

Projet de fin d'étude en vue de l'obtention du diplôme de  
**D'INGENIEUR D'Etat**

Filière  
**Génie Industriel**

Spécialité  
**Génie Industriel**

Thème  
**Sélection d'un Algorithme de Deep Learning  
Approprié pour la Maintenance Prédictive.**

Réalisé par  
**LARIBI Hadjer  
MELIANI Ouezna**

**Les members de jury:**

<b>GHOMARI Leila</b>	<b>(MCA)</b>	Présidente
<b>BELAYADI Djahida</b>	<b>(MCB)</b>	Encadrante
<b>AMRANI Mohamed</b>	<b>(MAA)</b>	Examineur

**Alger, 25/06/2023**

**Année universitaire 2022–2023**

# Dédicaces

“

*Avec tous mes sentiments de respect, avec l'expérience de ma reconnaissance. je dédie ce travail et ma joie :*

*À celle qui est mon havre de paix , à la prunelle de mes yeux , à la source de ma joie et de mon bonheur , à ma lune qui allume mon chemin d'espoir, à celle qui est ma moitié,à ” **ma chère maman** ” .*

*À celui qui a fait de moi une femme , ma source de vie,d'amour et d'affectation, à mon support qui était toujours à mes cotés ,me soutenant et m'encourageant,à ” **mon prince papa** ” .*

*À mes sœurs, **Sarah,Sonia etChaima** , qui ont été d'une précieuse aide en me prodiguant conseils, encouragement et soutien tout au long de mes études. Leur présence et leur soutien constant ont été une source de motivation et de force pour moi.*

*À mes chers petits neveux **Iyad et Saed**, ainsi qu'à ma petite nièce **Bibicha** .*

*À tous les membres de ma grande famille ;mes tantes, mes oncles, mes cousins et cousines.*

*À mon ami **Meziane**, qui a été constamment présent pour moi en termes de soutien moral et d'assistance précieuse. Sa présence et son aide ont été inestimables tout au long de cette période.*

*À tous mes amis ,**Imen, Ilicia, Djamila, Lyna, Ghaya, Ryma, Maya, Amira, Zaki,Aymen, Rida.***

*Sans oublier mon binôme, **Ouezna**, pour son soutien moral constant, sa patience et sa compréhension tout au long de ce projet. Son précieux apport a été essentiel pour mener à bien cette collaboration.*

*À tous ceux qui ont participé à ma réussite et à tout qui m'aiment.*

”

**Hadjer**

“

*Du fond de mon cœur, je dédie ce travail à tous ceux qui sont chers.*

*À ma chère **Famille**, les mots ne sauraient pas exprimer mon respect, mon amour et ma considération pour les sacrifices que vous avez consenti pour mon instruction et mon bien-être. Je vous remercie du fond du cœur pour le soutien, la bienveillance, la générosité et l'amour que vous me portez, que ce modeste travail soit l'exhaussement de vos prières et le fruit de vos sacrifices, bien que je ne vous en acquitterai jamais assez. Je compte toujours vous rendre encore plus fiers.*

*À mon bras droit mon cher frère **Omar** et ma moitié ma chère sœur **Souad**, qui ont toujours cru en moi et encouragé à réaliser mes passions, je vous suis infiniment reconnaissante pour tout l'amour, le soutien et la confiance que vous m'accordez.*

*À mes partenaires dans la vie **Ilicia, Melina, Lydia, Selma, Nadia, Yamina, Djamila, Ghaya, Lyna.B** et **Lyna.O**, pour tous ces années d'amitié inoubliables, les bons moments et les rires qu'on a partagés.*

*À mon binôme **Hadjer**, je tiens à te remercier pour ta présence inébranlable, ta loyauté et ton amitié indéfectible. Sans toi, ce projet n'aurait jamais été le même, et je suis honoré de pouvoir te dédier ce travail, symbole de notre partenariat exceptionnel.*

*À la mémoire de mon très cher **Père**, et ma chère **Grand-mère** que dieu bénisse leur âme qui était et restera à jamais dans mon cœur.*

”

*Ouezna*

# Remerciements

Nous tenons tout d'abord à remercier en premier lieu "الله" , le tout puissant, de nous avoir dotés d'un immense courage, d'une patience infinie et d'une volonté inébranlable afin de mener à bien ce projet.

Nous tenons à exprimer nos sincères remerciements aux membres du jury, madame **GHO-MARIE Leila** et monsieur **AMRANIE Mohamed** pour l'honneur qu'ils nous font en consacrant leur temps à examiner et évaluer ce travail.

Nos sincères gratitudes à notre chère encadrante **Dr.BELAYADI Djahida** pour la qualité de son encadrement, ses valeureux conseils, et son suivi incontestable tout au long de ce projet. Son orientation a constitué un apport considérable sans lequel ce travail n'aurait pas pu être mené au bon port. La réalisation de ce projet fut une expérience extrêmement enrichissante grâce à elle.

Nous tenons à remercier l'ensemble du personnel de la raffinerie d'Alger pour leur chaleureux accueil, leurs conseils, et l'intérêt qu'ils ont attribué à notre travail. Nos remerciements s'étendent particulièrement à notre promoteur **Mr.MCHANTEL Mourad**.

Des remerciements infinis vont à tous les enseignants, qui ont participé, contribué et prodigué toutes les connaissances nécessaires à notre formation durant toute la durée que nous avons passé à l'Ecole Nationale Supérieure de Technologie, en particulier au sein du département de Génie Industriel et Maintenance. Nous tenons également à exprimer notre gratitude pour leur participation et leur soutien tout au long de la réalisation de ce travail, qui représente le fruit de plusieurs années d'études.

Pour finir, nous souhaitons remercier toute personne ayant contribué de près ou de loin à la réalisation de ce travail.

# Table des matières

Dédicaces . . . . .	I
Remerciements . . . . .	III
Introduction générale . . . . .	1
<b>Première partie</b> <b>Partie Théorique</b> . . . . .	<b>3</b>
<b>1 Etat de l’art : Généralités sur la Maintenance Prédictive</b> . . . . .	<b>4</b>
1.1 Introduction . . . . .	5
1.2 Maintenance Industrielle . . . . .	5
1.2.1 Définition de la Maintenance Industrielle . . . . .	5
1.2.2 Objectifs de la Maintenance Industrielle . . . . .	5
1.2.3 Types de la Maintenance Industrielle . . . . .	6
1.3 Maintenance Prédictive . . . . .	7
1.3.1 Avantages de la Maintenance Prédictive . . . . .	8
1.3.2 Limitations de la Maintenance Prédictive . . . . .	8
1.3.3 Techniques de Surveillance de la Maintenance Prédictive . . . . .	9
1.3.4 Les Approches de la Maintenance Prédictive . . . . .	11
1.3.5 Les techniques de la Maintenance Prédictive . . . . .	12
1.3.6 Processus d’Implémentation de la Maintenance Prédictive Basée sur les Données . . . . .	13
1.4 Table Comparative des Types de Maintenance Industrielle . . . . .	14
1.5 Conclusion . . . . .	14
<b>2 Etat de l’art : Généralités sur les Approches de Deep Learning</b> . . . . .	<b>15</b>
2.1 Introduction . . . . .	16
2.2 Intelligence Artificielle . . . . .	16
2.3 Machine Learning . . . . .	18
2.3.1 Définition du Machine Learning : . . . . .	18
2.3.2 Approches du Machine Learning : . . . . .	18
2.4 Deep Learning . . . . .	19
2.4.1 Définition du Deep Learning : . . . . .	19
2.4.2 Notions de Deep Learning . . . . .	20
2.4.3 Machine learning vs Deep Learning . . . . .	22
2.4.4 Approches de Deep Learning : . . . . .	22
2.5 Deep Learning pour la Maintenance Prédictive . . . . .	26
2.5.1 Revue de la Littérature . . . . .	27

2.5.2	Étude Comparative des Algorithmes de Deep Learning pour la Maintenance Prédictive . . . . .	31
2.6	Conclusion . . . . .	35

**Deuxième partie Partie Expérimentale . . . . . 36**

**3 Implémentation et Evaluation des Meilleurs Algorithmes DL pour la PdM 37**

3.1	Introduction . . . . .	38
3.2	Méthodologie . . . . .	38
3.3	Description des Données d'Implémentation . . . . .	38
3.3.1	Base de Données 1 . . . . .	39
3.3.2	Base de Données 2 . . . . .	39
3.4	Prétraitement des Données . . . . .	40
3.4.1	Nettoyage des Données . . . . .	40
3.4.2	Sélection des Caractéristiques . . . . .	40
3.4.3	Normalisation des Données . . . . .	42
3.5	Division des Données . . . . .	42
3.6	Conception et Implementation . . . . .	43
3.6.1	Construction des Modèles : . . . . .	43
3.6.2	Les Hyperparamètres : . . . . .	44
3.6.3	Les Métriques d'Évaluation : . . . . .	45
3.7	Résultats et Discussion . . . . .	47
3.8	Comparaison avec des Études Similaires . . . . .	54
3.9	Conclusion . . . . .	54

**Conclusion et perspectives . . . . . 55**

**Annexes . . . . . 62**

**A Présentation de la Raffinerie d'Alger . . . . . 63**

**B Le compresseur 520-k-052A/B . . . . . 65**

**C Les codes de simulation . . . . . 66**

# Table des figures

1.1	Les types de maintenance [8] . . . . .	6
1.2	Pourcentage d'utilisation de différentes techniques [15] . . . . .	9
1.3	Schéma d'implémentation de la Maintenance Prédicative . . . . .	13
2.1	Les événements marquants dans l'histoire de l'IA . . . . .	17
2.2	Les domaines de l'Intelligence Artificielle . . . . .	17
2.3	Approches du Machine Learning . . . . .	18
2.4	Structure du neurone biologique [40] . . . . .	20
2.5	Neurone Artificiel [40] . . . . .	21
2.6	Réseaux de neurones feedforward [40] . . . . .	23
2.7	Réseaux de neurones convolutionnels [40] . . . . .	23
2.8	Réseaux de neurones récurrents [50] . . . . .	24
2.9	Long short-term memory [52] . . . . .	24
2.10	AutoEncodeurs [40] . . . . .	25
2.11	Réseaux Adversariaux Génératifs [40] . . . . .	25
2.12	Machines de Boltzmann Restreintes [40] . . . . .	26
2.13	Réseaux de croyances profondes [40] . . . . .	26
2.14	Evolution du nombre de publications sur le Deep Learning pour la Maintenance Prédicative dans le moteur de recherche Google Scholar [48] . . . . .	27
2.15	Critères d'évaluation des algorithmes DL pour la PdM à partir de l'état de l'art . . . . .	33
3.1	La structure générale du processus de l'étude . . . . .	38
3.2	Heatmap de corrélation de BDD1 . . . . .	41
3.3	Code de normalisation des données . . . . .	42
3.4	Code de division des données . . . . .	43
3.5	Code de LSTM . . . . .	43
3.6	Code de RNN . . . . .	44
3.7	Code de CNN . . . . .	44
3.8	Matrice de confusion . . . . .	46
3.9	Matrice de confusion du LSTM-BDD1 . . . . .	48
3.10	Matrice de confusion du RNN-BDD1 . . . . .	48
3.11	Matrice de confusion du CNN-BDD1 . . . . .	48
3.12	Courbes ROC des différents modèles-BDD1 . . . . .	49
3.13	Courbes Précision-Rappel des différents modèles-BDD1 . . . . .	49
3.14	Courbes LOSS des différents modèles-BDD1 . . . . .	49
3.15	Les métriques des différents modèles-BDD1 . . . . .	49
3.16	Matrice de confusion du LSTM-BDD2-1 . . . . .	50
3.17	Matrice de confusion du RNN-BDD2-1 . . . . .	50
3.18	Matrice de confusion du CNN-BDD2-1 . . . . .	50

3.19	Courbes ROC des différents modèles-BDD2 . . . . .	51
3.20	Courbes Précision-Rappel des différents modèles-BDD2 . . . . .	51
3.21	Courbes LOSS des différents modèles-BDD2 . . . . .	51
3.22	Les métriques des différents modèles-BDD2 . . . . .	51
3.23	Matrice de confusion du LSTM-BDD2-2 . . . . .	52
3.24	Matrice de confusion du RNN-BDD2-2 . . . . .	52
3.25	Matrice de confusion du CNN-BDD2-2 . . . . .	52
A.1	Organigramme de la raffinerie d'Alger . . . . .	64
B.1	Compresseur 520-k-052A/B . . . . .	65



# Liste des tableaux

1.1	Avantages de la Maintenance Prédicative par rapport à la maintenance corrective et à la maintenance préventive systématique . . . . .	8
1.2	Tableau récapitulatif des techniques de la Maintenance Prédicative [15] . . . . .	11
1.3	Tableau récapitulatif des types de Maintenance Industrielle . . . . .	14
2.1	L’analogie entre les neurones biologiques et les neurones artificiels . . . . .	21
2.2	Tableau comparatif entre Machine learning et Deep learning . . . . .	22
2.3	Tableau récapitulatif des articles sélectionnés dans la littérature sur le Deep Learning pour Maintenance Prédicative . . . . .	30
2.4	Comparaison entre les Algorithmes DL pour la PdM selon les Critères . . . . .	34
2.5	Scores des algorithmes par ordre décroissant . . . . .	34
3.1	Les variables mesurés du BDD1 . . . . .	39
3.2	Les variables du BDD2 . . . . .	40
3.3	Les caractéristiques de simulation des bases de données . . . . .	42
3.4	Les hyperparamètres optimaux des algorithmes pour chaque BDD . . . . .	45
3.5	Les métriques des différentes modèles BDD1 . . . . .	48
3.6	Les métriques des différents modèles BDD2-cas1 . . . . .	50
3.7	Les métriques des différents modèles BDD2-cas2 . . . . .	53
3.8	Comparaison avec les résultats d’études similaires . . . . .	54

# Liste des Acronymes

<b>AE</b>	<i>AutoEncoders</i>
<b>AFNOR</b>	<i>Association Française de NORmalisation</i>
<b>ANN</b>	<i>Artificial Neural Network</i>
<b>AUC</b>	<i>Area Under the Curve</i>
<b>AUPR</b>	<i>Area Under the Precision-Recall Curve</i>
<b>BDD1</b>	<i>Base de données 1</i>
<b>BDD2</b>	<i>Base de données 2</i>
<b>CNN</b>	<i>Convolutional Neural Network</i>
<b>CM</b>	<i>Confusion Matrix</i>
<b>DBN</b>	<i>Deep Belief Networks</i>
<b>DL</b>	<i>Deep Learning</i>
<b>FN</b>	<i>Faux Négatifs</i>
<b>FNN</b>	<i>Feedforward Neural Network</i>
<b>FP</b>	<i>Faux Positifs</i>
<b>GAN</b>	<i>Generative Adversarial Networ</i>
<b>GMAO</b>	<i>Gestion de la Maintenance Assistée par Ordinateur</i>
<b>HVAC</b>	<i>Heating, Ventilation, and Air Conditioning</i>
<b>IA</b>	<i>Intelligence Artificielle</i>
<b>IHM</b>	<i>Interface Homme Machine</i>
<b>IoT</b>	<i>Intenet Of Things</i>
<b>LSTM</b>	<i>Long Short-Term Memory</i>
<b>ML</b>	<i>Machine Learning</i>
<b>PdM</b>	<i>Prédictive Maintenance</i>

<b>RBM</b>	<i>Restricted Boltzmann Machines</i>
<b>RNN</b>	<i>Recurrent Neural Network</i>
<b>ROC</b>	<i>Receiver Operating Characteristics</i>
<b>RUL</b>	<i>Remaining Useful Life</i>
<b>TFN</b>	<i>Taux de Faux Négatifs</i>
<b>TVP</b>	<i>Taux de Vrais Positifs</i>
<b>VN</b>	<i>Vrais Négatifs</i>
<b>VP</b>	<i>Vrais Positifs</i>

# Introduction générale

De nos jours, la maintenance est un facteur stratégique pour assurer une productivité élevée des systèmes industriels. Cependant, en raison des contraintes économiques, les entreprises sont obligées de réduire les coûts de maintenance, ce qui entraîne de graves implications pour la fiabilité à long terme. Par conséquent, il est devenu crucial pour les entreprises de concevoir des stratégies de maintenance adaptées qui garantissent un fonctionnement efficace des usines de production en termes de qualité et de disponibilité. En conséquence, la définition et les pratiques de la maintenance ont subi d'importantes transformations au fil du temps, en grande partie grâce aux apports substantiels des efforts de recherche.

