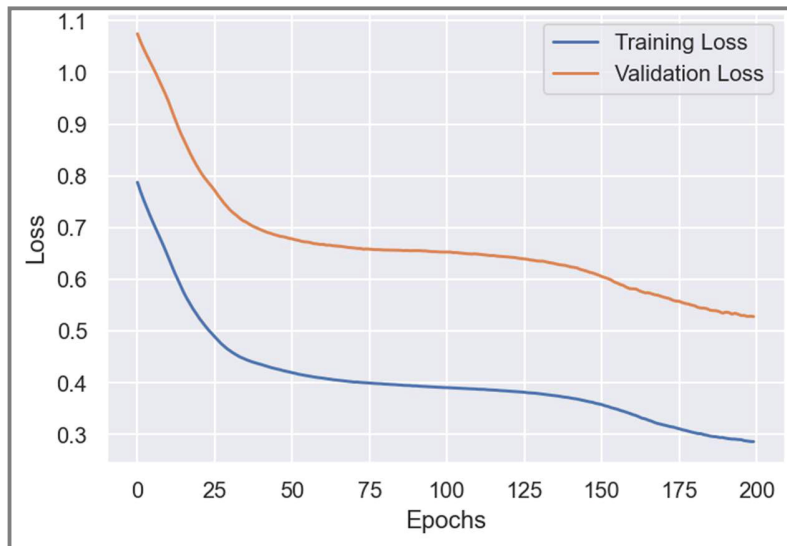


Figure 4.6 - Graphique de la perte d'entraînement et de validation

Un histogramme permet de visualiser la distribution des erreurs du jeu de test en fonction du seuil utilisé. La Figure 4.7 présente la distribution obtenue avec le jeu de données de test. Le seuil identifié par la ligne verticale sépare bien les données normales des anomalies. La séparation entre les données normales et les anomalies est assez claire. L'erreur est faible pour les données normales et elle est plus élevée pour les anomalies. Le modèle apprend bien la structure des données normales, mais les erreurs restent élevées.

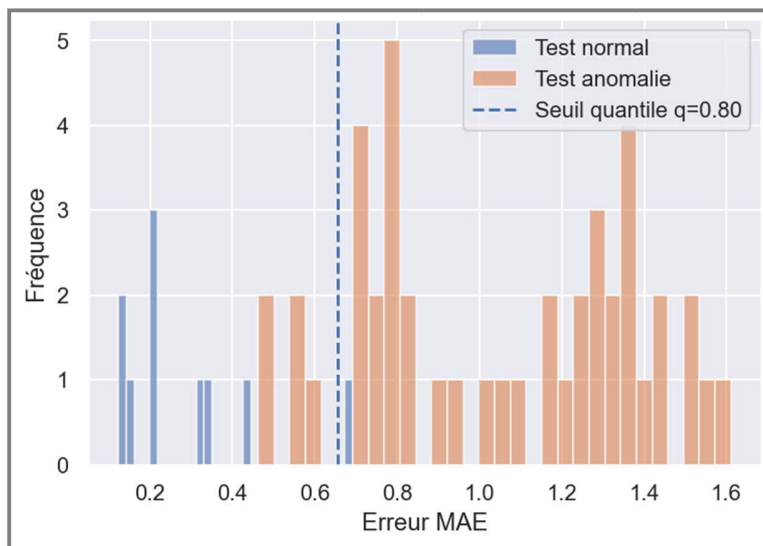


Figure 4.7 - Distribution des erreurs en fonction du seuil

Ensuite, une matrice de confusion permet de visualiser les résultats de la détection d'anomalies. Pour chaque échantillon testé, la matrice affiche la prédiction en fonction de l'étiquette réelle. La Figure 4.8 présente les mêmes données, cette fois sous le format d'une matrice de confusion. Elle indique que le modèle prédit 9 fois que les données sont normales alors que c'est effectivement le cas, 39 fois il prédit une anomalie alors que c'est effectivement le cas, 5 fois il prédit des données normales alors qu'il s'agit d'anomalies et il prédit 1 anomalie alors que la donnée est normale.

Réal	Normal	9	1
	Anomaly	5	39
		Normal	Anomaly
		Prédiction	

Figure 4.8 - Matrice de confusion
pour les données de test

Finalement, le modèle est testé à l'aide d'une fonction où en sélectionnant l'index d'un enregistrement et il est possible de vérifier le bon fonctionnement du modèle et connaître la fiabilité du résultat retenu. La fonction retourne les paramètres de l'enregistrement analysé, la prédiction et un pourcentage associé à la probabilité de la prédiction. La probabilité affichée, permet au technologue d'évaluer de façon plus intuitive, l'évaluation du modèle. Une fonction sigmoïde transforme l'erreur de reconstruction obtenue par le modèle. Plus l'erreur est grande, plus le pourcentage augmente pour l'anomalie et descend pour une donnée normale. Le Tableau 4.7 montre les résultats pour un échantillon choisi manuellement. La prédiction pour cet échantillon est normale avec une probabilité à 99%. L'utilisateur peut également consulter les caractéristiques de l'échantillon.

Tableau 4.7 - Résultats obtenus pour un échantillon

Ligne index	3
Valeurs :	
Courant (A)	8.548
x vitesse (mm/s)	5.97
y vitesse (mm/s)	2.98
z vitesse (mm/s)	2.95
débit (gpm)	15.95
Prédiction	normal
Probabilité anomalie	1.38 %
Probabilité normal	98.62 %

Cette fonction apporte donc une valeur pédagogique pour le technologue, car le technologue peut visualiser les données et la prédiction associée. Ainsi, il peut se familiariser avec les données et mieux comprendre lesquelles ont un plus grand impact sur les prédictions ou les performances du système. Pour cette étape, le technologue doit donc connaître les différents types de modèles d'apprentissage automatiques généralement utilisés en maintenance, leurs applications et les étapes de préparation des données nécessaires en fonction des analyses.

4.2.1.6 Inférences à partir du modèle

Une fois le modèle déployé, le technologue reçoit des informations indiquant les conditions de fonctionnement du système. Le technologue doit être en mesure de déterminer la qualité des renseignements obtenus et si les résultats sont susceptibles d'être faussés par des conditions changeantes ou si le modèle nécessite d'être entraîné avec de nouvelles données. À cette étape, le technologue est un maillon important pour valider la qualité des informations obtenues et il doit pouvoir réagir ou informer les experts des données s'il détecte de problèmes liés aux analyses. Bien qu'il doive pouvoir se fier aux résultats d'analyses pour tirer un bénéfice dans son travail, il doit pouvoir réagir s'il a un doute au sujet des résultats. Il doit également pouvoir suggérer des améliorations ou des changements au besoin. Pour cette étape, il a donc besoin d'une bonne connaissance du fonctionnement des différents modèles.

4.3 Résultats de la détection d'anomalies et discussion

Le protocole d'expérimentation présenté au Chapitre 4.2 montre que la majeure partie des étapes du *pipeline* concerne l'acquisition, la préparation, l'exploration et le prétraitement des données. En plus de nécessiter beaucoup plus de temps, comparativement au temps nécessaire pour les analyses, ces étapes sont d'une grande importance. En mettant à profit les connaissances du domaine du technologue, il est en mesure de cibler les données nécessaires pour suivre les conditions de fonctionnement de la pompe. L'utilisation d'une AMDEC lui permet de déterminer les modes de détections appropriés pour le suivi en fonction des défaillances les plus critiques. Suite à sa formation, le technologue aura les connaissances pour être en mesure d'appliquer de façon rigoureuse les étapes du pipeline d'apprentissage automatique. Ce pipeline permet donc au technologue de développer son expérience et développer la capacité à donner un sens aux données de fonctionnement et ainsi l'aider dans sa prise de décisions tandis qu'il gagne de l'expérience durant la progression de sa carrière. La connaissance du domaine combinée au traitement des données, à l'analyse de signal et à l'augmentation des données, permet de réaliser des analyses avec un jeu de données considérablement réduit comparativement à l'utilisation de données brutes. L'utilisation d'une AMDEC a permis de déterminer que les données nécessaires à la détection des défaillances dont la criticité est la plus élevée sont le courant, la vibration et le débit. Ces caractéristiques ont donc été sélectionnées pour l'entraînement de l'autoencodeur utilisé pour la détection d'anomalies. L'utilisation de la passerelle permet de simplifier les étapes importantes afin de rendre accessibles au technologue les techniques d'apprentissage automatique. En se familiarisant avec l'exploration des données du système, le technologue est en mesure de suivre l'évolution des paramètres de fonctionnement de la pompe et déterminer si une intervention est nécessaire. L'Internet des objets permet au technologue d'avoir accès aux données plus facilement, peu importe, où il se trouve et l'apprentissage automatique permet de traiter en continu, une quantité d'information plus grande que ce qu'il serait en mesure de faire sans ces outils. L'utilisation des données de fonctionnement de la pompe est une bonne approche pour débiter un projet d'intelligence artificielle en maintenance, car sa mise en place est simplifiée par l'utilisation de la passerelle et une fois les premières étapes réalisées, elle

peut évoluer vers d'autres analyses plus poussées. Le technologue en maintenance industrielle possède une expertise liée au fonctionnement et aux modes de défaillances des équipements qui lui permettent de simplifier les analyses en utilisant les données les plus pertinentes, au lieu d'utiliser seulement des données brutes.

L'utilisation d'un autoencodeur permet de déployer rapidement un premier projet d'apprentissage automatique en maintenance industrielle pour détecter les défaillances d'un système. La démarche utilisée permet d'introduire les concepts importants pour qu'un technologue n'ayant aucune expérience en apprentissage automatique puisse participer au développement d'un premier projet. Le technologue doit pouvoir collaborer au développement et à l'implantation de projets d'analyse des données et d'apprentissage automatique. L'implication du technologue doit se faire au niveau de la définition du problème et des requis, la participation au développement des modèles et des tests et donner leurs avis au niveau de l'interprétation des résultats, car c'est lui qui recevra les alertes en cas de défaillances. Les compétences nécessaires ont été mises en lumière à l'aide de la démarche utilisée. Ces compétences sont la compréhension, la manipulation et la présentation des données, l'utilisation des mathématiques et des statistiques utilisées pour la manipulation et le traitement des données, l'utilisation d'un langage de programmation, connaître les principales bibliothèques utilisées en traitement et analyse des données, l'analyse de signal, les infrastructures, techniques de manipulation et de préparation des données, l'utilisation des outils de visualisation, les techniques d'analyse de défaillances comme l'AMDEC, la compréhension du fonctionnement des algorithmes d'apprentissage automatique et la préparation des jeux de données, l'utilisation des algorithmes d'apprentissage automatique en maintenance, la connaissance des termes utilisés et le fonctionnement général des différents modèles pour collaborer avec les experts en science des données. Le technologue en maintenance industrielle utilise déjà différentes techniques de surveillance pour déterminer l'état de fonctionnement des équipements dont il est responsable. En collectant les données de fonctionnement pour alimenter des modèles de détection d'anomalies, la solution proposée lui permet de combiner et traiter une très grande quantité d'information, dans le but de déterminer quelles sont les conditions de fonctionnement et s'il est nécessaire d'intervenir sur un équipement. Différentes

techniques d'analyse vibratoires sont utilisées depuis longtemps pour le suivi et l'analyse des machines. Ces techniques permettent de convertir un signal temporel en différents descripteurs afin de fusionner les données de vibration avec les autres données de maintenance et les données de procédé. Pour la détection d'anomalies, elle est utilisée pour augmenter le nombre de caractéristiques utilisées pour la détection de défauts. Dans cette étude, c'est le mode de détection qui permet d'identifier le plus de modes de défaillances parmi celles ayant la plus grande criticité. Dans des travaux futurs, l'utilisation de l'analyse vibratoire est donc un outil essentiel pour faire de la catégorisation de défauts et ainsi identifier le type de défaillance et son emplacement dans le système analysé. L'utilisation de plusieurs caractéristiques pour l'analyse des enregistrements permet d'obtenir des résultats rapidement malgré une quantité de données relativement faible. Une fois que le technologue maîtrise les différentes bases du *pipeline* d'analyse des données, il pourra également utiliser plus efficacement différentes solutions commerciales comme Edge Impulse ou Nano Edge AI Studio. Ces solutions, combinées à l'utilisation de la passerelle proposée, permettent de simplifier davantage les étapes du pipeline grâce à l'utilisation d'une interface utilisateur plus intuitive. L'utilisation d'une AMDEC permet de déterminer quelles données sont nécessaires en fonction du type de défaillances à détecter et il permet d'orienter rapidement le choix des analyses et le type de défaillances que le modèle sera en mesure de détecter.