Actuellement, il y a une transition vers la quatrième révolution connue sous le nom d'Industrie 4.0, qui se caractérise par l'intégration des systèmes cyber-physiques et l'Internet industriel des objets. Cela implique l'utilisation de logiciels, de capteurs et d'unités de contrôle intelligentes pour améliorer les processus industriels et répondre à leurs exigences. Ces avancées technologiques permettent la mise en place de techniques avancées de Maintenance Prédictive automatisée, qui analysent de grandes quantités de données de processus grâce à la surveillance de l'état. La Maintenance Prédictive (PdM) initialement conçue à la fin des années 1940 [Prajapati], est devenue une approche prometteuse, apportant des solutions pour la durée de vie restante des équipements grâce à la prédiction des données collectées par différents capteurs sur les équipements. Il a atteint une importance critique pour les industries en raison de la complexité croissante des interactions entre les diverses activités de production au sein de vastes écosystèmes de fabrication.

Au fil du temps, l'industrie s'est de plus en plus focalisée sur l'utilisation de l'Intelligence Artificielle et des techniques d'apprentissage automatique, en raison de leur capacité à concevoir des modèles automatiques qui traitent la quantité massive de données actuellement collectées, qui croît de façon exponentielle. La tendance de la recherche sur l'apprentissage automatique s'est orientée vers des modèles plus complexes, notamment l'apprentissage en profondeur, en raison de leur précision accrue dans le traitement de vastes ensembles de données. De nombreuses méthodes de Deep Learning ont été proposées pour la Maintenance Prédictive, telles que les AutoEncodeurs, les réseaux neuronaux récurrents, les réseaux neuronaux convolutifs et les réseaux de mémoire à court terme. Ces méthodes visent à prédire les défaillances des équipements en apprenant automatiquement à partir des données historiques et en estimant la probabilité de défaillances futures. Ces approches ont été largement exploitées dans la littérature, notamment dans les travaux de Guo et al. (2017) [1], Wu et al. (2020) [2], Yun et al. (2021) [3] et Wahid et al. (2022) [4].

Dans ce contexte, il existe effectivement de nombreux algorithmes de Deep Learning disponibles, chacun ayant ses propres caractéristiques, et plusieurs facteurs peuvent influencer leur

performance. Par conséquent, choisir le meilleur algorithme DL pour la PdM n'est pas évident.

## Problématique

Comment choisir l'algorithme de Deep Learning le plus approprié pour assurer les tâches de la Maintenance Prédictive ?

À cet égard les principaux objectifs de ce projet sont :

- Sélectionner les algorithmes de Deep Learning utilisés dans le cadre de la PdM, en effectuant un état de l'art pour le domaine d'application.
- Choisir les trois meilleurs algorithmes de Deep Learning en réalisant une étude comparative basée sur les critères les plus couramment utilisés dans les articles de recherche.
- Formuler des recommandations sur le choix des algorithmes de Deep Learning les plus appropriés pour le domaine de la Maintenance Prédictive et faire une comparaison avec d'autres travaux similaires, en interprétant et en discutant les résultats obtenus lors des expérimentations.

Dans le but d'assurer la solidité et la crédibilité de nos résultats, nous avons effectué un stage de deux mois au sein de la raffinerie d'Alger, une filiale de Sonatrach. Cette raffinerie est spécialisée dans le traitement du pétrole brut pour produire une variété de produits pétroliers raffinés. Pendant ce stage, notre objectif principal était de collecter les données nécessaires à partir des capteurs d'un compresseur 520-K-052A/B, afin de réaliser la simulation des algorithmes sélectionnés après l'étude comparative.

La structure de notre travail se décompose en trois chapitres :

Le **premier chapitre** aborde le concept de la maintenance, en présentant ses différents types, et se focalise principalement sur la Maintenance Prédictive, en explorant ses techniques, approches et outils.

Le **deuxième chapitre** offre un bref aperçu de l'Intelligence Artificielle et du Machine Learning, puis explore en détail le sujet du Deep Learning en définissant ses différents algorithmes et en soulignant sa relation avec la Maintenance Prédictive. De plus, il présente une synthèse des travaux récents qui combinent ces deux concepts dans la littérature. Enfin, une comparaison des algorithmes du Deep Learning est réalisée en utilisant des critères couramment employés dans les articles de recherche afin d'identifier les algorithmes les plus efficaces pour résoudre les problèmes liés à la Maintenance Prédictive.

Le **troisième chapitre** est consacré en premier lieu à la présentation des différentes bases de données sélectionnées pour le test, ainsi que la construction des algorithmes choisis dans le deuxième chapitre et la définition des métriques d'évaluation. Par la suite, nous évaluons la capacité prédictive de nos modèles sur l'ensemble des bases de données, et nous effectuons une synthèse en combinant les divers résultats.

Première partie  
Partie Théorique

# Chapitre 1

## Etat de l'art : Généralités sur la Maintenance Prédictive

### 1.1 Introduction

La maintenance joue un rôle crucial dans le secteur industriel, tant sur le plan technologique que sur le plan économique. Son objectif principal est d'améliorer la disponibilité optimale des installations industrielles et de prolonger leur durée de vie. Elle a un rôle préventif essentiel en évitant les incidents et les accidents, assurant ainsi la sécurité des personnes et des biens [5].

Dans ce chapitre, nous aborderons tout d'abord la définition de la maintenance industrielle ainsi que ses divers types, parmi lesquels figure la maintenance prédictive, pour laquelle nous accordons une attention particulière, en soulignant ses avantages, ses différentes approches et les multiples techniques et outils utilisés, et en fournissant également un aperçu de son processus d'implémentation.

### 1.2 Maintenance Industrielle

La Maintenance Industrielle fait référence au processus consistant à s'assurer que les équipements et les machines industriels sont entretenus, réparés et mis à niveau pour garantir des performances, une fiabilité et une sécurité optimales.

#### 1.2.1 Définition de la Maintenance Industrielle

Elle est définie par l'Association Française de Normalisation (AFNOR) en 1994 comme « L'ensemble de toutes les actions techniques, administratives et de management durant le cycle de vie d'un bien, destinées à le maintenir ou à le rétablir dans un état dans lequel il peut accomplir la fonction requise » [6].

#### 1.2.2 Objectifs de la Maintenance Industrielle

Contrairement à la perception courante, la fonction de maintenance ne se limite pas à la résolution rapide des pannes. Son véritable objectif consiste plutôt à anticiper et à éviter toutes les pertes résultant de problèmes liés à des équipements ou à des systèmes défectueux. Dans une organisation de premier plan, le département de maintenance a pour responsabilité d'accomplir et de maintenir les objectifs suivants [5][7] :

- La disponibilité et la durée de vie optimale des équipements.
- Stock minimum de pièces de rechange.
- L'optimisation des coûts de maintenance.
- La protection des personnes et des biens.



### 1.2.3 Types de la Maintenance Industrielle

Il existe diverses formes de maintenance qui peuvent être mises en œuvre dans l'industrie. Un schéma récapitulant ces différents types est présenté dans la figure (1.1).

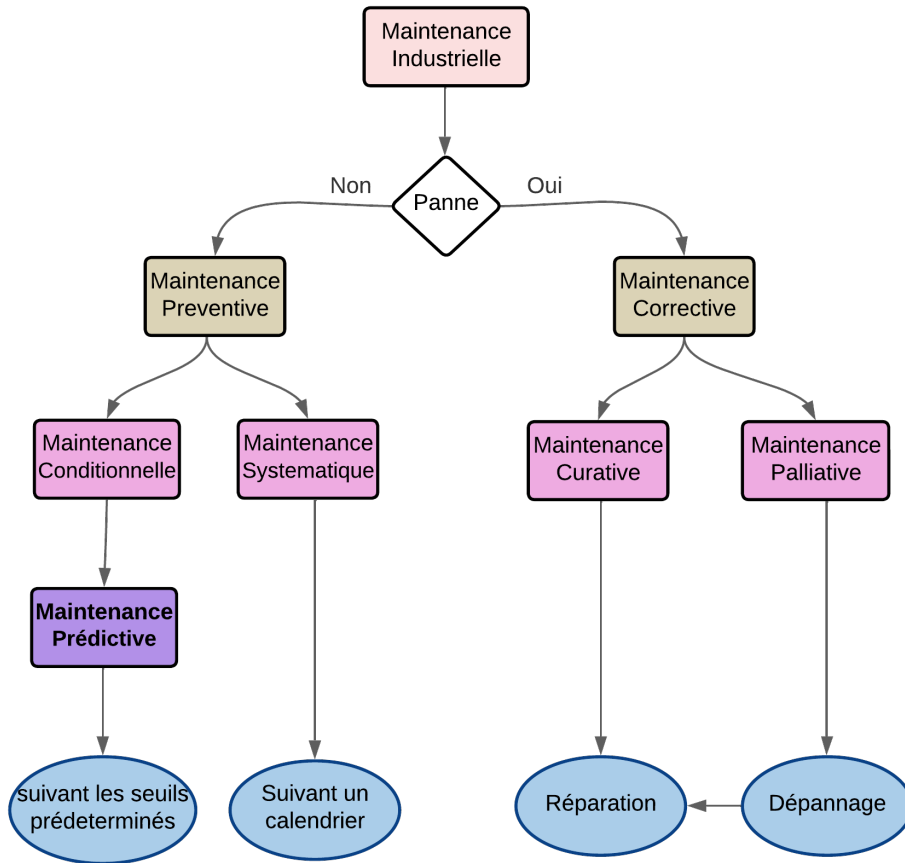


FIG. 1.1 : Les types de maintenance [8]

#### 1. Maintenance Corrective

La maintenance corrective est définie selon l'AFNOR (FD X 60-000) comme suit : « C'est la maintenance exécutée après détection d'une panne et destinée à remettre un bien dans un état dans lequel il peut accomplir une fonction requise » [6].

Elle se présente sous deux formes :

- **Maintenance Curative :**

Selon l'AFNOR (FD X60-000) « C'est une action de maintenance corrective ayant pour objet de rétablir un bien dans un état spécifié pour lui permettre d'accomplir une fonction requise ». Les actions réalisées doivent aboutir à un résultat permanent. Des modifications et améliorations peuvent être apportées, afin de réduire l'occurrence d'apparition de la défaillance ou d'en limiter l'incidence [6].

- **Maintenance Palliative :**

Selon l'AFNOR (FD X60-000) « C'est une action de maintenance corrective destinée à permettre à un bien d'accomplir provisoirement tout ou partie d'une fonction requise appelée couramment (dépannage) ». Elle se compose principalement d'actions temporaires qui doivent être suivies par des actions curatives [6].

### 2. Maintenance Préventive

La maintenance préventive est définie selon l'AFNOR (FD X 60-000) comme suit : « C'est la maintenance exécutée à des intervalles prédéterminés ou selon des critères prescrits et destinée à réduire la probabilité de défaillance ou la dégradation du fonctionnement d'un bien » [6].

Elle est subdivisée en deux types distincts :

- **Maintenance Systématique :**

Selon l'AFNOR (FD X60-000) « C'est une maintenance préventive exécutée à des intervalles de temps préétablis ou selon un nombre défini d'unités d'usage mais sans contrôle préalable de l'état du bien » [6]. Elle est déclenchée selon un calendrier (heures de travail, kilomètres effectués, ect), et se traduit par des interventions planifiées visant à nettoyer, réparer ou remplacer périodiquement un organe sans effectuer de contrôle préalable de l'équipement [9].

- **Maintenance Conditionnelle :**

Selon l'AFNOR (FD X60-000) « C'est une maintenance préventive basée sur une surveillance du fonctionnement du bien et/ou des paramètres significatifs de ce fonctionnement et intégrant les actions qui en découlent ». La surveillance du fonctionnement et des paramètres peut être exécutée selon un calendrier, ou à la demande, ou de façon continue [6]. Le principe de la maintenance préventive conditionnelle repose sur l'évaluation de l'état du système à partir de la surveillance d'indicateurs significatifs. Lorsque cet état atteint un seuil défini, indiquant un risque important, une action de maintenance est déclenchée. L'erreur n'est plus liée à la probabilité d'apparition du risque, comme c'est le cas en maintenance préventive systématique. Elle est plutôt liée à la définition du seuil et à la décision de déclencher l'action de maintenance [10].

La Maintenance Prédictive qui est une évolution de la maintenance conditionnelle (figure 1.1), et elle sera détaillé dans la section suivante.

## 1.3 Maintenance Prédictive

La maintenance prédictive (PdM) est un domaine ancien, mais son histoire n'est pas correctement documentée. Cependant, il est largement supposé qu'elle est opérationnelle dans le monde industriel depuis les années 1990 [11]. Depuis lors, elle est devenue le type le plus récent de maintenance qui a attiré l'attention des chercheurs et des industriels. Elle est définie selon la

norme NF EN 13306 X 60-319 : start-up et entreprises spécialisées comme : « Une maintenance conditionnelle exécutée en suivant les prévisions extrapolées de l'analyse et de l'évaluation de paramètres significatifs de la dégradation du bien » [6], cela signifie que l'on peut anticiper et détecter la panne avant qu'elle ne se produise, grâce une surveillance continue.

### 1.3.1 Avantages de la Maintenance Prédictive

La Maintenance Prédictive présente de nombreux avantages par rapport aux autres types de maintenance, dont certains sont énumérés dans le tableau (1.1) :

TAB. 1.1 : Avantages de la Maintenance Prédictive par rapport à la maintenance corrective et à la maintenance préventive systématique

---

#### Avantages de la Maintenance Prédictive par rapport à la maintenance systématique préventive [ouadah2021selections] [9] [12]

---

- Augmenter la durée de vie des équipements.
- Diminuer l'utilisation des pièces de rechange.
- Meilleure planification des interventions.
- Utilisation efficace des ressources.

---

#### Avantages de la Maintenance Prédictive par rapport à la maintenance corrective [12] [13]

---

- Optimiser les coûts d'entretien.
  - Réduire les arrêts et les pannes non-programmés.
  - Sécurité renforcée.
  - Meilleures relations entre les services Production et Maintenance.
- 

### 1.3.2 Limitations de la Maintenance Prédictive

Bien que la Maintenance Prédictive présente de nombreux avantages, elle est également sujette à certaines limitations qui peuvent entraver son adoption. Voici quelques-unes des limites fréquemment rencontrées [9] [14] :

- Coûts de mise en œuvre : la mise en place d'un système de Maintenance Prédictive peut nécessiter des investissements importants en termes d'infrastructure, de capteurs, de logiciels d'analyse de données et de formation du personnel.
- Données insuffisantes ou mal interprétées : la Maintenance Prédictive repose sur la collecte et l'analyse de données en temps réel provenant des équipements. Si les données collectées sont insuffisantes, de mauvaises qualités ou inappropriées pour la modélisation prédictive, la précision des prédictions peut être compromise.
- Facteurs externes imprévisibles : La Maintenance Prédictive est basée sur la prévision des pannes d'équipement en surveillant les paramètres prévisibles. Cependant, il existe des facteurs externes imprévisibles tels que des événements naturels, des erreurs humaines ou

des conditions de fonctionnement changeantes qui peuvent affecter les performances de l'équipement de manière inattendue.

- La mise en œuvre et l'exploitation d'un système de Maintenance Prédictive peuvent être complexes et nécessiter des compétences spécialisées.

### 1.3.3 Techniques de Surveillance de la Maintenance Prédictive

Pour maintenir un suivi de l'état d'un équipement, il est essentiel de recueillir régulièrement des informations concernant son niveau de dégradation ou de performance. Diverses techniques d'analyse sont employées dans ce but (voir la figure (1.2)) [15] [16].

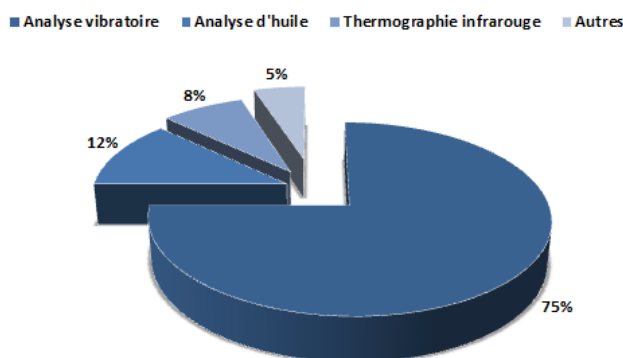


FIG. 1.2 : Pourcentage d'utilisation de différentes techniques [15]

#### 1. Analyse Acoustique :

L'analyse acoustique permet de détecter tout bruit anormal à l'aide de microphones placés à distance de l'équipement.

#### 2. Le Contrôle par Ultrasons :

Le contrôle par ultrasons permet de détecter des défauts de faibles amplitudes à hautes fréquences tels que l'initiation de la dégradation d'un roulement.

#### 3. La Thermographie Infrarouge :

Tout dysfonctionnement d'un équipement se traduit systématiquement par une augmentation de température. Même si cette augmentation n'est pas visible à l'œil nu, elle se révèle instantanément dans le spectre infrarouge. C'est pourquoi la thermographie infrarouge représente une technologie de Maintenance Prédictive efficace permettant de localiser rapidement et en toute sécurité les problèmes. Grâce à la thermographie infrarouge, il est possible de détecter les défauts avant qu'ils ne causent une panne des équipements, évitant ainsi les conséquences néfastes telles que les perturbations ou les arrêts de production, voire même le déclenchement d'un incendie dans les cas les plus graves.

#### 4. Analyse Vibratoire :

L'analyse vibratoire est considérée comme la méthode la plus familière et répandue, en raison de sa capacité à s'ajuster aux éléments mécaniques et aux machines industrielles

en fonctionnement. Elle offre la capacité de repérer la plupart des imperfections pouvant survenir dans les machines tournantes.

### 5. Analyse d'Huile :

L'analyse d'huile s'applique à tous les dispositifs qui renferment des fluides de lubrification, tels que des réducteurs et des moteurs thermiques. La méthode implique le prélèvement d'un échantillon d'huile, l'analyser afin d'évaluer l'état de l'équipement.

Un tableau récapitulatif des différentes techniques, qui résume les avantages, les inconvénients, ainsi que les domaines d'application privilégiés de chaque technique, est présenté dans le tableau ci-dessous (1.3.3).

TAB. 1.2 : Tableau récapitulatif des techniques de la Maintenance Prédictive [15]

Technique	Principaux avantages	Principales limitations	Champs d'applications
Analyse acoustique	<ul style="list-style-type: none"> <li>• Facilite la détection précoce des défauts sonores.</li> <li>• Permet une surveillance continue.</li> </ul>	<ul style="list-style-type: none"> <li>• Sensibilité au bruit ambiant.</li> <li>• La difficulté d'établir un diagnostic.</li> </ul>	La détection d'un son anormal qui peut ensuite être analysé à l'aide d'une analyse vibratoire.
Analyse vibratoire	<ul style="list-style-type: none"> <li>• Détection de défauts à un stade précoce.</li> <li>• Opportunités d'effectuer un diagnostic approfondi.</li> <li>• La possibilité de surveiller à distance.</li> </ul>	En ce qui concerne la surveillance continue, les installations impliquées peuvent être assez coûteuses.	Détection des défauts de tous les organes mobiles de la machine (déséquilibres, problèmes d'alignement) et de sa structure.
La thermographie Infrarouge	<ul style="list-style-type: none"> <li>• Facilite la réalisation d'un contrôle rapide de l'installation.</li> <li>• Une interprétation qui est généralement immédiate.</li> </ul>	<ul style="list-style-type: none"> <li>• Ne permet pas d'effectuer un diagnostic approfondi.</li> <li>• Le contrôle est restreint à ce que la caméra peut percevoir visuellement.</li> </ul>	La détection de tous les défauts engendrant un échauffement (manque de lubrification en particulier)
Analyse d'huile	<ul style="list-style-type: none"> <li>• Identifier une présence anormale de pollution dans le lubrifiant afin de prévenir toute détérioration ou augmentation de température résultant de celle-ci.</li> <li>• Possibilité d'identifier l'origine de l'anomalie en analysant les particules.</li> </ul>	<ul style="list-style-type: none"> <li>• Ne permet pas de fournir une localisation précise du défaut.</li> <li>• Il est essentiel de prendre de nombreuses précautions lors du prélèvement de l'échantillon.</li> </ul>	Contrôle des propriétés physico-chimiques du lubrifiant, détection d'une insuffisance de lubrification et l'analyse des éléments d'usure

### 1.3.4 Les Approches de la Maintenance Prédictive

Pour appliquer la Maintenance Prédictive, plusieurs approches sont disponibles, à savoir :

#### 1. Modèles basés sur la Connaissance

Ce type de système repose sur une base de connaissances contenant des informations accumulées auprès d'experts du domaine, ainsi qu'une base de règles permettant d'appliquer ces connaissances à des problèmes spécifiques [17]. Il s'appuie sur des expériences qui peuvent être représentées par des règles, des faits ou des cas collectés au fil des années d'exploitation et de maintenance du système technique [18].

Ce modèle peut être divisé en trois catégories ; algorithmes basés sur des règles ou basés sur des cas et basés sur des connaissances floues [19].

### 2. Modèles basés sur la physique

Ce type de modèles utilise une représentation mathématique explicite pour formaliser la compréhension physique d'une machine ou d'un équipement en dégradation [17] ; ils exigent des compétences élevées en mathématiques et en physique des phénomènes pour l'application [18].

Ces modèles pourraient être divisés en : le modèle mathématique, le modèle de Markov caché, le modèle de distribution de probabilité et les modèles de filtre [19].

### 3. Modèles basés sur les Données

Les modèles basés sur les données ont gagné beaucoup d'importance ces dernières années grâce à l'amélioration de la disponibilité de la puissance de calcul et à la production de grandes quantités de données provenant chaque jour des systèmes techniques [18]. Ces modèles exploitent des algorithmes et des techniques avancées d'apprentissage automatique pour extraire des informations utiles à partir des données collectées utilisant des capteurs installés à des endroits appropriés [20].

Habituellement, les approches basées sur les données sont classées en techniques d'apprentissage automatique (ML) et en techniques statistiques [17].

### 4. Modèles Hybrides

Une autre approche efficace et émergente, qui ancre les avantages des divers modèles de pronostic actuellement disponibles, est l'« approche hybride ». C'est une combinaison de différentes approches qui aboutit à un modèle hybride [21].

## 1.3.5 Les techniques de la Maintenance Prédictive

Pour mettre en œuvre la Maintenance Prédictive, différents outils et technologies peuvent être utilisés. Voici quelques-uns des outils couramment utilisés [22][23] :

- **Internet des Objets(IoT) :**

L'Internet des Objets (IoT) propose des solutions en collectant et transmettant les données des capteurs vers des plateformes Cloud. Ces solutions facilitent également l'interconnexion des moyens de production au sein d'un réseau, permettant même une communication entre eux.

- **Les Domaines de l'Intelligence Artificielle :**

L'utilisation de capteurs permet de surveiller les différentes composantes d'une machine. Grâce aux algorithmes d'apprentissage automatique et profond, il est possible d'assurer la Maintenance Prédictive des équipements. Les données fournies par les capteurs peuvent servir de base de données pour informatiser les algorithmes de ML et DL.

- **Logiciels de Gestion de la Maintenance Assistée par Ordinateur GMAO :**

Ces logiciels offrent une solution intégrée pour gérer toutes les opérations de maintenance, y compris la Maintenance Prédictive. Ils permettent la planification des inspections, le suivi des résultats d'analyse, la gestion des réparations et la génération de rapports.

### 1.3.6 Processus d'Implémentation de la Maintenance Prédictive Basée sur les Données

Le schéma d'implémentation de la Maintenance Prédictive suggéré tire parti d'un large éventail de technologies émergentes, telles que l'IoT, l'apprentissage automatique (ML) et l'apprentissage profond (DL), et implique une série d'actions séquentielles, à savoir l'installation de capteurs, l'analyse et le traitement des données, visualisation des données et la prise de décision, comme illustré dans la figure (1.3) :

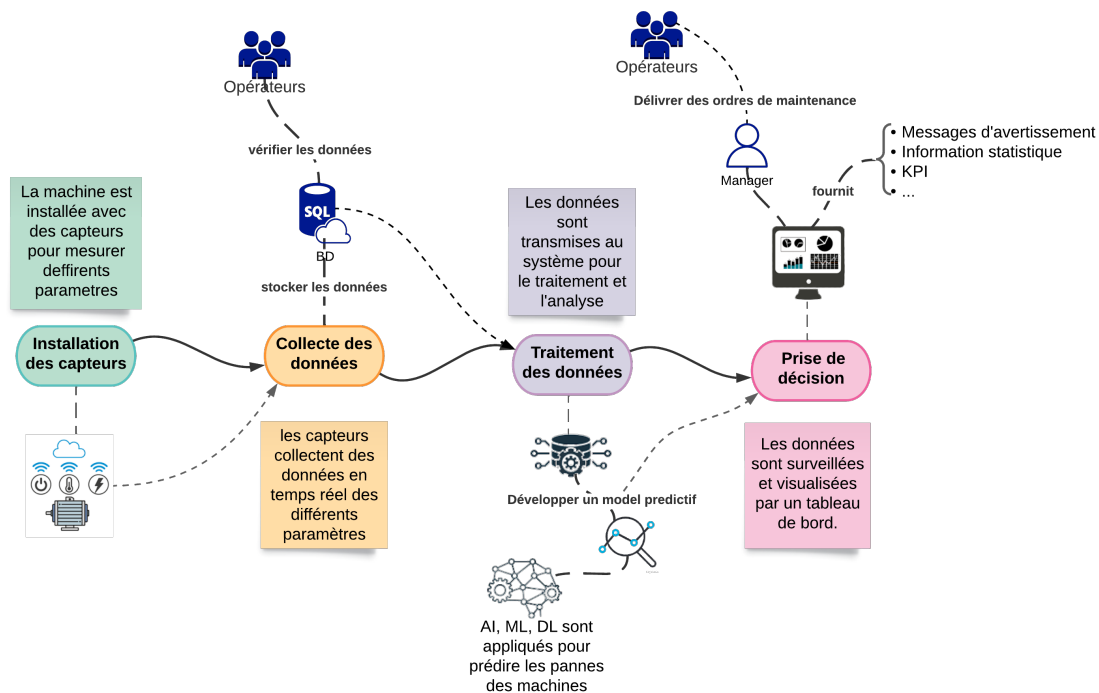


FIG. 1.3 : Schéma d'implémentation de la Maintenance Prédictive

Pour lancer la Maintenance Prédictive, des données en temps réel de divers paramètres, notamment la température, la pression et les vibrations, sont collectées via des capteurs installés sur l'équipement ; il est important de rassembler autant de données que possible pour former un ensemble de données solide. Ces données sont ensuite vérifiées, stockées dans des bases de données et soumises à un nettoyage, une transformation et une réduction des données. Le résultat de ce processus est utilisé dans la mise en œuvre d'une analyse exploratoire des données pour identifier les schémas, les tendances et les relations entre les variables. Cela peut aider à comprendre les facteurs qui influencent la maintenance et à déterminer les variables pertinentes pour la prédiction. Une fois les données sont prêtes, les algorithmes ML/DL sont utilisés pour créer un modèle prédictif. Ce dernier sera intégré avec un tableau de bord permettant de suivre



et de visualiser les résultats de l'étape précédente. Enfin, le manager assiste les techniciens lors des interventions de maintenance en fournissant des conseils et des ordres [24] [25] [26].

### 1.4 Table Comparative des Types de Maintenance Industrielle

TAB. 1.3 : Tableau récapitulatif des types de Maintenance Industrielle

Type de maintenance	Motif de l'utilisation	Etat de l'équipement	Objectif	Coût
Corrective	Défaillance	Arrêté	Remise en service de la machine	Coûteux en termes de perte de production et de sécurité
Préventive systématique	Révision périodique selon un intervalle de temps	Arrêté	Reprise des tâches de maintenance	Coûteuse en matériaux
Préventive conditionnelle	Révision programmée	Arrêté	Garantir le fonctionnement d'une machine pendant une période déterminée	Coûteux en programmant les arrêts nécessaires
Prédictive	Inspection programmée	En fonctionnement	Détecter les anomalies à temps et programmer leurs corrections	Coûteuse

### 1.5 Conclusion

La Maintenance Industrielle est essentielle pour assurer le bon fonctionnement des installations industrielles, et la Maintenance Prédictive constitue une approche proactive et innovante pour optimiser les performances et minimiser les coûts. En investissant dans la Maintenance Prédictive, les entreprises peuvent améliorer leur productivité, leur rentabilité et leur compétitivité sur le marché.