CHAPITRE 5

COMPÉTENCES NÉCESSAIRES

5.1 Introduction

Le technologue en maintenance industrielle possède les compétences techniques pour assurer le bon fonctionnement des équipements sous sa responsabilité. La nature de son travail fait en sorte qu'il est très près des équipements et il est en mesure de prendre les bonnes décisions lorsqu'un problème de fonctionnement survient. Les fiabilistes, quant à eux, améliorent la fiabilité et la performance des équipements industriels en surveillant les actifs et en analysant les défaillances. Ils utilisent des outils de suivi des indicateurs clés de performance. Les résultats obtenus permettent ensuite de développer des stratégies de maintenance dans le but de trouver des solutions durables aux problèmes de fonctionnement, réduire les pannes et les temps d'arrêt. En maintenance industrielle, la science des données peut être abordée de différentes façons en fonction des entreprises et de leurs contextes. Les compétences nécessaires pour mettre sur pied ce type de solutions peuvent varier énormément en fonction de l'envergure du projet d'analyse. Un technologue peut facilement débiter à l'aide d'outils comme Excel et un peu de programmation en Python. Ensuite, des connaissances en analyse des données permettent de structurer et visualiser les données. Pour les projets de plus grande envergure, une équipe spécialisée est nécessaire pour déployer des infrastructures afin de stocker, accéder et partager les données, développer des interfaces de programmation d'application ou API (Application Programming Interface) afin de croiser les sources de données et les résultats d'analyses, et pour maintenir les solutions mises en place.

Blok et al. proposent un cadre de compétences visant à développer la création de parcours d'apprentissage en IA. Ce travail est le fruit d'une collaboration entre le collègue Dawson et l'université Concordia dans le but de créer des parcours d'apprentissage pour les collèges et universités. L'outil proposé regroupe les compétences en modules adaptés à différents besoins d'apprentissage. Le cadre de compétence identifie les compétences fondamentales au niveau des connaissances, des compétences et des habiletés nécessaires en fonction des domaines

techniques, d'affaires et de l'humain (Blok, Trudeau et Cassidy, 2021). Au niveau du domaine technique, les sujets liés aux différents modules des compétences fondamentales nécessaires pour utiliser l'IA sont :

- Les données
- Les mathématiques et les statistiques
- La programmation
- L'apprentissage automatique
- L'apprentissage profond
- Les infrastructures de données
- Les bibliothèques et environnements de développement

Pour les besoins de cette étude, le cadre de compétence a été traduit et la numérotation a été adaptée dans le but de suivre la répartition des compétences à travers les étapes du pipeline proposé. Pour chacun des sujets identifiés, l'auteur présente les thèmes à aborder et les compétences qui y sont rattachées. Le tableau de l'ANNEXE I présente les compétences fondamentales pour les domaines techniques tels que décrits par Blok et al. Le tableau présente les compétences nécessaires pour utiliser l'IA en fonction des sujets et des thèmes importants. La dernière colonne identifie comment les compétences fondamentales ont été utilisées dans le pipeline proposé et s'il est nécessaire d'ajouter la compétence à la formation du technologue en maintenance industrielle. La légende suivante est utilisée :

- A : La compétence est utilisée dans la démarche proposée et devrait être incluse dans la formation;
- B : La compétence n'est pas utilisée dans la démarche proposée, mais devrait être incluse dans la formation;
- C : Cette compétence ne fait pas partie du rôle attendu du technologue, elle appartient à d'autres expertises;
- D : Cette compétence fait déjà partie de la formation du technologue.

Le tableau de l'ANNEXE I montre qu'au niveau des données, les compétences à intégrer dans la formation touchent les thèmes liés à la compréhension des données, les implications légales et éthiques, la manipulation et le traitement des données ainsi que la représentation des données. L'apport du technologue est surtout au niveau de la représentation, la préparation et la manipulation des données. Il devra également analyser des visualisations et des représentations tabulaires des données. Les thèmes associés aux mathématiques et aux statistiques font déjà partie de la formation des technologues. Ces thèmes peuvent donc être adaptés en fonction des compétences fondamentales utilisées pour les projets d'apprentissage automatique. Pour ce qui est de la programmation, les thèmes à aborder sont les compétences de bases en programmation, le développement logiciel et les notions de bases de données non relationnelles. Les principales tâches pour le technologue, à ce niveau, consistent à interpréter et écrire du code pour traiter et analyser des données. Au besoin, il pourra ajouter les bibliothèques nécessaires pour les tâches à réaliser. En ce qui a trait à la construction de bases de données relationnelles, le technologue n'est généralement pas impliqué au niveau de la création des bases de données, mais il peut être amené à manipuler des données issues de bases de données. Cette compétence n'est donc pas ajoutée au programme de TMI. Au niveau de l'apprentissage automatique, le technologue doit considérer les aspects liés au prétraitement des données et à l'apprentissage supervisé ou non supervisé, bien que son implication réside principalement dans la préparation des caractéristiques pour les tâches d'apprentissage. Au sujet de l'apprentissage profond, ce sont les écosystèmes de l'IA qui sont importants pour le technologue, pour examiner les perspectives liées à l'utilisation de l'IA et participer à l'intégration des systèmes dans un contexte réel. Pour ce qui est des infrastructures des données, le technologue doit être en mesure de manipuler des données de façon sécuritaire, selon le type de stockage utilisé, que ce soit dans des fichiers, des bases de données ou des services infonuagiques. L'utilisation des réseaux de neurones et les notions de calculs parallèle ne sont pas intégrées à la formation, car l'objectif pour le technicien est de collaborer à alimenter les modèles, donc d'avoir une compréhension de base du fonctionnement, sans avoir à utiliser directement les modèles et les processeurs pour l'entraînement. Finalement, le technologue doit pouvoir utiliser les outils associés à un environnement de développement et

les bibliothèques généralement utilisés en analytique et en apprentissage automatique. Encore une fois, dans le cas des outils à utiliser pour développer des modèles, le technologue doit pouvoir utiliser les fonctionnalités de base des outils de développement, sans qu'il ne soit nécessaire d'utiliser les fonctions plus avancées pour personnaliser, améliorer ou modifier les outils.

Blok et al. ont détaillés les compétences fondamentales, présentées plus haut en sous-compétences pour les domaines techniques. En fonction des compétences fondamentales à intégrer dans la formation des technologues, les sous-compétences pertinentes ont été identifiées. Le tableau de l'ANNEXE II reprend les étapes de la méthodologie de cette étude, pour y associer les sous-compétences nécessaires à chacune des étapes du pipeline proposé. Ces sous-compétences répondent directement aux besoins pour faire de la détection d'anomalies en maintenance industrielle et elles permettent au technologue de collaborer avec les experts en analyses des données tout au long du pipeline présenté. Comme c'est le cas pour de nombreux programmes collégiaux, le nombre d'heures disponible implique de faire des choix sur la matière qu'il est possible d'enseigner. Les compétences qui n'ont pas à être intégrées dans le programme concernent la construction de bases de données relationnelles, l'utilisation des outils spécifiques au deep learning, l'utilisation des infrastructures de calculs parallèle et les fonctions plus avancées des outils de développement.

5.2 Démarche proposée pour intégrer les compétences

À partir du croisement entre les étapes du pipeline et les compétences fondamentales présentées dans le tableau de l'ANNEXE II, un lien a été établi avec les cours de l'actuel programme de TMI au Cégep du Vieux Montréal. Ces liens sont présentés à l'ANNEXE III et proposent une façon d'introduire les compétences fondamentales en IA dans le parcours des étudiants. Le sigle du cours identifié permet de le situer dans le programme de TMI au cégep du Vieux Montréal. Par exemple, dans les sigles 241-M11-VM et 241-D65-VM, 241 identifie au collégial, un programme d'études techniques liées au domaine de la mécanique, D fait références aux cours spécifiques du programme de TMI alors que M réfère à un cours du tronc

commun avec le programme de technique de génie mécanique (TGM) le chiffre suivant indique la session où se donne le cours suivi d'un chiffre correspondant à la chronologie du cours dans la session. Finalement VM indique que c'est un cours élaboré par le cégep du Vieux Montréal. Donc 241-D65-VM représente un cours de TMI donné en sixième session au CVM alors que 241-M11-VM indique un cours du tronc commun entre TGM et TMI donné en première session au CVM. L'ANNEXE III présente le croisement entre le pipeline et les compétences, il ressort que quatre cours du programme actuels pourraient potentiellement comprendre des sous-compétences en lien avec l'IA et il serait nécessaire d'ajouter deux cours, identifiés 241-XXX-VM, pour être en mesure de couvrir l'ensemble des compétences nécessaires à réaliser la démarche de cette étude. Ces cours sont présentés au tableau de l'ANNEXE III. Dans le cours de Métrologie et capteurs, en première session, il serait possible d'établir un pipeline de données en établissant une connexion entre les capteurs et les systèmes informatiques pour la collecte de données. Dans le cours de Mathématique II (code 201) donné en deuxième session, les données sont analysées à l'aide de tables et de graphiques. Un cours de programmation et science des données est ajouté en quatrième session, car plusieurs compétences doivent être ajoutées et elles sont loin de ce qui se donne actuellement dans le programme de TMI. Dans le cours Gestion de maintenance I, toujours en quatrième session, l'étudiant pourra construire et manipuler des bases de données. Dans le cours Projet en fiabilité, à la sixième session, il est possible d'analyser les cas où il est possible d'appliquer l'apprentissage automatique en maintenance. Ces différents modules intégrés au parcours de l'étudiant permettront ensuite d'ajouter un dernier cours à la sixième session, où l'étudiant sera amené à réaliser un projet d'application de l'apprentissage automatique en maintenance industrielle. L'étudiant participe donc à toutes les étapes du projet, cependant, pour la portion d'entraînement, de validation et d'ajustement du modèle, le technologue assiste à une démonstration de ces étapes sans les réaliser, car ces tâches relèvent d'autres expertises que celles du technologue. À ce stade, les solutions d'AutoML peuvent s'avérer intéressantes pour réaliser ces étapes sans en maîtriser le fonctionnement. Les modifications proposées à la grille de cours auront un impact sur le contenu actuel il faudra s'assurer de faire de la place pour ajouter les nouvelles compétences sans modifier de façon significative le nombre d'heures total de la formation.

La Figure 5.1 présente la chronologie des cours à adapter ou à ajouter, ainsi que le contenu associé aux principales étapes du pipeline. À la première session, les étudiants établissent un pipeline des données pour enregistrer des données à partir de capteurs. À la deuxième session, les étudiants utilisent différents outils pour analyser et visualiser les données. À la quatrième session, un langage de programmation est utilisé pour manipuler les données, puis les données sont utilisées pour alimenter les systèmes de gestion de la maintenance. À la sixième session, un projet permet de sélectionner les outils d'apprentissage automatique et d'appliquer la solution pour répondre à un besoin de maintenance.

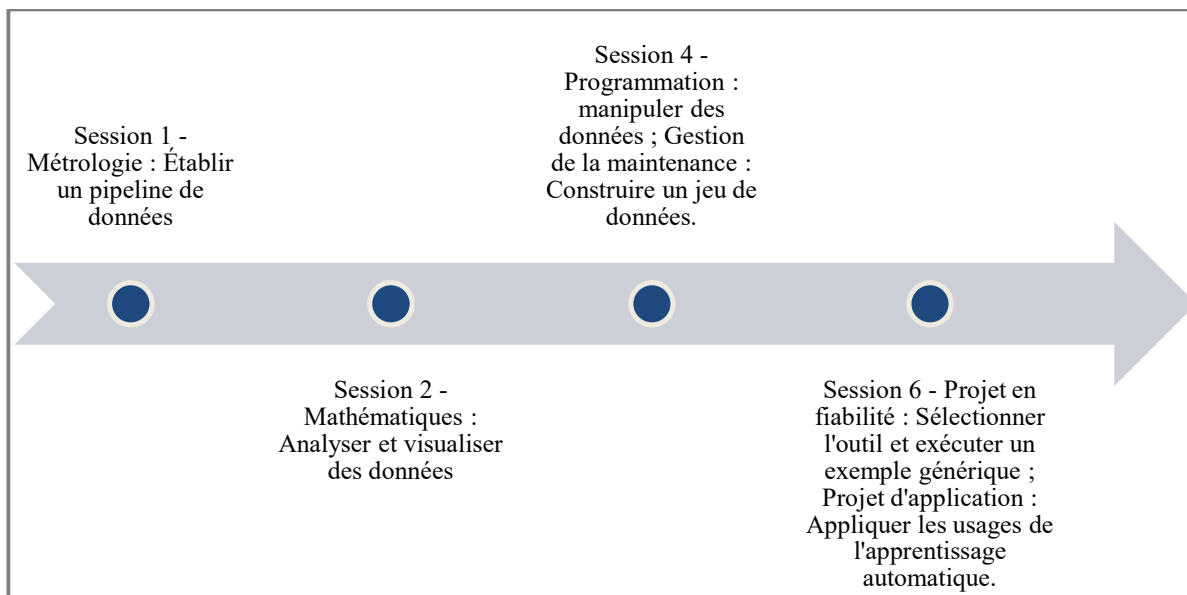


Figure 5.1 - Chronologie des cours et des principales étapes

5.3 Grille diagnostique des compétences nécessaires

Le tableau de l'ANNEXE IV présente les sous-compétences associées au pipeline et les activités d'apprentissage associées aux compétences nécessaires. Ces activités d'apprentissage sont donc, les étapes importantes à réaliser pour être en mesure de mettre en œuvre les étapes du pipeline. Cette démarche est propre aux besoins du technologue, car elle combine les étapes de résolution de problèmes utilisés en maintenance, la collecte, le chargement, le traitement et

l'exploration des données, l'analyse de signal et le calcul des caractéristiques, la programmation couramment utilisée en science des données, le pipeline de données pour alimenter les modèles et les analyses, en plus d'inclure le technologue dans la prise de décision, l'amélioration de la solution et de retirer un bénéfice de l'utilisation de la solution déployée. La colonne tâche, permet de déterminer les compétences et les activités d'apprentissage nécessaire pour être en mesure de réaliser l'étape du pipeline associée. Ce tableau permet, donc, de cibler les compétences à développer en fonction des besoins spécifiques des technologues en fonction de l'étape du pipeline à maîtriser. À partir de cette grille, il est ensuite possible de développer des plans de formations personnalisés.

Ces nouvelles compétences permettront aux technologues d'utiliser les données de fonctionnement afin d'anticiper les pannes, prévenir les défaillances et améliorer les stratégies de maintenance. Ces compétences lui permettront une maîtrise de base des outils utilisés pour faire de l'apprentissage automatique, de communiquer avec les experts de la science des données et de comprendre le résultat des analyses afin de pouvoir tirer des conclusions à partir des données. Dans ce contexte, le technologue occupe une place importante dans la mise en place des projets d'analyse des données industrielles. Non seulement son travail sera directement impacté par les nouvelles solutions, mais il est aussi en mesure de collaborer avec les fiabilistes et scientifiques des données pour identifier les données à utiliser, collecter et structurer les données, établir un lien entre les données et les conditions de fonctionnement, identifier les modes de défaillance et les méthodes de détections appropriées et donner du sens aux résultats obtenus.

CONCLUSION

Le développement de compétences pour être en mesure d'utiliser l'IA dans les entreprises est maintenant incontournable. En intégrant le technologue en maintenance industrielle dans les processus d'analyse des données, il a la capacité d'établir les liens entre les machines industrielles et les outils d'analyse et de traitement des données. En collaborant avec les experts des données, il est en mesure d'apporter son expertise au niveau de la maintenance industrielle pour identifier efficacement les bonnes données à utiliser, participer à la collecte et effectuer le prétraitement des données afin de faciliter le travail des scientifiques des données. En retour, les équipes de maintenance bénéficieront d'outils adaptés à leurs besoins. En participant au développement de ces outils, ils seront assurés d'avoir accès à des outils pertinents répondant à un réel besoin. Cette étude a permis de proposer une méthode pour intégrer l'utilisation de l'apprentissage automatique dans la formation des technologues en maintenance industrielle. La méthode proposée utilise un cas concret pour tester toutes les étapes du déploiement d'un projet d'apprentissage automatique, dans le but d'assurer l'opérationnalisation du processus. Le technologue peut donc faire le lien entre les données des machines, les conditions de fonctionnement et les bonnes pratiques de maintenance afin d'accompagner les ingénieurs dans l'analyse des données et la mise en place de solutions utilisant l'apprentissage automatique afin d'améliorer les processus de production. Le travail réalisé a permis de déterminer les outils nécessaires pour collecter et analyser des données industrielles et détecter des problèmes de fonctionnement en utilisant l'apprentissage automatique. Ce travail a permis d'identifier les compétences nécessaires à chaque étape du pipeline et ainsi proposer une démarche pour développer les compétences du technologue. À partir de ces constats, une grille diagnostique permet d'identifier les activités d'apprentissage à réaliser afin de mettre en œuvre chacune des étapes du pipeline.

L'approche proposée consiste à :

- Impliquer le technologue dès le départ pour identifier la problématique et les cas où l'apprentissage automatique peut permettre de répondre au besoin.

- Ajouter des compétences à sa formation pour qu'il soit en mesure d'utiliser les outils d'apprentissage automatique pour l'acquisition, le traitement, l'exploration et la préparation des données. Ainsi, il sera en mesure de comprendre les étapes et les objectifs de l'utilisation de l'apprentissage automatique en maintenance industrielle.
- Séparer le processus en étapes pour intégrer les compétences dans la grille de cours actuelle en favorisant les synergies avec le programme de TMI au CVM et minimiser les impacts sur la grille de cours.
- Proposer une grille diagnostique pour développer des activités d'apprentissages en fonction des compétences et des étapes du pipeline.

Contributions

Cette étude propose une méthode s'appuyant sur des solutions à code ouvert pour intégrer l'apprentissage automatique dans la formation des technologues en maintenance industrielle. Les technologues bénéficiant de ces nouvelles compétences seront les vecteurs pour faciliter l'intégration de l'apprentissage automatique dans les industries où ils évolueront après leurs études collégiales. La grille diagnostique propose des activités d'apprentissages facilement transférables à d'autres programmes de formation désirant intégrer l'apprentissage automatique dans leur cursus. Cette étude présente un pipeline complet pour comprendre toutes les étapes nécessaires pour mettre en œuvre un premier projet d'apprentissage automatique. Ce processus permet de bien définir les solutions à envisager, les limites, les opportunités et les étapes subséquentes pour faire évoluer l'utilisation de l'apprentissage automatique dans un contexte de maintenance industrielle.

Coûts, risques et enjeux

Les solutions d'internet industriels des objets (IIo) pour collecter et analyser les données d'équipements industriels présentent plusieurs barrières à l'entrée, dont un coût élevé pour faire l'acquisition des équipements et des frais mensuels importants. La solution proposée comporte l'avantage de ne pas être sujette à un service externe dont l'utilisateur devient dépendant. Le technologue et les équipes de maintenance auront dorénavant accès à de nouveaux indicateurs, indiquant potentiellement plus de situations anormales. Il sera donc

important d'évaluer comment cette surcharge potentielle ou ces indicateurs négatifs pourront être gérés afin de ne pas produire une situation donnant l'impression d'une moins bonne performance qu'avant le déploiement de la solution.

Un aspect important à prendre en compte est l'évolution rapide de l'IA générative. Dans le contexte de ce projet, cette technologie permet de faire l'analyse des données, de générer facilement le code pour créer, améliorer les algorithmes et analyser les résultats obtenus. Elle peut donc s'insérer à plusieurs endroits dans le processus. Bien que l'IA générative permette de faciliter l'accès et l'utilisation du pipeline proposé, comme le technologue évoluera dans un contexte industriel où l'explicabilité et la traçabilité est importante, les compétences fondamentales présentées restent pertinentes pour comprendre le code et le fonctionnement de la solution. Le technologue doit donc être responsable de la solution mise en place et comprendre comment l'utiliser adéquatement. Les compétences fondamentales en apprentissage automatique et l'utilisation de l'IA générative peuvent donc être utilisées comme des outils pour augmenter la performance du technologue.

Travaux futurs

Afin d'aller plus loin avec les données collectées, il est possible de simuler plus d'anomalies et ajouter plus de conditions de fonctionnement. Le système gagnerait à être configuré pour faciliter l'accès aux composantes internes, au montage et au démontage, afin de générer plus d'anomalies et faciliter les inspections pour le suivi des conditions. Dans cette étude seulement la détection d'anomalie a été testée, mais il est possible de faire de la classification de défauts avec les mêmes données. L'analyse des séquences temporelles pourraient permettre de suivre l'évolution des défauts et analyser les dommages cumulatifs sur l'équipement. Une solution industrielle plus robuste peut être développée pour tester la solution dans un environnement industriel. Dans cette étude, une pompe est utilisée, mais de nombreux autres équipements peuvent faire l'objet de suivi des conditions de fonctionnement ou de la gestion de santé des actifs. On pense entre autres au suivi de moteurs électriques, systèmes de ventilation, compresseurs d'air, réducteurs de vitesse, convoyeurs, pont roulants, systèmes hydrauliques,

etc. Ces systèmes sont utilisés dans de nombreuses industries au Québec et sont généralement critiques au bon fonctionnement des processus de production. Il est également possible :

- D'intégrer une configuration automatique du système pour qu'elle s'adapte à la dynamique de différents équipements;
- Ajouter des données provenant d'image ou de vidéos pour bonifier les jeux de données;
- Analyser les impacts du déploiement d'un premier projet d'apprentissage automatique sur les tâches de l'équipe de maintenance;
- Analyser le rendement de la pompe à l'aide de l'apprentissage automatique;
- Explorer les applications basées sur le TinyML et AutoML pour des applications d'apprentissage automatique en maintenance industrielle.

En appliquant la méthodologie proposée dans cette étude, le technologue pourra développer les compétences nécessaires pour utiliser l'apprentissage automatique dans un contexte de maintenance industrielle, tout en collaborant avec les experts en analyse des données.