La Maintenance Prédictive est étroitement associée à l'Intelligence Artificielle plus précisément aux méthodes et aux algorithmes d'apprentissage profond qui permettent d'explorer des modèles cachés dans les données de surveillance des équipements.

Pour ceci, dans le chapitre suivant, nous aborderons l'apprentissage profond et sa relation avec la Maintenance Prédictive.

## Chapitre 2

### Etat de l'art : Généralités sur les Approches de Deep Learning

### 2.1 Introduction

Le Deep Learning (DL) de son nom apprentissage profond en français, est une forme d'Intelligence Artificielle (IA) et de Machine Learning (ML) plus précisément ; qui s'est révélée révolutionnaire dans la résolution de problèmes complexes et dans la réalisation de tâches qui étaient auparavant considérées comme hors de portée des machines. Ce domaine a connu des avancées spectaculaires ces dernières années et a permis des progrès significatifs dans plusieurs domaines en raison de leur précision accrue dans le traitement de vastes ensembles de données.

Dans ce chapitre, nous exposons tout d'abord une brève revue de l'IA, ML et DL. Ensuite, nous approfondirons le concept du DL et nous présenterons ses approches les plus importantes. Enfin, une comparaison des algorithmes du DL est effectuée à l'aide de critères couramment utilisés dans les articles de recherche, dans le but d'identifier les algorithmes les plus efficaces pour résoudre les problèmes de la Maintenance Prédictive.

### 2.2 Intelligence Artificielle

L'Intelligence Artificielle, désigne le développement de systèmes informatiques capables d'exécuter des tâches qui requièrent habituellement l'intelligence humaine. La norme ISO/IEC 22989:2020 la définit comme « La capacité d'acquérir, de traiter, de créer et d'appliquer des connaissances, détenues sous la forme d'un modèle, pour effectuer une ou plusieurs tâches données » [27]. Elle vise à développer des machines intelligentes qui possèdent la capacité de réflexion, de raisonnement et de résolution de problèmes similaires à celles des êtres humains. L'objectif est d'exploiter ces machines pour résoudre des défis complexes et améliorer le quotidien des divers domaines tels que la santé, les finances, les transports et l'éducation.

En 1956, une réunion s'est tenue au Dartmouth College, rassemblant un groupe d'experts en informatique passionnés par l'étude de l'intelligence. L'objectif de cette conférence était d'officialiser le terme "Intelligence Artificielle" et de l'établir comme un domaine de recherche. Depuis lors, de nombreux événements significatifs ont marqué son histoire. Voici dans la figure (2.1) quelques-uns des événements les plus importants qui ont façonné son développement [28].

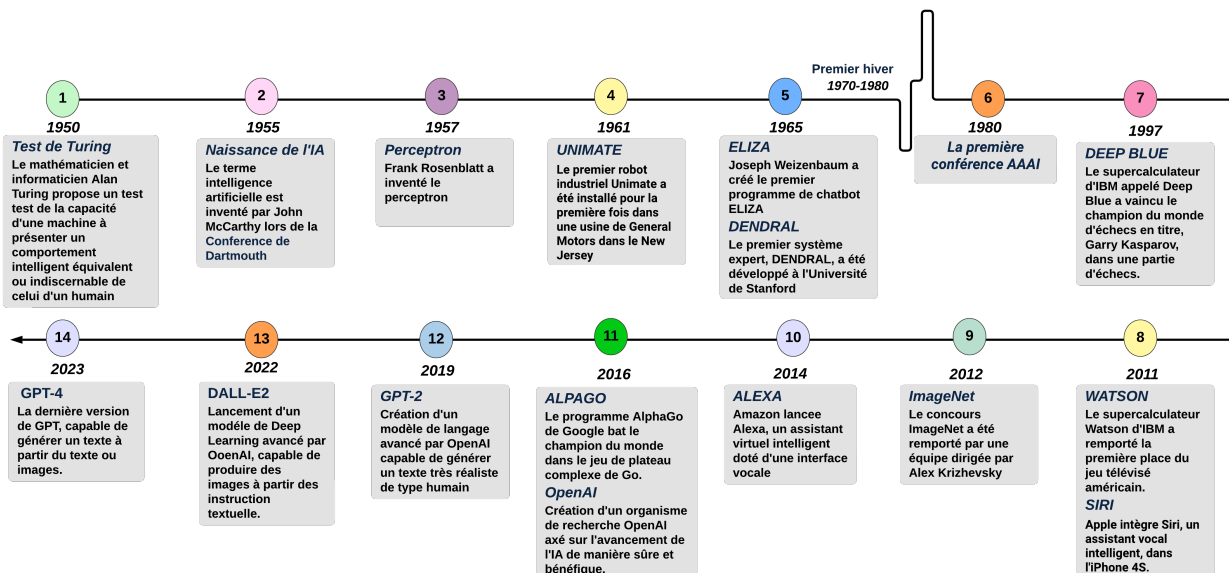


FIG. 2.1 : Les événements marquants dans l'histoire de l'IA

L'Intelligence Artificielle est un domaine étendu et dynamique qui englobe une multitude de sous-domaines et d'applications, comme illustré dans le schéma (2.2) ci-dessous [29].



FIG. 2.2 : Les domaines de l'Intelligence Artificielle

Actuellement, le Machine Learning est l'une des branches les plus répandues et les plus actives de l'Intelligence Artificielle, et il sera traité dans la section suivante.

## 2.3 Machine Learning

### 2.3.1 Définition du Machine Learning :

Machine Learning, une branche de l'Intelligence Artificielle, qui se concentre sur le développement d'algorithmes et de modèles capables d'acquérir des connaissances à partir de données, afin de faire des prédictions ou des prises de décision sans être explicitement programmés.

En 1959, Arthur Samuel a défini l'apprentissage automatique dans son livre intitulé "Machine Learning", comme étant "Un domaine d'étude qui confère aux ordinateurs la capacité d'apprendre sans être explicitement programmés" [30].

L'objectif du Machine Learning est donc de construire des systèmes capables d'apprendre et de s'améliorer à partir de l'expérience, plutôt que d'être programmés avec des instructions spécifiques pour chaque tâche.

### 2.3.2 Approches du Machine Learning :

En général, les approches du Machine Learning peuvent être catégorisées en fonction de l'effort humain requis pour les coordonner et de la manière dont elles utilisent les données étiquetées. La figure (2.3) résume les plus courantes.

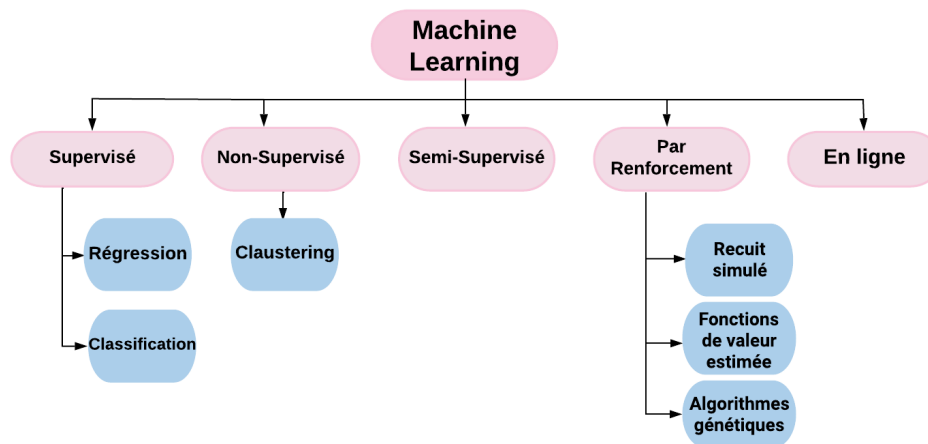


FIG. 2.3 : Approches du Machine Learning

- **Apprentissage Supervisé :**

Dans l'apprentissage supervisé, un algorithme apprend à faire des prédictions ou des décisions basées sur des données d'apprentissage étiquetées ; c'est-à-dire, les données d'entrée sont associées à une étiquette de sortie ou à une valeur cible correspondante, et l'algorithme apprend à mapper l'entrée à la sortie [31] [32].

Il est possible de subdiviser l'apprentissage supervisé en deux catégories distinctes : la régression et la classification.

- **Apprentissage Non Supervisé :**

L'apprentissage non supervisé vise à identifier des modèles ou des structures dans les données sans recourir à des étiquettes ou à des cibles explicites. En d'autres termes, les données ne sont pas étiquetées, et l'algorithme doit d'une façon autonome découvrir la structure sous-jacente et les relations entre les données [31] [32].

- **Apprentissage Semi-Supervisé :**

L'apprentissage semi-supervisé est une technique qui fusionne des aspects des deux types précédents. Dans ce type d'apprentissage, un petit ensemble de données étiquetées est combiné avec un ensemble plus vaste de données non étiquetées pour entraîner un modèle.

Les données étiquetées servent de guide lors du processus d'apprentissage, tandis que les données non étiquetées aident le modèle à généraliser et à acquérir des fonctionnalités plus robustes. Cette approche est particulièrement précieuse dans les situations où l'obtention de grandes quantités de données étiquetées peut s'avérer difficile ou coûteuse [33] [31].

- **Apprentissage par Renforcement :**

L'apprentissage par renforcement est une approche où un agent interagit avec un environnement afin d'apprendre à prendre des actions qui maximisent une récompense cumulative. L'agent apprend par essais et erreurs en recevant des commentaires sous forme de récompenses ou de pénalités en fonction de ses actions.

Cette technique implique trois éléments clé : l'agent, l'environnement et le signal de récompense. L'agent prend des mesures dans l'environnement en fonction de son état actuel et reçoit un signal de récompense qui indique à quel point ses actions étaient bonnes ou mauvaises. L'agent utilise ces commentaires pour mettre à jour sa politique et améliorer ses actions futures [34] [31].

- **Apprentissage en Ligne :**

Apprentissage en ligne (également connu sous le nom d'apprentissage incrémental ou out-of-core) est une approche fondamentalement différente, qui tient compte du fait que les environnements d'apprentissage peuvent changer d'une seconde à l'autre. Il correspond à un système capable de recevoir, d'intégrer et de mettre à jour le modèle en temps réel à l'aide de nouvelles données sans avoir à effectuer un apprentissage complet [35].

Le domaine du Machine Learning, est incroyablement vaste, offrant de nombreuses approches et méthodes pour résoudre des problèmes complexes. Le déploiement du Deep Learning est l'une de ces approches qui connaît une évolution fulgurante. Dans la section suivante, nous allons plonger plus en profondeur dans cette technique prometteuse.

## 2.4 Deep Learning

### 2.4.1 Définition du Deep Learning :

L'apprentissage en profondeur (DL), est une branche du Machine Learning, qui utilise des réseaux de neurones artificiels pour apprendre à partir des données. Ces réseaux de neurones

sont spécialement conçus pour imiter le comportement des neurones biologiques du cerveau humain. Ils fonctionnent en traitant les données à travers plusieurs couches et utilisent des algorithmes d'apprentissage automatique pour affiner de manière itérative les poids et les biais de chaque couche, obtenant finalement des prédictions ou des résultats précis [36].

Le Deep Learning se démarque par sa capacité à relever les défis liés à l'extraction de caractéristiques. Contrairement à d'autres méthodes, ses modèles peuvent apprendre de manière autonome à se concentrer sur les fonctionnalités pertinentes, nécessitant ainsi peu d'interventions de la part du programmeur. En substance, le Deep Learning permet aux ordinateurs d'acquérir la capacité d'apprendre par l'expérience, de manière similaire à l'apprentissage instinctif des êtres humains [37].

### 2.4.2 Notions de Deep Learning

#### 1. Neurones Biologiques :

Le cerveau humain est constitué d'un vaste réseau de cellules nerveuses spécialisées appelées neurones. Leur rôle principal est de transmettre des informations entre les cellules en utilisant des signaux électriques et chimiques [38].

Comme illustré dans la figure (2.4), chaque neurone est composé d'un corps cellulaire qui contient le noyau et les organites nécessaires au fonctionnement cellulaire, de dendrites qui sont de courtes extensions en forme de branches qui reçoivent des informations d'autres neurones ou cellules sensorielles et d'un axone qui est une extension longue et mince qui transporte les signaux du corps cellulaire vers d'autres neurones ou vers des cellules cible, telles que les muscles ou les glandes. Ces neurones sont interconnectés par des synapses, qui se forment lorsque la dendrite d'un neurone se connecte à l'axone d'un autre neurone, formant ainsi un système neuronal complexe [39].

Lorsqu'un neurone reçoit un signal électrique via son axone, il l'active ou le désactive en fonction de sa force, puis transmet ce signal à tous les neurones connectés via ses dendrites.

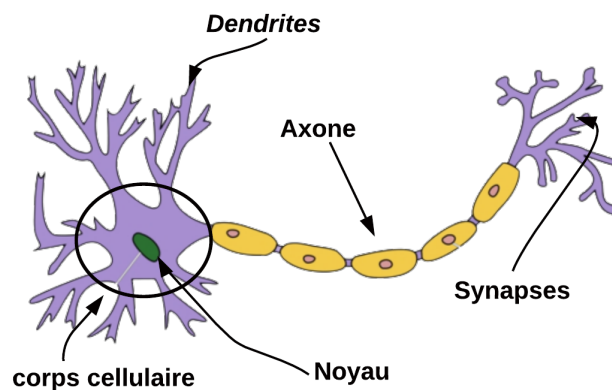


FIG. 2.4 : Structure du neurone biologique [40]

2. Neurone Artificiel :

Le neurone artificiel, connu également sous le nom de perceptron, est une unité de base dans les réseaux de neurones artificiels (ANN). Il est spécifiquement conçu pour imiter, de manière simplifiée, le fonctionnement des neurones biologiques présents dans le cerveau. Il a comme fonction le traitement de l'information, tel qu'il reçoit les entrées et produit un résultat à la sortie.

La figure (2.5) présente un schéma qui illustre bien la structure générale d'un neurone artificiel.

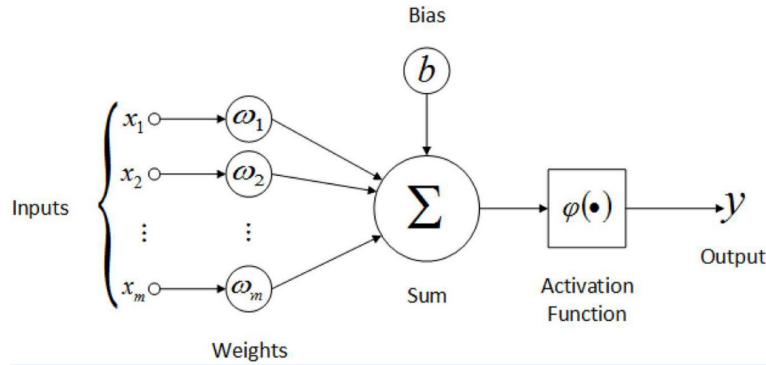


FIG. 2.5 : Neurone Artificiel [40]

Le tableau (2.1) met en évidence la comparaison possible entre le neurone biologique et les neurone artificiel.