ANNEXE I

TABLEAU DES COMPÉTENCES EN IA POUR LES DOMAINES TECHNIQUES

Tableau-A I-1 - Compétences fondamentales en IA – domaine technique (Traduit et adapté de (Blok, Trudeau et Cassidy, 2021))

Sujets	Thèmes	Compétences	Intégration de la compétence
1. Données	1.1. Compréhension des données	1.1.1. Utiliser différents types de données et leurs représentations 1.1.2. Analyser les usages typiques des données en apprentissage automatique et en intelligence artificielle	A B
	1.2. Implications légales et éthiques	1.2.1. Évaluer les implications légales de la collecte et de l'utilisation des données 1.2.2. Évaluer les implications éthiques de la collecte et de l'utilisation des données	B B
	1.3. Manipulation et traitement des données	1.3.1. Préparer les données pour un projet en apprentissage automatique ou en intelligence artificielle 1.3.2. Manipuler les données 1.3.3. Mettre en place des mesures de sécurité des données	A A B
	1.4. Représentation des données	1.4.1. Analyser les visualisations graphiques des données 1.4.2. Analyser les représentations tabulaires des données	A A
2. Mathématiques statistiques	2.1. Probabilités et statistiques	2.1.1. Résoudre des problèmes en utilisant des méthodes statistiques et des concepts de probabilité	D

Sujets	Thèmes	Compétences	Intégration de la compétence
	2.2. Vecteurs	2.2.1. Analyser des problèmes impliquant des vecteurs	D
	2.3. Calcul différentiel	2.3.1. Analyser des problèmes en utilisant le calcul différentiel	D
	2.4. Calcul intégral	2.4.1. Analyser des problèmes en utilisant le calcul intégral	D
	2.5. Méthodes numériques	2.5.1. Interpoler et ajuster des courbes 2.5.2. Résoudre numériquement des équations différentielles ordinaires 2.5.3. Résoudre numériquement des problèmes d'optimisation	D D D
3. Programmation	3.1. Compétences de base en programmation	3.1.1. Écrire du code en respectant la syntaxe et la structure 3.1.2. Intégrer des bibliothèques 3.1.3. Améliorer la performance du code	A A A
	3.2. Développement logiciel	3.2.1. Utiliser le contrôle de version 3.2.2. Appliquer un processus de développement logiciel	B B
	3.3. Bases de données	3.3.1. Construire une base de données relationnelle 3.3.2. Construire une base de données non relationnelle	C B
4. Apprentissage automatique	4.1. Prétraitement des données	4.1.1. Préparer des caractéristiques pour des tâches d'apprentissage supervisé ou non supervisé 4.1.2. Établir des pipelines de données	A A

Sujets	Thèmes	Compétences	Intégration de la compétence
	4.2. Apprentissage supervisé	4.2.1. Gérer un cadre d'apprentissage supervisé 4.2.2. Appliquer l'apprentissage supervisé à des tâches spécifiques	A A
	4.3. Apprentissage non supervisé	4.3.1. Gérer un cadre d'apprentissage non supervisé 4.3.2. Appliquer l'apprentissage non supervisé à des tâches spécifiques	B B
5. Apprentissage profond	5.1. Écosystèmes de l'IA	5.1.1. Examiner les perspectives techniques sur les implications sociales de l'IA 5.1.2. Identifier la relation entre l'IA et les systèmes informatiques 5.1.3. Discuter de l'interaction entre intelligence artificielle et intelligence humaine 5.1.4. Intégrer des systèmes d'IA dans des contextes réels	B B B A
	5.2. Réseaux de neurones artificiels	5.2.1. Utiliser des réseaux de neurones multicouches généraux 5.2.2. Utiliser des modèles spécifiques de deep learning	C C
6. Infrastructures de données	6.1. Stockage de données	6.1.1. Manipuler des données stockées dans des fichiers 6.1.2. Manipuler des données stockées dans des bases de données	A B
	6.2. Calcul parallèle	6.2.1. Utiliser des processeurs parallèles pour entraîner des modèles de données 6.2.2. Utiliser des clusters distribués pour entraîner des modèles de données	C C

Sujets	Thèmes	Compétences	Intégration de la compétence
	6.3. Infonuagique	6.3.1. Utiliser différents types d'architectures infonuagiques 6.3.2. Déployer des modèles à grande échelle efficacement 6.3.3. Assurer la sécurité des données et infrastructures	B C B
7. Bibliothèques et environnements de développement	7.1. Outils	7.1.1. Sélectionner l'outil approprié 7.1.2. Installer l'outil 7.1.3. Configurer l'outil 7.1.4. Trouver la documentation de l'outil 7.1.5. Exécuter des exemples génériques avec l'outil 7.1.6. Personnaliser l'outil pour un objectif spécifique 7.1.7. Proposer des améliorations pour l'outil 7.1.8. Modifier l'outil via des processus open source	A A A A A C C C

ANNEXE II

TABLEAU DES SOUS-COMPÉTENCES EN FONCTION DU PIPELINE

Tableau-A II-1 – Sous-compétences en fonction du pipeline

Étape du pipeline	Tâche	Sujets, thèmes et compétences	Sous-compétences
Définition du besoin	Identifier clairement le problème à régler	<p>1. Données 1.1 Compréhension des données</p> <p>1.1.1 Utiliser différents types de données et leurs représentations</p> <p>1.1.2 Analyser les usages typiques des données en apprentissage automatique et en intelligence artificielle</p>	<p>1.1.1.10 Identifier les effets de la manipulation des données sur leur représentation (par ex. suppression de valeurs aberrantes)</p> <p>1.1.1.11 Sélectionner des types et structures de données appropriés selon le contexte</p> <p>1.1.1.13 Adapter le schéma de données aux exigences du projet et au cas d'utilisation visé</p> <p>1.1.2.4 Évaluer si les données respectent les exigences éthiques (avec l'aide d'experts de domaine et de sciences sociales)</p> <p>1.1.2.8 Lister les applications du ML et de l'IA dans différents contextes</p> <p>1.1.2.10 Évaluer l'utilité d'un jeu de données pour le développement d'un modèle ML ou IA spécifique</p> <p>1.1.2.13 Évaluer les hypothèses des jeux de données acquis (seconde main, fournisseurs) quant à leur collecte, traitement et distribution</p> <p>1.1.2.14 Déterminer la lignée (provenance) des données qui ne sont pas de première main</p>
Identifier les technologies applicables	<p>5. Apprentissage profond 5.1 Ecosystèmes de l'IA</p> <p>5.1.2 Identifier la relation entre l'IA et les systèmes informatiques</p>		<p>5.1.2.1 Distinguer les artefacts technologiques qui utilisent ou non l'IA</p> <p>5.1.2.2 Caractériser les ingrédients technologiques clés d'un projet d'IA réussi</p> <p>5.1.2.3 Comparer l'IA et les approches TI traditionnelles à divers problèmes</p> <p>5.1.3.1 Analyser les caractéristiques qui rendent une entité intelligente</p> <p>5.1.3.2 Comparer l'intelligence artificielle et l'intelligence humaine sur divers problèmes</p> <p>5.1.3.3 Positionner la nécessité des compétences humaines dans un écosystème IA</p> <p>5.1.3.4 Discuter de la relation entre modèles d'IA et biologie</p>

Étape du pipeline	Tâche	Sujets, thèmes et compétences	Sous-compétences
	Sélectionner les outils	<p>7. Bibliothèques et environnements de développement</p> <p>7.1 Outils</p> <p>7.1.1 Sélectionner l'outil approprié</p> <p>7.1.4 Trouver la documentation de l'outil</p> <p>7.1.5 Exécuter des exemples génériques avec l'outil</p>	<p>7.1.1.1 Sélectionner l'outil approprié</p> <p>7.1.4.1 Trouver la documentation de l'outil</p> <p>7.1.5.1 Exécuter des exemples génériques avec l'outil</p>
Acquisition des données	Attribuer des étiquettes pour la classification des données	<p>1. Données</p> <p>1.1 Compréhension des données</p> <p>1.1.2 Analyser les usages typiques des données en apprentissage automatique et en intelligence artificielle</p>	<p>1.1.2.11 Constituer un jeu de données original pour l'entraînement et les tests d'un modèle ML ou IA</p>
	<p>Enregistrer les données dans des fichiers csv</p> <p>Manipuler les données par programmation</p>	<p>3. Programmation</p> <p>3.1 Compétences de base en programmation</p> <p>3.1.1 Écrire du code en respectant la syntaxe et la structure</p>	<p>3.1.1.1 Utiliser divers types et structures de données</p> <p>3.1.1.2 Réaliser des opérations avec des types numériques et des chaînes de caractères</p> <p>3.1.1.3 Réaliser des opérations avec des types mixtes impliquant des conversions</p> <p>3.1.1.4 Manipuler des structures de type tableau (instancier, itérer, mettre à jour, filtrer, mapper, réduire, découper, copier)</p> <p>3.1.1.5 Manipuler des structures de type objet (instancier, itérer, mettre à jour, filtrer, mapper, réduire, copier)</p> <p>3.1.1.6 Différencier les types mutables et immuables</p> <p>3.1.1.8 Travailler avec des fichiers</p> <p>3.1.1.9 Lire des fichiers depuis des sources locales ou distantes</p> <p>3.1.1.10 Écrire des fichiers dans un stockage local</p> <p>3.1.1.11 Appliquer la logique de contrôle d'exécution</p> <p>3.1.1.12 Itérer sur des structures de données</p> <p>3.1.1.13 Utiliser des boucles et conditions pour contrôler l'exécution</p> <p>3.1.1.14 Utiliser des fonctions et méthodes</p> <p>3.1.1.15 Intégrer de la logique dans des fonctions</p>

Étape du pipeline	Tâche	Sujets, thèmes et compétences	Sous-compétences
			3.1.1.17 Appliquer la portée (scope) dans la construction de fermetures 3.1.1.18 Composer des opérations par chaînage 3.1.1.19 Écrire des fonctions pour l'exécution asynchrone 3.1.1.20 Utiliser des modèles de programmation courants
	Utiliser les bibliothèques disponibles	3.1.2 Intégrer des bibliothèques	3.1.2.1 Gérer les dépendances d'un projet 3.1.2.2 Installer des dépendances (pip, apt, etc.) 3.1.2.3 Documenter les dépendances 3.1.2.4 Résoudre les conflits de dépendances 3.1.2.5 Mettre à jour les dépendances pour corriger des vulnérabilités ou bénéficier d'améliorations 3.1.2.6 Gérer les dépréciations 3.1.2.7 Travailler avec des logiciels tiers et API 3.1.2.9 Interpréter la documentation et le code source
	Charger les données à l'aide d'un script en Python	3. Programmation 3.3 Bases de données 3.3.2 Construire une base de données non relationnelle	3.3.2.1 Créer des enregistrements 3.3.2.2 Manipuler des enregistrements
		4. Apprentissage automatique 4.1 Prétraitement des données 4.1.1 Préparer des caractéristiques pour des tâches d'apprentissage supervisé ou non supervisé	4.1.1.7 Créer un dictionnaire de données (codebook) pour documenter les hypothèses et interprétations des caractéristiques
		6. Infrastructures de données 6.1 Stockage de données 6.1.1 Manipuler des données stockées dans des fichiers 6.1.2 Manipuler des données stockées dans des bases de données	6.1.1.5 Travailler avec des fichiers manuellement 6.1.1.6 Travailler avec des fichiers par programmation 6.1.1.7 Traiter des données stockées dans différents systèmes de fichiers 6.1.1.8 Agréger des données provenant de différents systèmes de fichiers 6.1.1.9 Gérer de grands ensembles de données qui ne tiennent pas en mémoire 6.1.2.10 Construire des modèles de données à partir de données stockées en bases de données

Étape du pipeline	Tâche	Sujets, thèmes et compétences	Sous-compétences
		6.2 Calcul parallèle 6.2.1 Utiliser des processeurs parallèles pour entraîner des modèles de données	6.2.1.4 Exécuter des programmes parallèles 6.2.1.5 Gérer les journaux et exceptions entre processus 6.2.1.6 Gérer les entrées/sorties fichiers et bases entre processus 6.2.1.7 Gérer les redémarrages de code
		6.3 Informatique 6.3.3 Assurer la sécurité des données et infrastructures	6.3.3.1 Assurer la sécurité des données au repos 6.3.3.5 Utiliser des VPN pour renforcer la sécurité de l'infrastructure 6.3.3.9 Assurer la sécurité des données en transit
Prétraitement des données	Nettoyer les données	1. Données 1.3 Manipulation et traitement des données 1.3.1 Préparer les données pour un projet en apprentissage automatique ou en intelligence artificielle	1.3.1.1 Évaluer la qualité des données 1.3.1.2 Calculer des métriques de qualité 1.3.1.3 Isoler les enregistrements problématiques 1.3.1.4 Améliorer la qualité des données 1.3.1.5 Filtrer les données 1.3.1.6 Imputer les données manquantes (selon le type)
	Convertir les données aux unités et aux formats appropriés	1. Données 1.3 Manipulation et traitement des données 1.3.1 Préparer les données pour un projet en apprentissage automatique ou en intelligence artificielle	1.3.1.11 Forcer la conversion des types de données (ex. dates)
	Fusionner et aligner les données	1. Données 1.3 Manipulation et traitement des données 1.3.1 Préparer les données pour un projet en apprentissage automatique ou en intelligence artificielle 1.3.2 Manipuler les données	1.3.1.12 Appliquer des transformations courantes (groupby, reduce, map, pivot, merge, join) 1.3.2.7 Appliquer des opérations au niveau des lignes

Étape du pipeline	Tâche	Sujets, thèmes et compétences	Sous-compétences
Traitement des données	Organiser les données pour alimenter les algorithmes	1. Données 1.3 Manipulation et traitement des données 1.3.1 Préparer les données pour un projet en apprentissage automatique ou en intelligence artificielle 1.3.2 Manipuler les données	1.3.1.10 Transformer les données 1.3.1.14 Traiter les enjeux de confidentialité 1.3.2.1 Isoler des jeux de données à partir de structures plus larges (requêtes) 1.3.2.2 Filtrer les données par sélection directe (indices) ou conditions logiques 1.3.2.3 Compiler les données par opérations de haut niveau 1.3.2.4 Exploiter des variables de type date/heure
	Appliquer les traitements supplémentaires lorsque requis	1. Données 1.3 Manipulation et traitement des données 1.3.2 Manipuler les données 4. Apprentissage automatique 4.1 Prétraitement des données 4.1.1 Préparer des caractéristiques pour des tâches d'apprentissage supervisé ou non supervisé	1.3.2.5 Calculer de nouvelles propriétés 4.1.1.5 Réduire la dimensionnalité des ensembles de données de grande dimension 4.1.1.6 Calculer des caractéristiques dans des fenêtres temporelles 4.1.1.9 Calculer des matrices de corrélation entre caractéristiques 4.1.1.10 Détecter les valeurs aberrantes parmi les caractéristiques
Appliquer aux données, les étiquettes correspondantes		1. Données 1.3 Manipulation et traitement des données 1.3.1 Préparer les données pour un projet en apprentissage automatique ou en intelligence artificielle	1.3.1.13 Étiqueter les données

Étape du pipeline	Tâche	Sujets, thèmes et compétences	Sous-compétences
	Encoder les données pour la classification	4. Apprentissage automatique 4.1 Prétraitement des données 4.1.1 Préparer des caractéristiques pour des tâches d'apprentissage supervisé ou non supervisé	4.1.1.1 Calculer des caractéristiques pour différents types de données (catégorielles, numériques, séries temporelles) 4.1.1.2 Encoder des données catégorielles 4.1.1.3 Identifier et corriger les erreurs dans les données catégorielles
	Séparer les jeux de données	1. Données 1.3 Manipulation et traitement des données 1.3.2 Manipuler les données	1.3.2.6 Appliquer des opérations au niveau des colonnes
Exploration des données	Visualiser les enregistrements dans des graphiques pour établir le lien entre les données	1. Données 1.4 Représentation des données 1.4.1 Analyser les visualisations graphiques des données	1.4.1.1 Interpréter les données représentées graphiquement 1.4.1.2 Identifier les tendances dans les nuages de points, histogrammes, diagrammes en barres et secteurs, cartes thermiques 1.4.1.5 Générer des visualisations courantes (nuages de points, histogrammes, diagrammes en barres et secteurs, cartes thermiques) 1.4.1.6 Générer des visualisations personnalisées
	Afficher les données en fonction des caractéristiques pour visualiser la classification	1. Données 1.4 Représentation des données 1.4.2 Analyser les représentations tabulaires des données	1.4.2.1 Interpréter les données présentées sous forme de tableau 1.4.2.2 Expliquer les mesures statistiques courantes (tendance centrale, dispersion, quartiles, extrêmes) 1.4.2.3 Interpréter la signification des mesures statistiques dans le contexte d'une question de recherche 1.4.2.4 Créer un rapport des propriétés statistiques 1.4.2.5 Calculer des mesures de tendance centrale, dispersion, quartiles et extrêmes 1.4.2.6 Calculer d'autres propriétés (comptages, sommes, fréquences, probabilités, covariances) 1.4.2.7 Identifier la pertinence des mesures statistiques selon le domaine et le type de données (prise en compte des distributions, valeurs aberrantes, etc.)

Étape du pipeline	Tâche	Sujets, thèmes et compétences	Sous-compétences
		4. Apprentissage automatique 4.1 Prétraitement des données 4.1.1 Préparer des caractéristiques pour des tâches d'apprentissage supervisé ou non supervisé	4.1.1.8 Évaluer les caractéristiques pour une utilisation dans des modèles ML 4.1.1.11 Mesurer l'importance des caractéristiques à partir de modèles ML
Préparation des données	Normaliser, standardiser et mélanger aléatoirement les données	4. Apprentissage automatique 4.1 Prétraitement des données 4.1.1 Préparer des caractéristiques pour des tâches d'apprentissage supervisé ou non supervisé 4.1.2 Établir des pipelines de données	4.1.1.4 Normaliser / standardiser des caractéristiques 4.1.2.2 Utiliser des structures de données natives des bibliothèques de ML 4.1.2.3 Rééchantillonner de grands ensembles de données
	Séparer les données en jeux d'entraînement et de test	4. Apprentissage automatique 4.2 Apprentissage supervisé 4.2.1 Gérer un cadre d'apprentissage supervisé	4.2.1.1 Diviser les données en ensembles d'entraînement, test et validation
Sélections, validation et ajustement du modèle	Entraîner le modèle	4. Apprentissage automatique 4.1 Prétraitement des données 4.1.2 Établir des pipelines de données 4.3 Apprentissage non supervisé	4.1.2.1 Connecter des sources de données aux modèles 4.3.2.1 Paramétrer et appliquer des méthodes de regroupement (clustering) 4.3.2.2 Comparer différentes méthodes de clustering

Étape du pipeline	Tâche	Sujets, thèmes et compétences	Sous-compétences
		<p>4.3.2 Appliquer l'apprentissage non supervisé à des tâches spécifiques</p>	<p>4.3.2.3 Sélectionner une méthode de clustering adaptée à la tâche 4.3.2.4 Paramétrer une méthode de clustering 4.3.2.5 Utiliser des bibliothèques de ML pour le clustering 4.3.2.6 Expérimenter avec des ensembles de données pour le clustering</p>
	Tester le modèle	<p>4. Apprentissage automatique 4.2 Apprentissage supervisé 4.2.1 Gérer un cadre d'apprentissage supervisé</p>	<p>4.2.1.2 Appliquer la validation croisée k-fold 4.2.1.8 Ajuster les hyperparamètres des méthodes de classification et régression 4.2.1.10 Appliquer des méthodes d'optimisation 4.2.1.11 Gérer le déséquilibre des classes 4.2.1.12 Rééchantillonner l'ensemble d'entraînement pour équilibrer les classes 4.2.1.13 Simuler des entrées pour les classes minoritaires 4.2.1.14 Ajuster les poids des classes dans les méthodes de classification 4.2.2.1 Paramétrer et appliquer des méthodes de classification</p>
		<p>4.2.2 Appliquer l'apprentissage supervisé à des tâches spécifiques</p>	<p>4.2.2.3 Sélectionner une méthode de classification adaptée à la tâche 4.2.2.4 Paramétrer une méthode de classification 4.2.2.7 Utiliser des bibliothèques de ML pour la classification 4.2.2.8 Expérimenter avec des tâches de classification courantes 4.2.2.9 Déterminer le biais induit de la méthode choisie</p>
	Utiliser un rapport de classification pour calculer les performances du modèle	<p>4. Apprentissage automatique 4.3 Apprentissage non supervisé 4.3.1 Gérer un cadre d'apprentissage non supervisé</p>	<p>4.3.1.1 Appliquer les bonnes mesures de performance 4.3.1.2 Identifier les mesures appropriées 4.3.1.3 Évaluer la performance du modèle</p>
	Ajuster le modèle pour améliorer la performance	<p>4. Apprentissage automatique 4.2 Apprentissage supervisé 4.2.1 Gérer un cadre d'apprentissage supervisé</p>	<p>4.2.1.5 Appliquer les bonnes mesures de performance pour la régression et la classification binaire ou multi-classes 4.2.1.6 Identifier les mesures appropriées</p>

Étape du pipeline	Tâche	Sujets, thèmes et compétences	Sous-compétences
	Utiliser les graphiques de pertes et des tableaux pour déterminer si le modèle généralise bien les apprentissages	4. Apprentissage automatique 4.2 Apprentissage supervisé 4.2.1 Gérer un cadre d'apprentissage supervisé	4.2.1.7 Évaluer la performance du modèle
	Utiliser des graphiques et des tableaux pour représenter la distribution des erreurs et visualiser la répartition des prédictions	4. Apprentissage automatique 4.2 Apprentissage supervisé 4.2.2 Appliquer l'apprentissage supervisé à des tâches spécifiques	4.2.2.2 Comparer différentes méthodes de classification
Inférences à partir du modèle	Utiliser le modèle pour valider son fonctionnement avec de nouvelles données	5. Apprentissage profond 5.1 Écosystèmes de l'IA 5.1.4 Intégrer des systèmes d'IA dans des contextes réels	5.1.4.1 Évaluer les systèmes d'IA pour des applications spécifiques
Prise de décision	À partir des données et des prédictions obtenues, décider des actions à poser	5. Apprentissage profond 5.1 Écosystèmes de l'IA 5.1.4 Intégrer des systèmes d'IA dans des contextes réels 7. Bibliothèques et environnements de développement 7.1 Outils 7.1.6 Personnaliser l'outil pour un objectif spécifique	5.1.4.3 Identifier le rôle de l'humain (Human-in-the-Loop) et savoir quand lui déléguer la décision (par ex. si le système IA est incertain) 5.1.4.4 Évaluer la pertinence et la représentativité des données synthétiques afin d'éviter les biais (avec consultation d'experts) 7.1.6.1 Personnaliser l'outil pour un objectif spécifique
	Valider la qualité des informations et réagir en cas de doute	1. Données 1.4 Représentation des données	1.4.1.3 Interpréter la signification des représentations graphiques en lien avec une question de recherche 1.4.1.4 Présenter les données de façon compréhensible