TAB. 2.1 : L'analogie entre les neurones biologiques et les neurones artificiels

Neurone Biologique	Neurone Artificiel
Synapses	Connections pondérées
Axone	Sortie
Dendrites	Entrées
Sommateur	Fonction d'activation

3. Réseau de Neurones Artificiels :

Un réseau de neurones artificiel (ANN) est un modèle mathématique, c'est une collection interconnectée de neurones artificiels organisés en couches et possèdent des entrées et des sorties [41]. Chaque neurone dans un ANN reçoit des signaux d'entrée provenant d'autres neurones et les traite en appliquant une fonction mathématique non-linéaire. Cette opération permet au neurone de calculer sa sortie. La sortie d'un neurone est ensuite transmise à d'autres neurones du réseau par le biais de connexions pondérées entre eux. Pendant la phase d'apprentissage, les poids des connexions entre les neurones sont ajustés afin de minimiser l'écart entre les prédictions du réseau et les sorties réelles [42].

Dans sa forme la plus basique, un ANN ne peut avoir que trois couches de neurones :

- La couche d'entrée : ou les données entrent dans le système.



- La couche cachée : ou les informations sont traitées.
- La couche de sortie : ou le système décide de ce qu'il doit faire en fonction des données.

### 2.4.3 Machine learning vs Deep Learning

Bien que le Deep Learning relève de la catégorie du Machine Learning, ces deux approches présentent des dissemblances distinctes, comme le montre le tableau (2.2).

TAB. 2.2 : Tableau comparatif entre Machine learning et Deep learning

Caractéristiques	Machine Learning	Deep learning
Intervention humaine [43] [44][45]	Nécessite une intervention humaine	Moins ou pas besoin d'intervention humaine
Type de données [44]	Besoin de données structurées	fonctionnement avec des données structurées et non structurées
Temps de formation [45] [46] [44]	Calcul rapide (secondes à heures)	calcul lent (jours à semaines)
Extraction de fonctionnalités [47] [44]	Nécessite une étape d'extraction des données par un expert	Ne nécessite pas l'extraction des caractéristiques ; il tente plutôt d'apprendre à partir de données brutes par lui-même
Exigences en données [43] [46][44]	Nécessite de grandes quantités de données d'entraînement	Nécessite d'énormes quantités (plus) de données d'entraînement
Exigences en matérielles [43][46][44]	Peut fonctionner sur des appareils bas de gamme	nécessite l'utilisation de GPU et donc du système haut de gamme

### 2.4.4 Approches de Deep Learning :

Il existe plusieurs techniques de Deep Learning qui sont largement utilisées dans différents domaines. Voici quelques-unes des approches les plus courantes :

#### 1. Réseaux de Neurons Feedforward (FNN) :

La première architecture, la plus courante et la plus simple, est caractérisée par une série de neurones empilés disposés en couches. Les neurones de chaque couche sont connectés à tous les neurones de la couche suivante, transmettant leur sortie en entrée. Cependant, il n'y a pas de connexions entre les neurones des couches précédentes ou au sein de la même couche. L'architecture se compose d'une couche d'entrée (input layer), de couches cachées (hidden layers) et d'une couche de sortie (output layer) comme illustré dans la figure (2.6). Pour former le réseau de neurones, on lui fournit des paires de caractéristiques d'entrée et

de caractéristiques cibles, qui sont utilisées pour établir la relation entre elles. Le réseau y parvient en minimisant l'erreur qu'il produit et en mappant les données d'entrée à la sortie souhaitée [48].

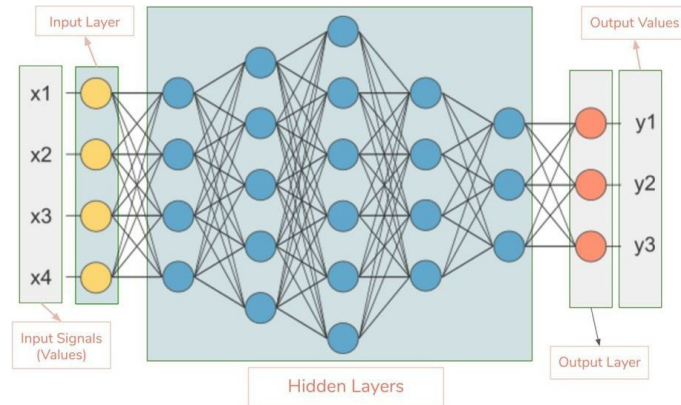


FIG. 2.6 : Réseaux de neurones feedforward [40]

2. Réseaux de Neurones Convolutionnels (CNN) :

CNN est un type de réseau à anticipation (feedforward) qui utilise des filtres convolutifs pour maintenir la connectivité entre les neurones. Il s'inspire du cortex visuel des animaux et trouve des applications dans divers domaines tels que la reconnaissance d'images et de signaux, les systèmes de recommandation et le traitement du langage naturel. Comme illustré dans la figure (2.7), CNN est composée des couches de convolution qui sont généralement linéaires et sont suivies par l'application d'une fonction d'activation, elles effectuent des calculs de convolution pour extraire des caractéristiques importantes. Par la suite, les couches de pooling sont utilisées pour réduire la dimensionnalité des données. Enfin, les couches de classification (couches entièrement connectées (fully connected)) sont utilisées pour prédire la classe la sortie [48].

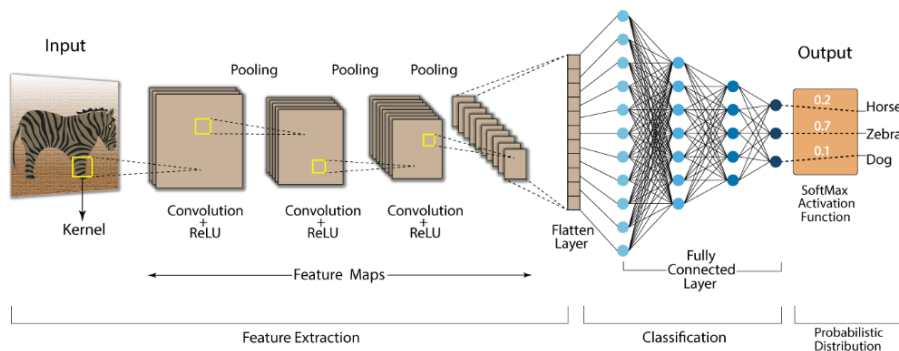


FIG. 2.7 : Réseaux de neurones convolutionnels [40]

3. Réseau de Neurones Récurrents (RNN) :

Un réseau de neurones récurrents est un réseau qui présente des connexions récurrentes (rétroactifs), avec toutefois des boucles de mémoire récurrentes (mémoire interne) qui prennent en entrée les couches et les états précédents et/ou identiques. Cela leur donne une possibilité unique de modéliser la dimension temporelle et une séquence arbitraire d'événements et d'entrées. Néanmoins, les RNNs se heurtent au problème de disparition

du gradient pour apprendre à mémoriser des événements passés. Des architectures particulières répondent à ce dernier problème, on peut citer en particulier les réseaux Long short-term memory [49].

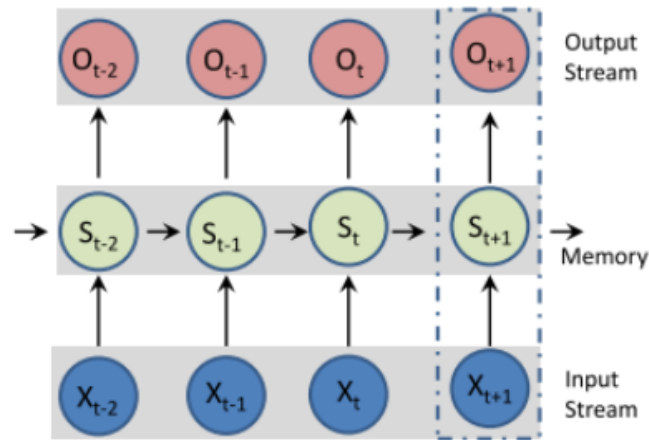


FIG. 2.8 : Réseaux de neurones récurrents [50]

#### 4. Long Short-Term Memory (LSTM) :

LSTM, représente l'architecture de réseau de neurones récurrents la plus couramment utilisée en pratique. Elle résout efficacement le problème de la disparition de gradient. Les LSTM sont très utiles dans la prédiction des séries chronologiques. Leur capacité à se souvenir des entrées précédentes les rend particulièrement adaptés à cet usage. En-dehors de cette application, les LSTM sont également employés pour créer de la musique en composant des notes et pour reconnaître la voix [51].

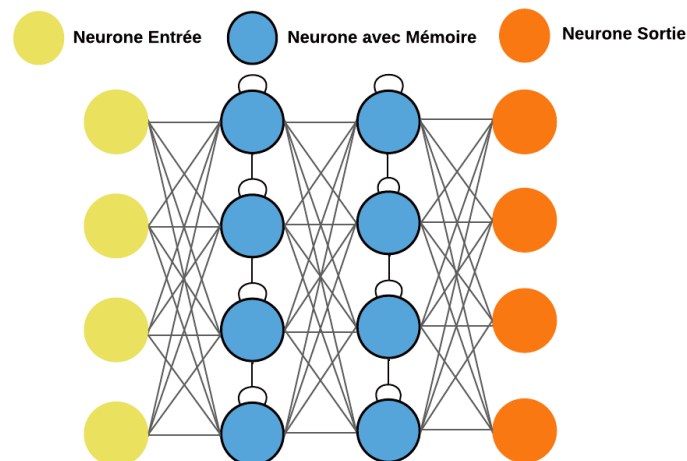


FIG. 2.9 : Long short-term memory [52]

#### 5. AutoEncodeurs (AE) :

L'autoencodeur est basé sur le concept de décomposition en valeurs singulières afin d'extraire les caractéristiques non-linéaires qui représentent au mieux les données d'entrée dans un espace de dimension réduite. Comme le montre la figure (2.10), il se compose de deux parties : un encodeur qui mappe les données d'entrée vers l'espace latent encodé, et

le décodeur, qui projette les données de l'espace latent vers l'espace reconstruit ayant la même dimension que les données d'entrée. Le réseau est entraîné à minimiser l'erreur de reconstruction, qui est la différence entre les données d'entrée et de sortie [48].

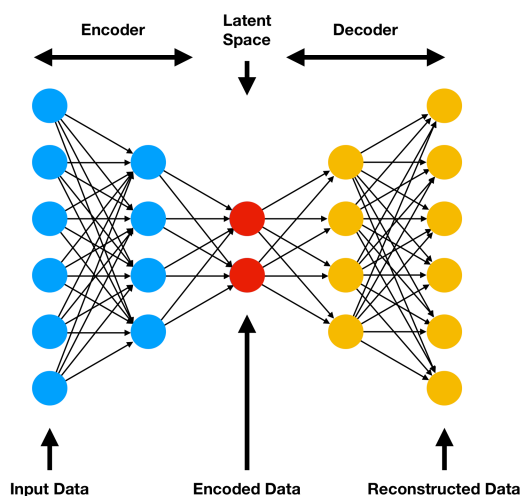


FIG. 2.10 : AutoEncodeurs [40]

## 6. Réseaux Adversariaux Génératifs (GAN) :

Le GAN s'agit d'un modèle comme le montre la figure (2.11), qui comprend deux réseaux de neurones : le générateur est chargé de produire des données synthétiques en prenant du bruit aléatoire en entrée, tandis que le discriminateur est chargé de faire la distinction entre les données réelles et synthétiques en examinant les deux types [53]. Notamment, ces réseaux sont entraînés simultanément et ils s'affrontent dans un jeu adversarial [54] [55].

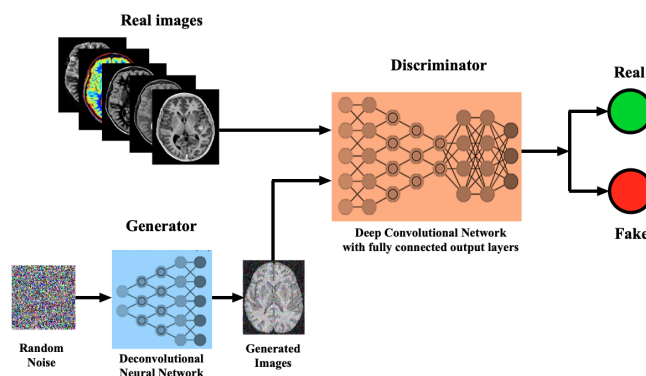


FIG. 2.11 : Réseaux Adversariaux Génératifs [40]

## 7. Machines de Boltzmann Restreintes (RBM) :

Les machines de Boltzmann restreintes sont un autre algorithme qui se compose d'un réseau à deux couches (figure (2.12)) comprenant une couche cachée et une couche visible. Dans les RBMs, seules les couches cachées et visibles sont connectées ; cependant, il n'y a aucune corrélation entre les neurones de la même couche. La couche visible est responsable des données d'entrée, tandis que la couche cachée effectue l'extraction des caractéristiques. De plus, dans les RBMs, les nœuds cachés sont conditionnellement auto-déterminants et

n'ont aucune interdépendance dans les RBMs. Les décalages et les poids des couches visibles et cachées sont mis à jour de manière itérative de telle sorte que la couche visible puisse estimer les données d'entrée [56].

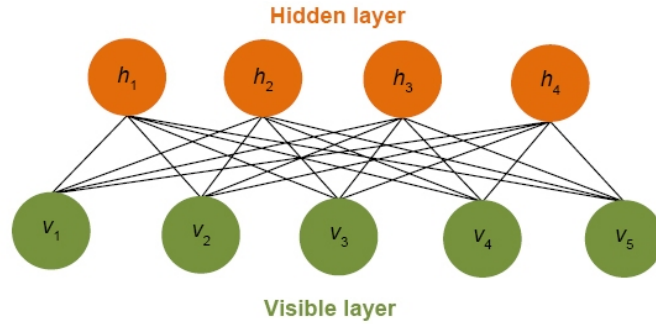


FIG. 2.12 : Machines de Boltzmann Restreintes [40]

### 8. Réseaux de Croyances Profondes (DBN) :

Un Deep Belief Network peut être considéré comme une composition de RBMs, où la couche cachée de chaque sous-réseau est connectée à la couche visible du RBM suivant. Les DBN ont des connexions non dirigées uniquement dans les deux couches supérieures et des connexions dirigées vers les couches inférieures. L'initialisation d'un DBN est obtenue grâce à une stratégie d'apprentissage glouton couche par couche efficace utilisant l'apprentissage non supervisé, puis elle est affinée en fonction des sorties cibles [50].