Étape du pipeline	Tâche	Sujets, thèmes et compétences	Sous-compétences
		<p>1.4.1 Analyser les visualisations graphiques des données</p> <p>5. Apprentissage profond 5.1 Écosystèmes de l'IA 5.1.4 Intégrer des systèmes d'IA dans des contextes réels</p>	<p>5.1.4.2 Évaluer les mesures choisies par rapport aux heuristiques ou autres métriques établies en consultant les experts et parties prenantes</p>
	<p>Proposer des améliorations ou des changements au besoin</p>	<p>7. Bibliothèques et environnements de développement 7.1 Outils 7.1.7 Proposer des améliorations pour l'outil</p>	<p>7.1.7.1 Proposer des améliorations pour l'outil</p>

ANNEXE III

TABLEAU DES COMPÉTENCES EN IA ASSOCIÉES AUX COURS DU PROGRAMME DE TMI AU CVM

Tableau-A III-1 - Compétences en IA associées aux cours du programme de TMI au CVM

No cours	Nom du cours	Contenu associé à l'IA	Compétences en IA
241-M11-VM	Métrologie et capteurs	Établir des pipelines de données	4.1.2.1
201-M20-VM	Mathématiques II	Analyser les représentations tabulaires des données Analyser les visualisations graphiques des données	1.4.1, 1.4.2
241-X4X-VM	Cours à ajouter : Programmation et science des données	Écrire du code en respectant la syntaxe et la structure Intégrer des bibliothèques Utiliser des processeurs parallèles pour entraîner des modèles de données Préparer les données pour un projet en apprentissage automatique Préparer des caractéristiques pour des tâches d'apprentissage supervisé ou non supervisé	3.1.1, 3.1.2, 6.2.1, 1.3.1, 1.3.2, 4.1.1, 1.3.1.13, 4.2.1.1
241-D45-VM	Gestion de la maintenance industrielle I	Construire une base de données non relationnelle Manipuler des données stockées dans des fichiers Manipuler des données stockées dans des bases de données	3.3.2, 6.1.1, 6.1.2
241-D65-VM	Projet en fiabilité	Analyser les usages typiques des données en apprentissage automatique Préparer des caractéristiques pour des tâches d'apprentissage supervisé ou non supervisé Assurer la sécurité des données et infrastructures	1.1.2.11, 4.1.1.7, 6.3.3

No cours	Nom du cours	Contenu associé à l'IA	Compétences en IA
241-X6X-VM	Cours à ajouter en session 6 : Projet d'application d'apprentissage automatique en maintenance	<p>Analyser les usages typiques des données en apprentissage automatique</p> <p>Identifier la relation entre l'IA et les systèmes informatiques</p> <p>Sélectionner l'outil approprié</p> <p>Exécuter des exemples génériques avec l'outil</p> <p>Intégrer des systèmes d'IA dans des contextes réels</p> <p>Personnaliser l'outil pour un objectif spécifique</p> <p>Proposer des améliorations pour l'outil</p>	<p>1.1.1, 1.1.2, 5.1.2, 7.1.1, 7.1.4, 7.1.5, 4.1.2.2, 4.1.2.3, 5.1.4, 7.1.6, 1.4.1.3, 1.4.1.4, 7.1.7</p>
	Démonstration de cas d'application de l'apprentissage automatique (combiner avec le projet d'application)	<p>Appliquer l'apprentissage non supervisé à des tâches spécifiques</p> <p>Gérer un cadre d'apprentissage supervisé</p> <p>Appliquer l'apprentissage supervisé à des tâches spécifiques</p> <p>Gérer un cadre d'apprentissage non supervisé</p>	<p>4.3.2, 4.2.1, 4.2.2, 4.3.1</p>

ANNEXE IV

GRILLE DIAGNOSTIQUE DES COMPÉTENCES NÉCESSAIRES

Tableau-A IV-1 - Grille diagnostique des compétences nécessaires

Étape du pipeline	Tâche / Outil / Technologie	Sous-compétences	Balises de contenu, activités d'apprentissage
Définition du besoin : <i>Utiliser les données pour déterminer les conditions d'opération des équipements et planifier les interventions</i>	Identifier clairement le problème à régler : <i>Alerter le technicien lorsqu'une intervention est requise.</i>	1.1.1.10 Identifier les effets de la manipulation des données sur leur représentation (par ex. suppression de valeurs aberrantes) 1.1.1.11 Sélectionner des types et structures de données appropriés selon le contexte 1.1.1.13 Adapter le schéma de données aux exigences du projet et au cas d'utilisation visé 1.1.2.4 Évaluer si les données respectent les exigences éthiques (avec l'aide d'experts de domaine et de sciences sociales) 1.1.2.8 Lister les applications du ML et de l'IA dans différents contextes 1.1.2.10 Évaluer l'utilité d'un jeu de données pour le développement d'un modèle ML ou IA spécifique 1.1.2.13 Évaluer les hypothèses des jeux de données acquis (seconde main, fournisseurs) quant à leur collecte, traitement et distribution 1.1.2.14 Déterminer la lignée (provenance) des données qui ne sont pas de première main	Identifier les défaillances que l'on souhaite être en mesure de détecter Faire une AMDEC Identifier les données à collecter, fréquence et disponibilité, durée de la campagne de collecte Déterminer comment les données seront sauvegardée et comment elles seront accessibles Déterminer le type d'application du ML qui s'applique au besoin
Identifier les techniques d'analyses applicables : <i>Détection d'anomalies</i>	Identifier les techniques normalement utilisées pour le type problème à résoudre, comparativement aux solutions utilisant le ML	5.1.2.1 Distinguer les artefacts technologiques qui utilisent ou non l'IA 5.1.2.2 Caractériser les ingrédients technologiques clés d'un projet d'IA réussi	

Étape du pipeline	Tâche / Outil / Technologie	Sous-compétences	Balises de contenu, activités d'apprentissage
		<p>5.1.2.3 Comparer l'IA et les approches TI traditionnelles à divers problèmes</p> <p>5.1.3.1 Analyser les caractéristiques qui rendent une entité intelligente</p> <p>5.1.3.2 Comparer l'intelligence artificielle et l'intelligence humaine sur divers problèmes</p> <p>5.1.3.3 Positionner la nécessité des compétences humaines dans un écosystème IA</p> <p>5.1.3.4 Discuter de la relation entre modèles d'IA et biologie</p>	<p>Identifier les risques et opportunités liés à l'utilisation du ML</p> <p>Identifier les avantages attendus du projet</p> <p>Identifier le rôle de l'humain face à l'utilisation de la solution et comment la solution va l'aider</p> <p>Identifier les forces et les faiblesses de l'humain face à la solution envisagée</p>
	<p>Sélectionner les outils :</p> <p><i>Identifier les modes de défaillances et les capteurs nécessaires à l'aide d'une AMDEC</i></p>	<p>7.1.1.1 Sélectionner l'outil approprié</p> <p>7.1.4.1 Trouver la documentation de l'outil</p> <p>7.1.5.1 Exécuter des exemples génériques avec l'outil</p>	<p>Parmi les solutions de ML, identifier lesquelles sont applicables au projet</p> <p>Consulter la documentation sur le fonctionnement de l'outil et les requis</p> <p>Tester des exemples d'application en lien avec les outils et le problème</p>
Acquisition des données	<p>Attribuer des étiquettes pour la classification des données :</p> <p><i>Enregistrer les conditions de fonctionnement (étiquettes) dans un registre</i></p>	<p>1.1.2.11 Constituer un jeu de données original pour l'entraînement et les tests d'un modèle ML ou IA</p>	<p>Déterminer les conditions qui doivent être détectées par l'outil de ML</p> <p>Déterminer comment seront segmentés les données une fois collectées</p> <p>Déterminer la stratégie de croisement des données et des étiquettes</p>

Étape du pipeline	Tâche / Outil / Technologie	Sous-compétences	Balises de contenu, activités d'apprentissage
	<p>Enregistrer les données dans des fichiers csv</p>	<p>3.1.1.8 Travailler avec des fichiers</p> <p>3.1.1.9 Lire des fichiers depuis des sources locales ou distantes</p> <p>3.1.1.10 Écrire des fichiers dans un stockage local</p> <p>3.1.1.1 Utiliser divers types et structures de données</p> <p>3.1.1.2 Réaliser des opérations avec des types numériques et des chaînes de caractères</p> <p>3.1.1.3 Réaliser des opérations avec des types mixtes impliquant des conversions</p> <p>3.1.1.4 Manipuler des structures de type tableau (instancier, itérer, mettre à jour, filtrer, mapper, réduire, découper, copier)</p> <p>3.1.1.5 Manipuler des structures de type objet (instancier, itérer, mettre à jour, filtrer, mapper, réduire, copier)</p> <p>3.1.1.6 Différencier les types mutables et immuables</p> <p>3.1.1.11 Appliquer la logique de contrôle d'exécution</p> <p>3.1.1.12 Itérer sur des structures de données</p> <p>3.1.1.13 Utiliser des boucles et conditions pour contrôler l'exécution</p> <p>3.1.1.14 Utiliser des fonctions et méthodes</p>	<p>Manipuler des fichiers de données à partir d'un code informatique</p> <p>Accéder à des données dans des fichiers locaux, sur des serveurs, dans un service informatique, dans un PLC, service web, etc.</p> <p>Manipuler des données catégorielles, continues, temporelles, textuelles, etc.</p> <p>Utiliser des bases de données, des fichiers, etc.</p> <p>Convertir des données dans différents types</p> <p>Filtrer, trier, séparer, mettre à jour des données</p>

Étape du pipeline	Tâche / Outil / Technologie	Sous-compétences	Balises de contenu, activités d'apprentissage
		<p>3.1.1.15 Intégrer de la logique dans des fonctions</p> <p>3.1.1.17 Appliquer la portée (scope) dans la construction de fermetures</p> <p>3.1.1.18 Composer des opérations par chaînage</p> <p>3.1.1.19 Écrire des fonctions pour l'exécution asynchrone</p> <p>3.1.1.20 Utiliser des modèles de programmation courants</p>	
	Utiliser les bibliothèques disponibles	<p>3.1.2.1 Gérer les dépendances d'un projet</p> <p>3.1.2.2 Installer des dépendances (pip, apt, etc.)</p> <p>3.1.2.3 Documenter les dépendances</p> <p>3.1.2.4 Résoudre les conflits de dépendances</p> <p>3.1.2.5 Mettre à jour les dépendances pour corriger des vulnérabilités ou bénéficier d'améliorations</p> <p>3.1.2.6 Gérer les dépréciations</p> <p>3.1.2.7 Travailler avec des logiciels tiers et API</p> <p>3.1.2.9 Interpréter la documentation et le code source</p>	<p>Utiliser des bibliothèques de traitement des données</p> <p>Installation des bibliothèques et des dépendances</p> <p>Introduction à linux, Python, cmd, etc.</p> <p>Utiliser des API</p> <p>Utiliser des applications open-source</p>
	Charger les données à l'aide d'un script en Python	<p>3.3.2.1 Créer des enregistrements</p> <p>3.3.2.2 Manipuler des enregistrements</p>	Charger de grandes quantités de données à partir de différentes sources