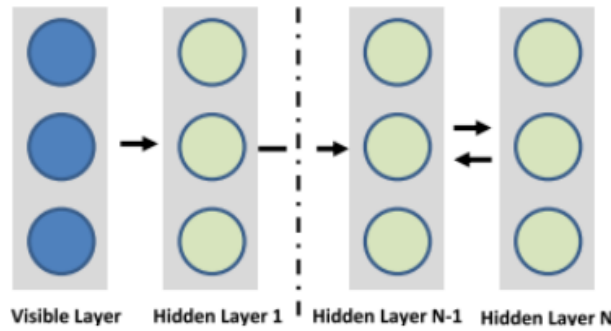


FIG. 2.13 : Réseaux de croyances profondes [40]

## 2.5 Deep Learning pour la Maintenance Prédictive

En raison de l'augmentation des exigences en matière de fiabilité, de disponibilité, de maintenabilité et de sécurité des systèmes, les stratégies de maintenance traditionnelles deviennent moins efficaces et obsolètes. De plus, la révolution de l'Industrie 4.0 offre un soutien plus pratique au vaste développement de la maintenance prédictive dans la pratique [57].

Récemment, le Deep Learning a démontré une capacité supérieure dans l'apprentissage des caractéristiques, la classification des défauts et la prédiction des défauts grâce à des transforma-

tions non-linéaires à plusieurs couches. Les Auto-Encoder, les Réseaux de Neurones Convolutionnels, les Réseaux de Croyance Profonde (DBN) et d'autres modèles d'apprentissage profond sont largement utilisés dans le domaine de la Maintenance Prédictive [48], et le nombre de publications dans ce domaine a augmenté de manière exponentielle ces dernières années comme le montre la figure (2.14).



FIG. 2.14 : Evolution du nombre de publications sur le Deep Learning pour la Maintenance Prédictive dans le moteur de recherche Google Scholar [48]

### 2.5.1 Revue de la Littérature

Ce qui suit, présente une revue de la littérature sur certains des articles de recherche les plus récents sélectionnés parmi différentes sources, qui abordent la combinaison de Deep Learning et de la Maintenance Prédictive.

- Dans l'article [1] intitulé "A recurrent neural network based health indicator for remaining useful life prediction of bearings", L.Guo et al abordent les défis liés à la prédiction de la durée de vie résiduelle (RUL) des roulements dans les méthodes prévisionnelles basées sur les données, donc ils proposent un nouvel indicateur de santé basé sur un réseau neuronal récurrent (RNN-HI) pour prédire la RUL des roulements. Cependant, il faut noter que la qualité et la gestion des données de vibration peuvent constituer un défi lors de l'application de l'approche RNN-HI dans des environnements réels.
- Haiyue Wu et al dans leurs article [2] intitulé « Avoiding Environmental Consequences of Equipment Failure via an LSTM-Based Model for Predictive Maintenance », propose un modèle RNN basé sur LSTM pour détecter la panne d'un système de fabrication et prédire son état de santé futur. Ce modèle prend en compte les caractéristiques séquentielles dans le temps des données historiques des capteurs. De plus, une nouvelle manière d'entrer les caractéristiques a été décrite pour contenir ces informations temporelles. Afin de valider le modèle proposé, celui-ci a été implémenté sur un jeu de données de vibration de roulement d'un moteur du « Center for Intelligent Maintenance Systems de l'Université

de Cincinnati ». Les résultats sont illustrés par une matrice de confusion et sont comparés à ceux d'un autre modèle basé sur SVM. Les résultats obtenus ont montré que le modèle avait une bonne accuracy sur les états critiques et un peu moins sur les états normaux. Une limitation de cette étude réside dans son évaluation qui se limite à l'accuracy et ne prend pas en compte d'autres métriques importantes pour évaluer les performances du modèle.

- Dans l'article [3] intitulé "A data-driven fault detection and diagnosis scheme for air handling units in building HVAC systems considering undefined states" W. Yun et al proposent une approche intéressante pour la détection et le diagnostic des défauts dans les systèmes HVAC. L'utilisation de l'auto-encodeur supervisé (SAE) comme modèle de détection et de diagnostic des défauts présente une meilleure détection et un meilleur diagnostic des défauts, ce qui est essentiel pour assurer une maintenance fiable des systèmes HVAC. Cependant, il est important de noter sa dépendance à des données étiquetées pour l'apprentissage, il peut être difficile d'obtenir des données de défauts étiquetées en quantité suffisante pour entraîner le modèle. Cela peut limiter la capacité du SAE à généraliser et à détecter des défauts qui n'ont pas été explicitement représentés dans les données d'apprentissage.
- Pour résoudre le problème limitation des données étiquetées, une approche semi-supervisée de FDD est proposée pour les systèmes HVAC, basée sur le réseau antagoniste génératif modifié (modified GAN) par cheng et al dans leur papier « semi-supervised approach to fault detection and diagnosis for building HVAC systems based on the modified generative adversarial network » [58], le modèle proposé est validé à l'aide des données d'un système HVAC, collectées dans un immeuble de bureaux du Oak Ridge National Laboratory dans le Tennessee, aux États-Unis ; l'immeuble est réservé aux expérimentations et n'est pas occupé. Les résultats obtenus lors des expériences démontrent la validité de l'approche suggérée. Néanmoins, il est important de souligner que l'efficacité de la méthode dépend de la représentativité des défauts introduits dans les données collectées. Si les scénarios de défauts simulés ne couvrent pas l'ensemble des possibilités de défauts réels, le modèle peut avoir des difficultés à détecter efficacement des défauts inédits.
- Dans l'article [4] intitulé "Prediction of Machine Failure in Industry 4.0: A Hybrid CNN-LSTM Framework", Abdul Wahid et al ont proposé un modèle d'apprentissage profond hybride (CNN-LSTM) pour la Maintenance Prédictive (PdM). Par la suite, ils ont mené des expériences sur la nouvelle méthode, et ils l'ont comparé avec le CNN et LSTM en utilisant le même ensemble de données provenant de l'étude de cas de Microsoft. Ces données ont été collectées sur une période d'une année et concernent un total de 100 machines. En analysant les indicateurs d'évaluation des modèles CNN, LSTM et CNN-LSTM hybride, il a été constaté que le modèle CNN-LSTM présentait la précision de prédiction la plus élevée. Une des limites de cette étude réside dans le fait que la nouvelle approche n'a pas été suffisamment testée et validée en simulant divers scénarios avec différentes bases de données.

### Synthèse :

Ces articles, ainsi que d'autres références pertinentes, ont chacun exploré les diverses approches de Deep Learning pour aborder le domaine de la Maintenance Prédictive. Notre travail s'inscrit dans la continuité de ces contributions présentes dans la littérature. Cependant, notre contribution se distingue en ce sens que nous nous concentrons spécifiquement sur la comparaison de ces approches afin de sélectionner l'algorithme le plus performant.

Le tableau récapitulatif (2.3) résume les différents articles cités ci-dessus :



TAB. 2.3 : Tableau récapitulatif des articles sélectionnés dans la littérature sur le Deep Learning pour Maintenance Prédictive

Titre	Description	Limitation	Date de consultation
A recurrent neural network based health indicator for remaining useful life prediction of bearings [1].	Construire un indicateur de santé basé sur un RNN-HI pour prédire la durée de vie résiduelle (RUL) des roulements.	Un défi peut se poser lors de l'application de cette méthode dans la réalité en ce qui concerne la qualité et la gestion des données de l'AHU.	Avril 2023
Avoiding Environmental Consequences of Equipment Failure via an LSTM-Based Model for Predictive Maintenance [2]	Proposer un modèle d'apprentissage profond hybride (CNN-LSTM) pour la Maintenance Prédictive (PdM).	L'évaluation du modèle est limité en utilisant que l'accuracy sans prendre en compte d'autres métriques et facteur pour justifier la performance.	Avril 2023
A data-driven fault detection and diagnosis scheme for air handling units in building HVAC systems considering undefined states [3].	Construire une approche pour la détection et le diagnostic des défauts dans les systèmes HVAC utilisant l'auto-encodeur supervisé (SAE)	La méthode dépend fortement de la disponibilité de données étiquetées pour l'apprentissage.	Avril 2023
Semi-supervised approach to fault detection and diagnosis for building HVAC systems based on the modified generative adversarial network [58].	Construire un modèle semi-supervisé basé sur l'algorithme GAN modifié pour résoudre le problème de limitation des données étiquetés.	Les données mises en œuvre sont des données de simulation, ce qui signifie qu'elles peuvent ne pas englober toutes les éventualités de défauts réels.	Avril 2023
Prediction of Machine Failure in Industry 4.0: A Hybrid CNN-LSTM Framework [4].	Proposer un modèle RNN basé sur LSTM pour détecter la défaillance d'un système de fabrication, en tenant compte des caractéristiques séquentielles temporelles des données.	L'approche proposé n'a pas été suffisamment testée et validée en simulant divers scénarios avec différentes bases de données.	Avril 2023

## 2.5.2 Étude Comparative des Algorithmes de Deep Learning pour la Maintenance Prédictive

L'objectif de cette étude est d'effectuer une comparaison approfondie entre plusieurs algorithmes de Deep Learning utilisés dans la Maintenance Prédictive.

Notre démarche consistera tout d'abord à établir des critères de comparaison clairs et pertinents, permettant ainsi d'évaluer et de quantifier la performance de ces algorithmes d'une manière objective.

### 1. Définitions des Critères de Comparaison :

Comme illustré dans la carte (2.15), les critères essentiels utilisés pour évaluer les algorithmes DL sont les suivants :

- **Accuracy :**

L'accuracy représente le rapport entre le nombre de prédictions correctes et le nombre total d'échantillons dans un ensemble de données.

Accuracy : Faible [ $< 25\%$ ]\* /Moyenne [ $25\%-50\%$ \*\* /grande [ $50\%-75\%$ \*\*\* /Très grande [ $75\%-100\%$ \*\*\*\*

- **Temps d'Apprentissage :**

Le temps d'apprentissage fait référence à la durée nécessaire pour que l'algorithme apprenne à partir des données fournies ; et il varie en fonction de la taille des données et de l'algorithme que nous utilisons.

Temps d'apprentissage : Moyen\*/ Bon\*\* /Très bon\*\*\* /Excellent\*\*\*\*

- **Temps de Prédiction :**

Le temps de prédiction se réfère à la durée nécessaire pour qu'un algorithme génère une prédiction ou une estimation à partir des données d'entrée, et il varie en fonction de la taille des données et de l'algorithme utilisé.

Temps de prédiction : Moyen\*/ Bon\*\* /Très bon\*\*\* /Excellent\*\*\*\*

- **Travaux de Recherche Connexe :**

Consiste au nombre des articles qui utilisent un des algorithmes DL dans le domaine de la PdM.

Nombre d'article : Très petit [ $<5$ ]\* /Petit [ $5-8$ \*\* / grand] $8-12$ \*\*\* /Très grand [ $12-16$ \*\*\*\*

- **Paramétrisation :**

Les paramètres sont les valeurs à régler lors de la configuration d'un algorithme. Ces valeurs ont une incidence sur le fonctionnement de l'algorithme, tels que la tolérance aux erreurs, le nombre d'itérations ou les variantes du comportement de l'algorithme.

Nombre de paramètre est : Grand [ $>8$ ]\* /Moyen [ $4-8$ \*\* /Petit [ $2-4$ \*\*\* /Très petit [ $<2$ \*\*\*\*

- **Taille de Mémoire :**

La taille de mémoire est la capacité requise pour contenir les paramètres du modèle, y compris les poids et les biais des différentes couches, les données et les variables.

Taille de mémoire : Très grande\* /Grande\*\* /Petite\*\*\* /Très Petite\*\*\*\*

- **Flexibilité :**

La flexibilité fait référence à la capacité d'un réseau à s'adapter aux modèles des bases de données.

Flexibilité : Faible [ $<25\%$ ]\* / Moyenne [ $25\%-50\%$ ]\*\* /Grande [ $50\%-75\%$ ]\*\*\* /Très grande [ $75\%-100\%$ ]\*\*\*\*

- **Facilité d'Interprétation :**

Ce critère fait référence à la capacité de comprendre et d'expliquer comment un modèle prend des décisions ou des prédictions.

La facilité d'interprétation : Difficile\* /moyen\*\* /Facile \*\*\* /Très facile \*\*\*\*

La carte mentale (Mind Map) dans la figure (2.15), résume les critères d'évaluation des algorithmes DL identifiés dans les articles de recherches.

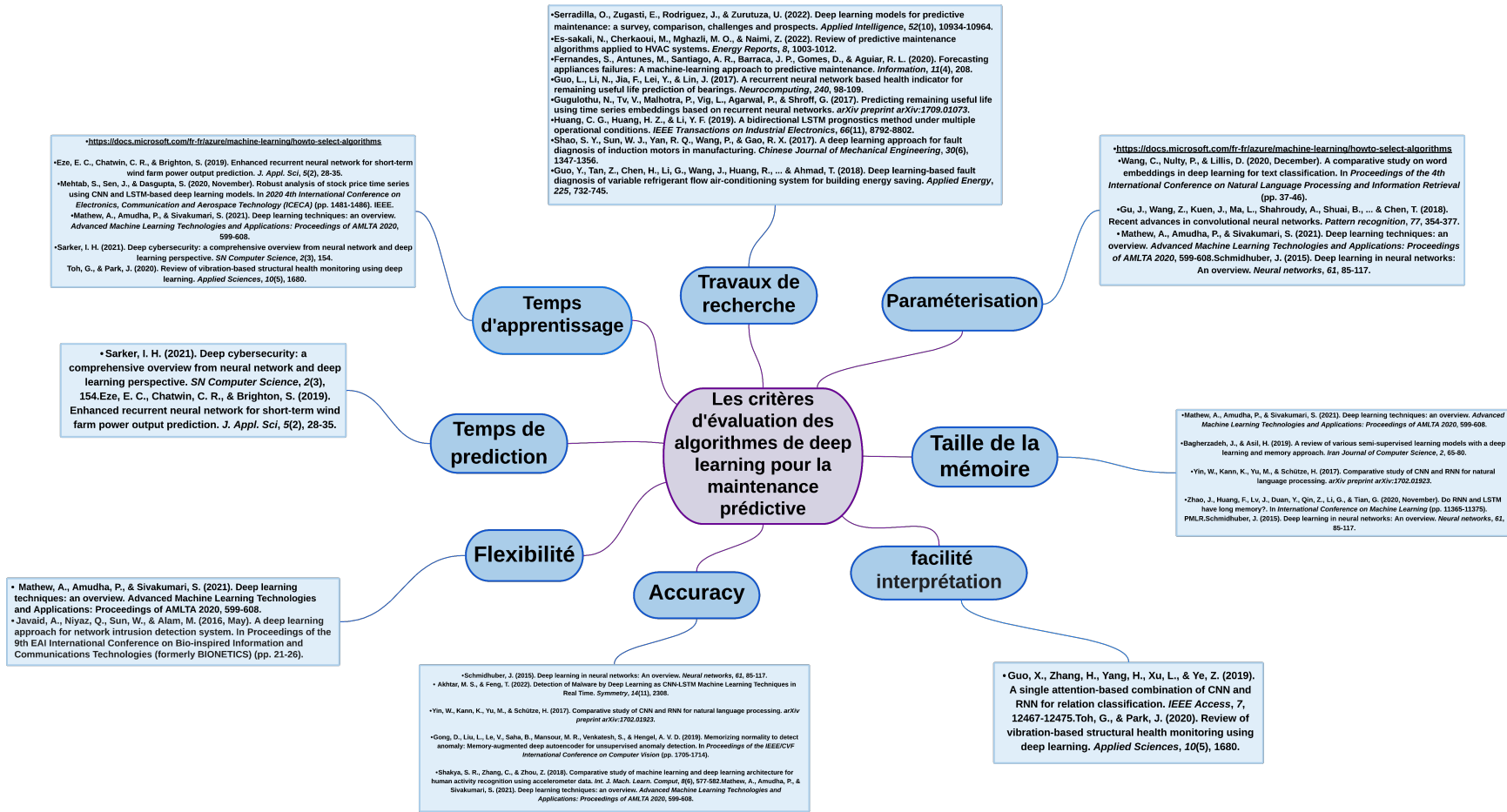


FIG. 2.15 : Critères d'évaluation des algorithmes DL pour la PdM à partir de l'état de l'art

2. Table Comparative des Algorithmes du DL pour la PdM

La table (2.4) est une table comparative des différents algorithmes de Deep Learning pour la Maintenance Prédictive, basée sur les critères mentionnés précédemment.

TAB. 2.4 : Comparaison entre les Algorithmes DL pour la PdM selon les Critères

Critère/Algorithme	CNN	RNN	LSTM	AE	GAN	FNN	RBM	DBN
Accuracy	****	****	****	***	***	***	***	****
Temps d'apprentissage	***	**	**	***	*	**	****	*
Paramétrisation	**	**	**	****	**	**	****	**
Taille de mémoire	***	***	****	**	**	**	**	**
Temps de prédiction	**	****	****	***	***	***	*	**
Flexibilité	****	****	****	**	**	**	****	**
Facilité d'interprétation	**	*	*	*	*	*	*	*
Travaux de recherche	***	***	****	***	**	*	**	**
Score	23	23	25	21	16	16	21	16

(\*\*\*\* étoiles représentent la meilleure performance et \* étoile la pire performance)

À partir de la table (2.4) de l'évaluation des algorithmes de Deep Learning, nous additionnons les étoiles pour indiquer les algorithmes à choisir ; les sommes sont présentées dans la table (2.5).

TAB. 2.5 : Scores des algorithmes par ordre décroissant

No	Algorithms	Score
<b>1</b>	<b>Long Short-Term Memory</b>	<b>25</b>
<b>2</b>	<b>Recurrent Neural Networks</b>	<b>23</b>
<b>3</b>	<b>Convolutional Neural Networks</b>	<b>23</b>
4	Autoencoders	20
5	Boltzmann machines rerestraint	20
6	Generative Adversarial Networks	16
7	Deep Belief Networks	16
8	Feedforward Neural Networks	15

Sur la base du tableau précédent (table (2.5)), l'algorithme LSTM a revendiqué la première position, l'algorithme RNN venant en deuxième position, suivi de l'algorithme CNN.

Les résultats de l'étude suggèrent que parmi les différents algorithmes considérés pour la PdM, ces trois excellent particulièrement et peuvent être considérés comme les plus performants selon les critères et les articles sélectionnés.

### 2.6 Conclusion

La Maintenance Prédicative est une application importante des algorithmes de Deep Learning qui peut aider les organisations à minimiser les défaillances du système, à améliorer la prévisibilité et à augmenter la disponibilité du système. Avec l'augmentation des espaces de données industriels dans le monde, les solutions DL ont gagné en popularité pour le PdM. Cependant, la sélection de l'architecture appropriée pour chaque cas d'utilisation peut être difficile en raison des nombreux facteurs impliqués.

L'objectif principal de ce chapitre été d'identifier un algorithme DL approprié pour PdM. Lors de l'examen de divers algorithmes DL, il devient évident que chaque algorithme possède son propre ensemble de forces et de faiblesses en ce qui concerne la précision, la durée de la formation et la compréhensibilité. Les résultats de l'étude indiquent que LSTM a obtenu le score le plus élevé pour l'ensemble de critères analysés, suivi de RNN puis de CNN. Ces trois algorithmes de DL sont considérés comme les plus efficaces pour la PdM.

Le chapitre suivant sera dédié à la mise en place d'une étude expérimentale sur les algorithmes sélectionnés. Cette étude impliquera l'utilisation d'ensembles de données réels et de simulation liés à la PdM. En se concentrant sur les résultats en termes de précision, de rappel, d'exactitude ou d'autres mesures de performance appropriées, l'algorithme le plus approprié pour les tâches PdM sera sélectionné.

Deuxième partie

Partie Expérimentale

## Chapitre 3

# Implémentation et Evaluation des Meilleurs Algorithmes DL pour la PdM



## 3.1 Introduction

D’après l’étude comparative effectuée dans le chapitre 2, les approches LSTM, RNN et CNN ont été sélectionnées comme les algorithmes les plus appropriés pour la Maintenance Prédictive. Dans ce chapitre, nous abordons la partie expérimentale de ce travail.

Ce chapitre expose les résultats obtenus en mettant en œuvre les différents algorithmes selon deux axes d’apprentissage : classification binaire et classification multi-classe. Nous utilisons à la fois des données réelles provenant d’un compresseur collectées lors d’un stage effectué à la raffinerie d’Alger, et des données de simulation téléchargées sur Internet.

La première partie de ce chapitre décrit la méthodologie générale, les bases de données utilisées et l’implémentation des différentes architectures. Le reste du chapitre consiste à évaluer la capacité prédictive de nos modèles sur nos ensembles de données à l’aide des métriques appropriées que nous définirons. Enfin, nous comparons les performances obtenues avec celles d’autres modèles issus de travaux connexes.

## 3.2 Méthodologie

Afin d’accomplir notre travail et d’atteindre notre objectif, ainsi que de répondre à la problématique de la sélection d’un algorithme de DL adapté au PdM, nous avons développé une approche méthodologique qui oriente notre processus de travail. Cette méthodologie est illustrée dans le schéma (3.1) ci-dessous.

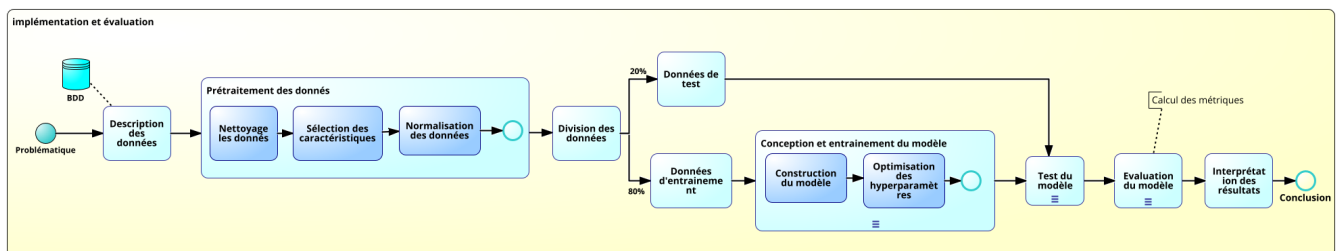


FIG. 3.1 : La structure générale du processus de l’étude

Les prochaines sections fourniront des explications détaillées ainsi que la mise en œuvre des différentes étapes.

## 3.3 Description des Données d’Implémentation

L’expérimentation s’est appuyée sur l’utilisation de deux ensembles de données distincts, comprenant un ensemble de données réelles (BDD1) ainsi qu’un ensemble de données simulées (BDD2) qui est utilisé dans deux différents cas (classification binaire et multi-classes) :

### 3.3.1 Base de Données 1

La base de données "BDD1" est constituée d'un ensemble de données réelles, collectées lors d'un stage réalisé à la raffinerie d'Alger - SIDI ERZINE. Ces données sont spécifiquement liées aux capteurs du compresseur 520-K-052A/B. Une présentation de la raffinerie ainsi de l'équipement est fournie en annexe (3.9).

"BDD1" se compose de 7419 instances avec 11 variables mesurables qui sont des mesures échantillonnées chaque une heure par capteur tout au long d'une année, ils mesurent différentes caractéristiques de l'équipement comme la température, la pression et la vibration. De plus, chaque échantillon est également étiqueté avec une classe correspondant à l'état de l'équipement (1=panne/0=non-panne). Les mesures des capteurs sont répertoriées dans le tableau (3.1) ci-dessous :

TAB. 3.1 : Les variables mesurés du BDD1

Variable	Désignation	Unité
TZI1905BA	Température	Celsius [°C]
TZI1905BB	Température	Celsius [°C]
TZI1905BC	Température	Celsius [°C]
TZI1906BA	Température	Celsius [°C]
PZI1910	Préssion	Bar [bar]
PZI1920B	Préssion	Bar [bar]
VI1903B	Vibration	m/s <sup>2</sup>
VI1904B	Vibration	m/s <sup>2</sup>
VI1901B	Vibration	m/s <sup>2</sup>
ZI1901B	Déplacement	Micrometer [ $\mu m$ ]
ZI1902B	Déplacement	Micrometer [ $\mu m$ ]

### 3.3.2 Base de Données 2

La base de données "BDD2" est constituée d'un ensemble de données de simulation téléchargée à partir de répertoire des données UCI :

<https://archive.ics.uci.edu/ml/datasets/AI4I+2020+Predictive+Maintenance+Dataset>.

"BDD2" se compose de 10 000 instances avec 14 caractéristiques en colonnes qui contiennent au total 5 variables mesurables qui sont des mesures échantillonnées, ils mesurent différentes caractéristiques de l'équipement comme la température, la vitesse de rotation et le torque ; et 3 variables catégorielles identifiant le type de produit. De plus, chaque échantillon est étiqueté selon l'état de l'équipement (1=panne/0=non-panne) ; et également selon le mode de panne qui peut être : Heat Dissipation Failure, Power Failure, Overstrain Failure, Tool Wear Failure, Random Failures ou No Failure. Les mesures des capteurs sont répertoriées dans le tableau (3.2) ci-dessous :

TAB. 3.2 : Les variables du BBD2

Variable	Désignation	Unité
Air temperature	Température de l'air	kelvin [k]
Process temperature	Température de processus	kelvin [k]
Rotational speed	Vitesse de rotation	tours par minute [rpm]
Torque	Couple	newton mètres [Nm]
Tool wear	Usure de l'outil	Minutes [min]
H	High : Variante de qualité supérieure.	/
M	Meduim : Variante de qualité intermédiaire.	/
L	Low : variante de qualité inférieure.	/

### 3.4 Prétraitement des Données

Habituellement, il n'y a pas de données propres que nous utilisons directement dans la modélisation et donc, beaucoup de défis rencontrés doivent être traités.

Dans cette section, nous abordons les techniques utilisées sur la base de données "BDD1" afin d'assurer la qualité des données.

#### 3.4.1 Nettoyage des Données

Cette étape consiste à traiter les valeurs manquantes, les doublons ou incorrectes de la base de données.

#### 3.4.2 Sélection des Caractéristiques

Dans cette étape, nous introduisons un heatmap pour visualiser les relations de corrélation entre les différentes variables, afin de sélectionner les caractéristiques les plus pertinentes pour améliorer les performances du modèle.

Le heatmap correspondant à "BDD1" est présenté dans la figure (3.2) :

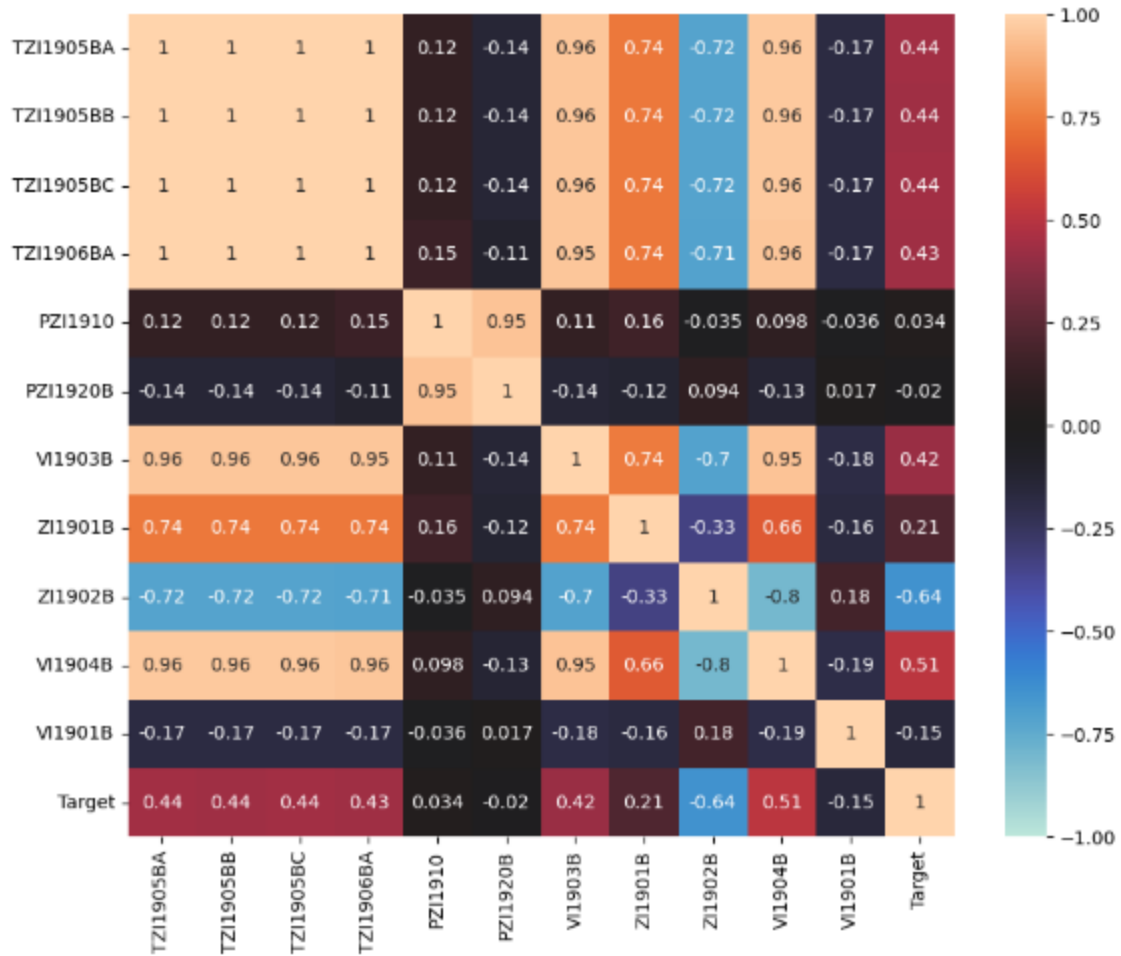


FIG. 3.2 : Heatmap de corrélation de BDD1

On remarque qu'il existe une très forte corrélation ( $> 0,8$ ) entre certaines caractéristiques : (TZI1905BA,TZI1905BB), (TZI1905BA,TZI1905BC), (TZI1905BA,TZI1906BA), (TZI1905BB,TZI1905BC), (TZI1905BB,TZI1906BA), (TZI1905BC,TZI1906BA), (PZI1910,PZI1920B), (VI1903B,TZI1905BA), (VI1903B,TZI1905BB), (VI1903B,TZI1905BC), (VI1903B,TZI1906BA).