Étape du pipeline	Tâche / Outil / Technologie	Sous-compétences	Balises de contenu, activités d'apprentissage
		<p>4.1.1.7 Créer un dictionnaire de données (codebook) pour documenter les hypothèses et interprétations des caractéristiques</p> <p>6.1.1.5 Travailler avec des fichiers manuellement</p> <p>6.1.1.6 Travailler avec des fichiers par programmation</p> <p>6.1.1.7 Traiter des données stockées dans différents systèmes de fichiers</p> <p>6.1.1.8 Agréger des données provenant de différents systèmes de fichiers</p> <p>6.1.1.9 Gérer de grands ensembles de données qui ne tiennent pas en mémoire</p> <p>6.1.2.10 Construire des modèles de données à partir de données stockées en bases de données</p> <p>6.2.1.4 Exécuter des programmes parallèles</p> <p>6.2.1.5 Gérer les journaux et exceptions entre processus</p> <p>6.2.1.6 Gérer les entrées/sorties fichiers et bases entre processus</p> <p>6.2.1.7 Gérer les redémarrages de code</p> <p>6.3.3.1 Assurer la sécurité des données au repos</p> <p>6.3.3.5 Utiliser des VPN pour renforcer la sécurité de l'infrastructure</p>	<p>Croiser des données de différentes sources</p> <p>Étiqueter des données</p> <p>Manipuler des données à l'aide d'un script</p> <p>Gérer les erreurs de chargement et d'exécution</p> <p>Manipuler les données de façon sécuritaire</p>

Étape du pipeline	Tâche / Outil / Technologie	Sous-compétences	Balises de contenu, activités d'apprentissage
		6.3.3.9 Assurer la sécurité des données en transit	
Prétraitement des données	Nettoyer les données	1.3.1.1 Évaluer la qualité des données 1.3.1.2 Calculer des métriques de qualité 1.3.1.3 Isoler les enregistrements problématiques 1.3.1.4 Améliorer la qualité des données 1.3.1.5 Filtrer les données 1.3.1.6 Imputer les données manquantes (selon le type)	Analyser les données comparativement aux attentes, format, valeurs, quantité, etc. Explorer les statistiques des jeux de données Améliorer des jeux de données Utiliser différentes stratégies pour gérer les données manquantes
Traitement des données	Convertir les données aux unités et aux formats appropriés Fusionner et aligner les données Organiser les données pour alimenter les algorithmes	1.3.1.11 Forcer la conversion des types de données (ex. dates) 1.3.1.12 Appliquer des transformations courantes (groupby, reduce, map, pivot, merge, join) 1.3.1.10 Transformer les données 1.3.1.14 Traiter les enjeux de confidentialité 1.3.2.1 Isoler des jeux de données à partir de structures plus larges (requêtes) 1.3.2.2 Filtrer les données par sélection directe (indices) ou conditions logiques	Faire la conversion de données et des regroupements Fusionner des jeux de données et manipuler des tables de données Organiser les jeux de données en fonction des modèles et des besoins Séparer les données pour différents besoins Manipuler des données temporelles

Étape du pipeline	Tâche / Outil / Technologie	Sous-compétences	Balises de contenu, activités d'apprentissage
		1.3.2.3 Compiler les données par opérations de haut niveau 1.3.2.4 Exploiter des variables de type date/heure	
	Appliquer les traitements supplémentaires lorsque requis : <i>Appliquer le traitement du signal vibratoire</i>	1.3.2.5 Calculer de nouvelles propriétés 4.1.1.5 Réduire la dimensionnalité des ensembles de données de grande dimension 4.1.1.6 Calculer des caractéristiques dans des fenêtres temporelles 4.1.1.9 Calculer des matrices de corrélation entre caractéristiques 4.1.1.10 Détecter les valeurs aberrantes parmi les caractéristiques	Calculer les spectres Calculer les descripteurs statistiques du signal Augmenter les données à l'aide de valeurs calculées Sélectionner les données les plus pertinentes Effectuer des opérations sur des données temporelles Détecter les données aberrantes
	Appliquer aux données, les étiquettes correspondantes aux conditions de fonctionnement	1.3.1.13 Étiqueter les données	Appliquer les étiquettes aux données
Encoder les données pour la classification : <i>Encoder les conditions de fonctionnement pour la détection d'anomalies</i>		4.1.1.1 Calculer des caractéristiques pour différents types de données (catégorielles, numériques, séries temporelles) 4.1.1.2 Encoder des données catégorielles	Encoder des données pour faire de la classification

Étape du pipeline	Tâche / Outil / Technologie	Sous-compétences	Balises de contenu, activités d'apprentissage
		<p>4.1.1.3 Identifier et corriger les erreurs dans les données catégorielles</p>	
Exploration des données	<p>Préparer et séparer les jeux de données</p> <p><i>Jeu de données normales et jeu de données anormales</i></p> <p>Visualiser les enregistrements dans des graphiques pour établir le lien entre les données et le système</p>	<p>1.3.2.6 Appliquer des opérations au niveau des colonnes</p> <p>1.4.1.1 Interpréter les données représentées graphiquement</p> <p>1.4.1.2 Identifier les tendances dans les nuages de points, histogrammes, diagrammes en barres et secteurs, cartes thermiques</p> <p>1.4.1.5 Générer des visualisations courantes (nuages de points, histogrammes, diagrammes en barres et secteurs, cartes thermiques)</p> <p>1.4.1.6 Générer des visualisations personnalisées</p>	<p>Conserver les caractéristiques pertinentes</p> <p>Générer des graphiques pour visualiser les caractéristiques des données</p> <p>Extraire de l'information à partir des graphiques</p>
	<p>Sélectionner les données les plus pertinentes</p> <p>Afficher les données en fonction des caractéristiques pour visualiser la classification</p>	<p>1.4.2.1 Interpréter les données présentées sous forme de tableau</p> <p>1.4.2.2 Expliquer les mesures statistiques courantes (tendance centrale, dispersion, quartiles, extrêmes)</p> <p>1.4.2.3 Interpréter la signification des mesures statistiques dans le contexte d'une question de recherche</p> <p>1.4.2.4 Créer un rapport des propriétés statistiques</p>	<p>Calculer et interpréter les statistiques des données</p> <p>Matrice de corrélation</p> <p>AMDEC</p> <p>Nuages de point, PCA, t-SNE pour visualiser la séparation des données</p>

Étape du pipeline	Tâche / Outil / Technologie	Sous-compétences	Balises de contenu, activités d'apprentissage
Préparation des données	Normaliser, standardiser et mélanger aléatoirement les données	<p>1.4.2.5 Calculer des mesures de tendance centrale, dispersion, quartiles et extrêmes</p> <p>1.4.2.6 Calculer d'autres propriétés (comptages, sommes, fréquences, probabilités, covariances)</p> <p>1.4.2.7 Identifier la pertinence des mesures statistiques selon le domaine et le type de données (prise en compte des distributions, valeurs aberrantes, etc.)</p> <p>4.1.1.8 Évaluer les caractéristiques pour une utilisation dans des modèles ML</p> <p>4.1.1.11 Mesurer l'importance des caractéristiques à partir de modèles ML</p>	
Séparation des données en jeux d'entraînement et de test	Normaliser, standardiser et mélanger aléatoirement les données	<p>4.1.1.4 Normaliser / standardiser des caractéristiques</p> <p>4.1.2.2 Utiliser des structures de données natives des bibliothèques de ML</p> <p>4.1.2.3 Rééchantillonner de grands ensembles de données</p>	Appliquer la normalisation, standardisation et autres étapes de préparation des données
Sélections, validation et ajustement du modèle	Séparer les données en jeux d'entraînement et de test	<p>4.2.1.1 Diviser les données en ensembles d'entraînement, test et validation</p>	Séparer les données en jeux d'entraînement et de test
Sélections, validation et ajustement du modèle	Définir le modèle : Définition de la structure et des hyperparamètres	<p>4.1.2.1 Connecter des sources de données aux modèles</p> <p>4.3.2.1 Paramétrer et appliquer des méthodes de regroupement (clustering)</p>	Présenter différents modèles en fonction des besoins Exécuter des démonstrations de l'utilisation des modèles

Étape du pipeline	Tâche / Outil / Technologie	Sous-compétences	Balises de contenu, activités d'apprentissage
	<p>Entraîner le modèle :</p> <p><i>Entraîner un autoencodeur à reconnaître l'état de fonctionnement normal</i></p>	<p>4.3.2.2 Comparer différentes méthodes de clustering</p> <p>4.3.2.3 Sélectionner une méthode de clustering adaptée à la tâche</p> <p>4.3.2.4 Paramétrer une méthode de clustering</p> <p>4.3.2.5 Utiliser des bibliothèques de ML pour le clustering</p> <p>4.3.2.6 Expérimenter avec des ensembles de données pour le clustering</p>	<p>Solution AutoMLs</p>
	<p>Tester le modèle :</p> <p><i>Tester l'autoencodeur pour déterminer s'il différencie les états de fonctionnements identifiés</i></p>	<p>4.2.1.2 Appliquer la validation croisée k-fold</p> <p>4.2.1.8 Ajuster les hyperparamètres des méthodes de classification et régression</p> <p>4.2.1.10 Appliquer des méthodes d'optimisation</p> <p>4.2.1.11 Gérer le déséquilibre des classes</p> <p>4.2.1.12 Rééchantillonner l'ensemble d'entraînement pour équilibrer les classes</p> <p>4.2.1.13 Simuler des entrées pour les classes minoritaires</p> <p>4.2.1.14 Ajuster les poids des classes dans les méthodes de classification</p> <p>4.2.2.1 Paramétrer et appliquer des méthodes de classification</p>	<p>Exécuter des démonstrations de l'utilisation des modèles</p> <p>Tester des solutions AutoML</p>

Étape du pipeline	Tâche / Outil / Technologie	Sous-compétences	Balises de contenu, activités d'apprentissage
		<p>4.2.2.3 Sélectionner une méthode de classification adaptée à la tâche</p> <p>4.2.2.4 Paramétrer une méthode de classification</p> <p>4.2.2.7 Utiliser des bibliothèques de ML pour la classification</p> <p>4.2.2.8 Expérimenter avec des tâches de classification courantes</p> <p>4.2.2.9 Déterminer le biais inductif de la méthode choisie</p>	
	Utiliser un rapport de classification pour calculer les performances du modèle	<p>4.3.1.1 Appliquer les bonnes mesures de performance</p> <p>4.3.1.2 Identifier les mesures appropriées</p> <p>4.3.1.3 Évaluer la performance du modèle</p>	<p>Présenter un rapport de classification.</p> <p>Lire et comprendre un rapport</p> <p>Évaluer la performance du modèle</p>
	<p>Ajuster le modèle pour améliorer la performance</p> <p>Ajuster le seuil de détection pour améliorer la performance</p>	<p>4.2.1.5 Appliquer les bonnes mesures de performance pour la régression et la classification binaire ou multi-classes</p> <p>4.2.1.6 Identifier les mesures appropriées</p>	Démonstration
	Utiliser les graphiques de de pertes et des tableaux pour déterminer si le modèle généralise bien les apprentissages	4.2.1.7 Évaluer la performance du modèle	Démonstration

Étape du pipeline	Tâche / Outil / Technologie	Sous-compétences	Balises de contenu, activités d'apprentissage
	Valider le modèle à l'aide de nouvelles données		
	Utiliser des graphiques et des tableaux pour représenter la distribution des erreurs et visualiser la répartition des prédictions	4.2.2.2 Comparer différentes méthodes de classification	Utiliser des visualisations pour les erreurs
Inférences à partir du modèle	Utiliser le modèle pour valider son fonctionnement avec de nouvelles données : <i>Utiliser un autoencodeur pour déterminer l'état de fonctionnement</i>	5.1.4.1 Évaluer les systèmes d'IA pour des applications spécifiques	Utiliser le modèle en production
Prise de décision	À partir des données et des prédictions obtenues, décider des actions à poser : <i>Le technicien est en mesure de déterminer si une intervention est nécessaire ou non</i>	5.1.4.3 Identifier le rôle de l'humain (Human-in-the-Loop) et savoir quand lui déléguer la décision (par ex. si le système IA est incertain) 5.1.4.4 Évaluer la pertinence et la représentativité des données synthétiques afin d'éviter les biais (avec consultation d'experts) 7.1.6.1 Personnaliser l'outil pour un objectif spécifique	Déterminer comment la solution aide le technicien dans son travail Évaluer si le modèle permet d'atteindre les objectifs

Étape du pipeline	Tâche / Outil / Technologie	Sous-compétences	Balises de contenu, activités d'apprentissage
	Valider la qualité des informations et réagir en cas de doute	1.4.1.3 Interpréter la signification des représentations graphiques en lien avec une question de recherche 1.4.1.4 Présenter les données de façon compréhensible	Vérifier si les résultats sont conformes aux attentes et au fonctionnement Fournir les résultats dans un format approprié
Proposer des améliorations ou des changements au besoin		5.1.4.2 Évaluer les mesures choisies par rapport aux heuristiques ou autres métriques établies en consultant les experts et parties prenantes 7.1.7.1 Proposer des améliorations pour l'outil	Proposer des améliorations suite à l'utilisation

APPENDICES

Appendice A - AMDEC de la pompe

Mode de défaillance	Cause	Effet de la défaillance	Mode de détection	Probabilité d'occurrence (P)	Sévérité (S)	Probabilité de non-détection (D)	Criticité (P * S * D)	Indicateur d'état système	
de débit	Usure des pièces internes, blocage partiel à l'aspiration	Réduction du rendement, cavitation	Capteur de débit	6	8	3	144	Surveillance requise	
tion excessive	Déséquilibre de l'impulseur	Dommmages aux autres composants	Capteur de vibration	5	7	4	140	Planifier une interv	
quilibre rotor	Usure ou défauts de montage	Vibration excessive, usure prématurée	Capteur de vibration	5	7	4	140	Planifier une interv	
ant élevé	Court-circuit, surcharge, blocage partiel au refoulement	Surchauffe, panne mécanique	Capteur de courant	5	9	3	135	Surveillance requise	
ement ou coussinet	Frottement, lubrification insuffisante, corrosion	Bruit anormal, vibrations, usure rapide	Capteur de vibration	4	8	4	128	Surveillance requise	
seur usé	Érosion, cavitation	Réduction du débit, Réduction du rendement	Capteur de débit	5	8	3	120	Surveillance requise	
ation	Hauteur de succion inadéquate, air dans le circuit	Dommmages à l'impulseur, perte de performance	Capteur de vibration	3	8	4	96	Planifier une interv	
d'étanchéité tueux	Dégradation des joints	Fuite d'eau	Capteur de pression	3	7	4	84	Planifier une interv	
de Pression	Fuite des joints	Performance réduite	Capteur de pression	3	7	4	84	Planifier une interv	
chauffe	Frottement, lubrification insuffisante	Pannes mécaniques	Capteur de température	4	9	2	72	Planifier une interv	
se de rotation male	Problèmes d'alimentation, variation de la charge	Usure prématurée du moteur	Capteur de vitesse	4	6	2	48	Surveillance requise	
ge de la pompe	Corp étranger	Arrêt complet de la circulation	Capteur de vibration	2	9	1	18	Défaillance en cours	
ou conduits mal	Mauvais serrage, vibration	Fuite d'eau	Capteur de pression	3	4	1	12	Planifier une interv	
ne données ou ses éronées	Capteur défectueux, capteur débranché, mauvais montage, ajustement ou blocage	Erreur de lecture	Mauvaise lecture au capteur	4	2	1	8	Planifier une interv	
		l'occurrence de la défaillance. Échelle de 1 (moins probable) à 10 (plus probable).						Fonctionnem	
		faillance sur le système. Échelle de 1 (moins sévère) à 10 (plus sévère).						Code d'état	
		détection de la défaillance. Échelle de 1 (facilement détectable) à 10 (difficile à détecter).						État du système	
								Système ok	
								Surveillance requise	
								Planifier une intervention	
								Défaillance en cours	
								Normal	
								Anormal	
								Anormal	
								Anormal	

LISTE DE RÉFÉRENCES BIBLIOGRAPHIQUES

- Achouch, Mounia, Mariya Dimitrova, Khaled Ziane, Sasan Sattarpanah Karganroudi, Rizck Dhouib, Hussein Ibrahim et Mehdi Adda. 2022. « On Predictive Maintenance in Industry 4.0: Overview, Models, and Challenges ». *Applied Sciences*, vol. 12, n° 16, p. 8081. <<https://doi.org/10.3390/app12168081>>.
- Adari, Suman Kalyan et Sridhar Alla. 2024. *Beginning Anomaly Detection Using Python-Based Deep Learning: Implement Anomaly Detection Applications with Keras and PyTorch*. Berkeley, CA : Apress. <<https://doi.org/10.1007/979-8-8688-0008-5>>. Consulté le 10 novembre 2024.
- Amiral Technologies. 2022. « La maintenance prescriptive, qu'est ce-que c'est ? » In *Amiral Technologies*. <<https://www.amiraltechnologies.com/actualites/blog/la-maintenance-prescriptive-quest-ce-que-cest/>>. Consulté le 27 août 2024.
- Belcic, Ivan et Cole Stryker. 2025. « Qu'est-ce qu'AutoML ? » <<https://www.ibm.com/fr-fr/think/topics/automl>>. Consulté le 18 mars 2025.
- Biggio, Luca et Iason Kastanis. 2020. « Prognostics and Health Management of Industrial Assets: Current Progress and Road Ahead ». *Frontiers in Artificial Intelligence*, vol. 3, p. 578613. <<https://doi.org/10.3389/frai.2020.578613>>.
- Blok, Sherry, Joel Trudeau et Robert Cassidy. 2021. *Artificial Intelligence Competency Framework*. <https://poleia.quebec/en/wp-content/uploads/2022/07/PIA_ConcordiaDawson_AICompetencyFramework.pdf>.
- Compétences Québec. 2024. « InforouteFTP - Technologie de maintenance industrielle ». In *inforouteftp.org*. <<https://www.inforouteftp.org/formation-technique/diplome-etudes-collegiales/323>>. Consulté le 27 août 2024.
- Costa, Victor L. L., Björn Eberhardt, Jiahang Chen et Jürgen Roßmann. 2023. « Towards Predictive Maintenance: an Edge-based Vibration Monitoring System in Industry 4.0 ». In *2023 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS)*. (Erode, India, 23 mars 2023), p. 1430-1437. IEEE. <<https://doi.org/10.1109/ICSCDS56580.2023.10104879>>. Consulté le 12 août 2024.
- Datacamp. 2024. « A Guide to the DBSCAN Clustering Algorithm ». <<https://www.datacamp.com/tutorial/dbscan-clustering-algorithm>>. Consulté le 20 novembre 2025.
- Géron, Aurélien. 2023. *Hands-on machine learning with Scikit-Learn, Keras and TensorFlow: concepts, tools, and techniques to build intelligent systems*, Third edition. Beijing : O'Reilly Media, Inc, 834 p.

- Hydraulic Institute. 2013. *Rotary Pump Guideline*.
- ITMI. 2021. *TECHNICIEN DE MAINTENANCE INDUSTRIELLE : Constats et perspectives de la profession au sein des entreprises québécoises*. Institut technologique de maintenance industrielle, 56 p.
- Jablonski, Adam. 2021. *Condition Monitoring Algorithms in MATLAB®*. Coll. « Springer Tracts in Mechanical Engineering ». Cham : Springer International Publishing. <<https://doi.org/10.1007/978-3-030-62749-2>>. Consulté le 10 septembre 2024.
- Knauer, Tom. 2024. « Identify Failures Before They Happen: The Potential-Functional Curve ». In *Balluf*. <<https://www.balluff.com/en-us/blog/identify-failures-before-they-happen-the-potential-functional-curve>>. Consulté le 10 mars 2026.
- Lord, Mansoureh et Adam Kaplan. 2021. « Mechanical Anomaly Detection on an Embedded Microcontroller ». In *2021 International Conference on Computational Science and Computational Intelligence (CSCI)*. (Las Vegas, NV, USA, décembre 2021), p. 562-568. IEEE. <<https://doi.org/10.1109/CSCI54926.2021.00159>>. Consulté le 17 janvier 2025.
- Maasoum, Seyyed Mahdi Hosseini, Alireza Mostafavi et Ali Sadighi. 2020. « An Autoencoder-Based Algorithm for Fault Detection of Rotating Machines, Suitable for Online Learning and Standalone Applications ». In *2020 6th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS)*. (Mashhad, Iran, 23 décembre 2020), p. 1-6. IEEE. <<https://doi.org/10.1109/ICSPIS51611.2020.9349574>>. Consulté le 4 février 2025.
- Mobley, R. Keith. 2002. *An introduction to predictive maintenance*, 2nd ed. Amsterdam New York : Butterworth-Heinemann.
- Mostafavi, Alireza et Ali Sadighi. 2021. « A Novel Online Machine Learning Approach for Real-Time Condition Monitoring of Rotating Machines ». In *2021 9th RSI International Conference on Robotics and Mechatronics (ICRoM)*. (novembre 2021), p. 267-273. <<https://doi.org/10.1109/ICRoM54204.2021.9663495>>. Consulté le 12 août 2024.
- Nandi, Asoke et Hosameldin Ahmed. 2019. *Condition Monitoring with Vibration Signals: Compressive Sampling and Learning Algorithms for Rotating Machine*, 1^{re} éd. Wiley. <<https://doi.org/10.1002/9781119544678>>. Consulté le 7 juillet 2025.
- Nielsen, Aileen. 2019. *Practical time series analysis: prediction with statistics and machine learning*. Sebastopol, CA : O'Reilly Media, Inc, 480 p.
- Perez, Robert X., éd. 2023. *Condition Monitoring, Troubleshooting and Reliability in Rotating Machinery*, 1^{re} éd. Wiley. <<https://doi.org/10.1002/9781119631620>>. Consulté le 10 septembre 2024.

- Saha, Swapnil Sayan, Sandeep Singh Sandha et Mani Srivastava. 2022. « Machine Learning for Microcontroller-Class Hardware: A Review ». *IEEE Sensors Journal*, vol. 22, n° 22, p. 21362-21390. <<https://doi.org/10.1109/JSEN.2022.3210773>>.
- Sai Charan, Kovuru. 2022. « An Auto-Encoder Based TinyML Approach for Real-Time Anomaly Detection ». *SAE International Journal of Advances and Current Practices in Mobility*, vol. 5, n° 4, p. 1496-1501. <<https://doi.org/10.4271/2022-28-0406>>.
- Shreve, Dennis H. 1995. « SIGNAL PROCESSING FOR EFFECTIVE VIBRATION ANALYSIS ». (1 novembre 1995), p. 11.
- Ward, Tyler, Kouroush Jenab, Jorge Ortega-Moody et Selva Staub. 2024. « Comprehensive Review of Machine Learning Techniques for Condition-Based Maintenance ». *International Journal of Prognostics and Health Management*, vol. 15, n° 2. <<https://doi.org/10.36001/ijphm.2024.v15i2.3850>>. Consulté le 5 novembre 2024.