Dans cette situation, l'utilisation simultanée de ces caractéristiques ne ferait qu'introduire du bruit dans nos modèles. Par conséquent, cela pourrait affecter les performances de certains algorithmes de Deep Learning. Ainsi, ces caractéristiques spécifiques seront ciblées pour être supprimées. Une seule de ces caractéristiques devrait être utilisée, celle qui fournit le plus d'informations pertinentes.

Le résultat de l'analyse du coefficient de corrélation pour chaque variable numérique par rapport à la cible révèle que la plupart des variables ne présentent pas de corrélation significative, à l'exception de (ZI1902B). Cette variable présente une corrélation négative modérée, suggérant que la valeur du capteur est généralement plus basse en cas de panne. Par conséquent, il est important d'utiliser ce capteur pour détecter la panne.

À la fin, les caractéristiques retenues pour chaque base de données, afin d'implémenter les modèles sont motionnées dans la table (3.3).

TAB. 3.3 : Les caractéristiques de simulation des bases de données

Base de Données	Caractéristiques
BDD1	“TZI1905BA”, “PZI1920B”, “VI1904B”, “VI1901B”, “ZI1902B” et “ZI1901B”
BDD2	“Air temperature”, “Process temperature”, “Rotational speed”, “Torque”, “Tool wear”, “H”, “M” et “L”

### 3.4.3 Normalisation des Données

Les capteurs utilisés mesurent des caractéristiques physiques variées, qui peuvent avoir des unités et des plages de valeurs différentes. Pour faciliter la comparaison et l’analyse des caractéristiques des capteurs, il est nécessaire de normaliser les données. Dans notre cas, nous avons utilisé la méthode de normalisation `MinMaxScaler`. Cette méthode consiste à transformer les valeurs des capteurs de manière à ce qu’elles se situent toutes dans une plage entre 0 et 1. Pour cela, le code (3.3) ci-dessous a été utilisé sur les deux bases de données.

```
#Importer la bibliothèque nécessaire
from sklearn.preprocessing import MinMaxScaler

# Créer une instance de MinMaxScaler
scaler = MinMaxScaler()

# Adapter et transformer les données
normalized_data = scaler.fit_transform(data)
```

FIG. 3.3 : Code de normalisation des données

## 3.5 Division des Données

Cette étape consiste à diviser l’ensemble de données en deux ou plusieurs sous-ensembles, dans le but de l’entraînement et le test d’un modèle. Dans notre cas, nous avons utilisé la méthode de 80/20, qui est une technique couramment utilisée dans l’apprentissage automatique pour diviser un ensemble de données en deux parties distinctes : un ensemble d’apprentissage (80%) et un ensemble de test (20%).

L’ensemble d’apprentissage est utilisé pour entraîner le modèle, c’est-à-dire pour lui permettre d’apprendre à partir des exemples fournis. Il est généralement plus grand, car il doit contenir suffisamment de données pour permettre au modèle de capturer les schémas et les tendances dans les données.

L’ensemble de tests est utilisé pour évaluer les performances du modèle formé sur l’ensemble d’apprentissage. Il fournit une estimation indépendante de la capacité du modèle à généraliser et à faire des prédictions précises sur de nouvelles données non vues lors de l’apprentissage.

L’instruction utilisée pour cette tâche est représentée dans la figure (3.4).

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

FIG. 3.4 : Code de division des données

## 3.6 Conception et Implementation

Cette section examine en détail trois aspects. Tout d'abord, elle se penche sur la construction des modèles, où l'objectif est de concevoir des modèles efficaces. Ensuite, elle aborde les hyperparamètres, qui jouent un rôle essentiel dans le processus de formation du modèle, en mettant l'accent sur la recherche d'ajustements optimaux. Enfin, cette section traite des métriques d'évaluation utilisées pour évaluer la performance et l'efficacité des modèles développés.

### 3.6.1 Construction des Modèles :

Une fois que les données ont été préparées dans le format approprié, il est possible de construire les différents modèles.

- **LSTM :**

Pour créer un modèle LSTM, on utilise un modèle séquentiel comme base. On ajoute ensuite une couche LSTM en précisant le nombre d'unités LSTM et la forme des données d'entrée. Pour améliorer la généralisation et éviter le surapprentissage, des couches Dropout peuvent être ajoutées. Ensuite, des couches Dense sont ajoutées au modèle selon les besoins. Une fois le modèle est construit, on le compile en spécifiant la fonction de perte, l'optimiseur et les métriques d'évaluation. Ensuite, on procède à l'entraînement du modèle en utilisant la méthode "fit()" avec les données d'entraînement, en spécifiant le nombre d'époques et la taille des lots. Une fois l'entraînement terminé, le modèle peut être utilisé pour effectuer des prédictions sur de nouvelles données.

```
#Créer le modèle LSTM
model_lstm = Sequential()
model_lstm.add(LSTM(units, input_shape=(X_train.shape[1], 1))) #units= nmb d'unités dans la couche LSTM
model_lstm.add(Dropout(dropout_rate)) #dropout_rate= taux de dropout
model_lstm.add(Dense(units, activation='ACTIVATION')) #units= nmb d'unités dans la couche Dense
#ACTIVATION= fonction d'activation

#compiler le modèle
model_lstm.compile(loss='binary_crossentropy', optimizer='OPTIMIZER',
                  metrics=['accuracy']) #OPTIMIZER= optimiseur

#Entraîner le modèle
model_lstm.fit(X_train, y_train, epochs=['EPOCHS'], #EPOCHS= nmb d'époques d'entraînement
              batch_size=['BATCH_SIZE'], verbose=0) #BATCH_SIZE= taille des lots
```

FIG. 3.5 : Code de LSTM

- **RNN :**

Un modèle RNN est construit en utilisant un modèle séquentiel. On ajoute une couche RNN en spécifiant le nombre d'unités RNN et la forme des données d'entrée. Une couche Dropout est intégrée pour améliorer la généralisation du modèle et réduit le risque de surapprentissage. Ensuite, une couche Dense est ajoutée pour générer la sortie finale en spécifiant le nombre d'unités et éventuellement la fonction d'activation. Le modèle est

compilé en définissant la fonction de perte, l'optimiseur et les métriques d'évaluation. L'entraînement du modèle est effectué en utilisant la méthode "fit()" avec les données d'entraînement, en spécifiant le nombre d'époques et la taille des lots. Une fois l'entraînement terminé, le modèle peut être utilisé pour effectuer des prédictions sur de nouvelles données.

```
#Créer le modèle RNN
model_rnn = Sequential()
model_rnn.add(SimpleRNN(units, input_shape=(X_train.shape[1], 1))) #units= nmb d'unités dans la couche RNN
model_rnn.add(Dropout(dropout_rate)) #dropout_rate= taux de dropout
model_rnn.add(Dense(units, activation='ACTIVATION')) #units= nmb d'unités dans la couche Dense
#ACTIVATION= fonction d'activation

#compiler le modèle
model_rnn.compile(loss='binary_crossentropy', optimizer='OPTIMIZER',
                  metrics=['accuracy']) #OPTIMIZER= optimiseur

#Entraîner le modèle
model_rnn.fit(X_train, y_train, epochs=['EPOCHS'], #EPOCHS= nmb d'époques d'entraînement
             batch_size=['BATCH_SIZE'], verbose=0) #BATCH_SIZE= taille des lots
```

FIG. 3.6 : Code de RNN

- **CNN :**

Pour créer un modèle CNN, on instancie un modèle séquentiel auquel on ajoute des couches de convolution pour extraire les caractéristiques visuelles des données d'entrée. Les couches de convolution sont suivies de couches de pooling pour réduire la dimensionnalité des caractéristiques extraites. Ensuite, on peut ajouter des couches Dense pour combiner ces caractéristiques et prendre des décisions basées sur ces caractéristiques. Une fois le modèle construit, on le compile en spécifiant la fonction de perte, l'optimiseur et les métriques d'évaluation. Ensuite, on entraîne le modèle en utilisant la méthode "fit()" avec les données d'entraînement, en précisant le nombre d'époques et la taille des lots. Pendant l'entraînement, les poids des différentes couches sont ajustés pour minimiser la perte. Une fois l'entraînement terminé, on peut utiliser le modèle pour effectuer des prédictions sur de nouvelles données.

```
#Créer le modèle CNN
model_cnn = Sequential()
model_cnn.add(Conv1D(filters=FILTERS, kernel_size=KERNEL_SIZE, #FILTERS= nmb de filtres de convolution
                    #KERNEL_SIZE= Taille e noyau
                    activation='ACTIVATION', input_shape=(X_train.shape[1], 1)))
model_cnn.add(MaxPooling1D(pool_size=POOL_SIZE)) #POOL_SIZE= Taille de fenetres de pooling
model_cnn.add(Flatten())
model_cnn.add(Dense(units, activation='ACTIVATION')) #units= nmb d'unités dans la couche Dense
model_cnn.add(Dropout(dropout_rate)) #dropout_rate= taux de dropout
model_cnn.add(Dense(1, activation='ACTIVATION')) #ACTIVATION= fonction d'activation

#compiler le modèle
model_cnn.compile(loss='binary_crossentropy', optimizer='OPTIMIZER',
                  metrics=['accuracy']) #OPTIMIZER= optimiseur

#Entraîner le modèle
model_cnn.fit(X_train, y_train, epochs=['EPOCHS'], #EPOCHS= nmb d'époques d'entraînement
             batch_size=['BATCH_SIZE'], verbose=0) #BATCH_SIZE= taille des lots
```

FIG. 3.7 : Code de CNN

### 3.6.2 Les Hyperparamètres :

Afin de renforcer les performances globales du modèle et d'augmenter la précision des prédictions tout en réduisant le risque de surapprentissage, une optimisation des hyperparamètres

a été réalisée pour chaque algorithme, en prenant en compte chaque base de données spécifique. Les détails du code utilisé sont présentés en annexe (A.3), et le tableau (3.4) résume les différentes combinaisons d’hyperparamètres obtenues.

TAB. 3.4 : Les hyperparamètres optimaux des algorithmes pour chaque BDD

Base de Données	modèles	Hyperparamètres
BDD1	LSTM	“optimizer=adam”, “hidden-layer-sizes=100”, “epochs=150”, “dropout-rate=0.4”, “batch-size=32”, “activation=sigmoid”, “loss=binary-crossentropy”.
	RNN	“optimizer=adam”, “hidden-layer-sizes=110”, “epochs=50”, “dropout-rate=0.2”, “batch-size=32”, “activation=sigmoid”, “loss=binary-crossentropy”.
	CNN	“optimizer=adam”, “pool-size=3”, “kernel-size=3”, “hidden-units=128”, “filters=128”, “epochs=100”, “dropout-rate=0.3”, “batch-size=64”, “activation=sigmoid”, “loss=binary-crossentropy”.
BDD2-cas1	LSTM	“optimizer=adam”, “hidden-layer-sizes=128”, “epochs=100”, “dropout-rate=0.2”, “batch-size=”, “activation=sigmoid”, “loss=binary-crossentropy”.
	RNN	“optimizer=adam”, “hidden-layer-sizes=128”, “epochs=50”, “dropout-rate=0.3”, “batch-size=32”, “activation=sigmoid”, “loss=binary-crossentropy”.
	CNN	“optimizer=adam”, “pool-size=2”, “kernel-size=5”, “hidden-units=128”, “filters=128”, “epochs=100”, “dropout-rate=0.5”, “batch-size=32”, “activation=sigmoid”, “loss=binary-crossentropy”.
BDD2-cas2	LSTM	“optimizer=adam”, “hidden-layer-sizes=110”, “epochs=110”, “dropout-rate=0.2”, “batch-size=32”, “activation=softmax”, “loss=categorical-crossentropy”.
	RNN	“optimizer=adam”, “hidden-layer-sizes=50”, “epochs=50”, “dropout-rate=0.2”, “batch-size=32”, “activation=softmax”, “loss=categorical-crossentropy”.
	CNN	“optimizer=adam”, “pool-size=2”, “kernel-size=3”, “hidden-units=64”, “filters=64”, “epochs=50”, “dropout-rate=0.5”, “batch-size=32”, “activation=softmax”, “loss=categorical-crossentropy”.

### 3.6.3 Les Métriques d’Évaluation :

Lors de l’évaluation des algorithmes de classification, plusieurs métriques peuvent être utilisées pour mesurer leurs performances. Voici quelques-unes des métriques d’évaluation couramment utilisées :



- **La Matrice de Confusion (CM)**

CM est une représentation tabulaire de taille (n×n) utilisée pour visualiser les résultats des modèles prédictifs dans les problèmes de classification, où "n" est le nombre de classes dans l'ensemble de données (voir la figure 3.8). Cette matrice croise les classes cibles réelles avec les classes prédites obtenues. Ceci offre également le nombre d'instances correctement classées et mal classées [59].

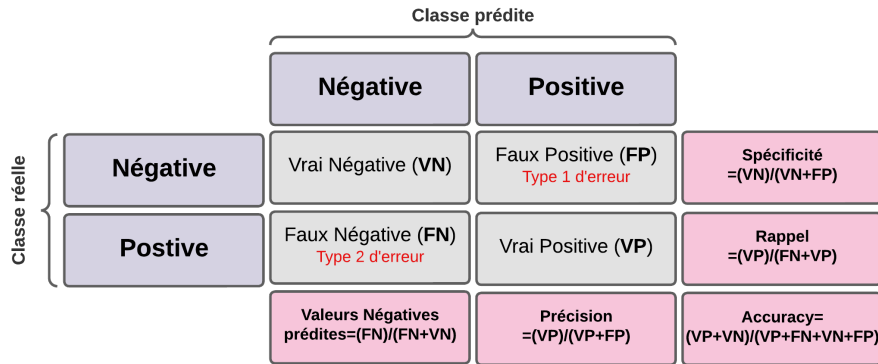


FIG. 3.8 : Matrice de confusion

- **VP** : Vrais Positifs est le nombre d'instances positives correctement classifiées.
- **FP** : Faux Positifs est le nombre d'instances négatives et qui sont prédites comme positives.
- **FN** : Faux Négatifs est le nombre d'instances positives classifiées comme négatives
- **VN** : Vrais Négatifs est le nombre d'instances négatives correctement classifiées.

Dans ce qui suit, nous allons expliquer plusieurs métriques qui peuvent être calculées à partir de la matrice de confusion.

- **L'Accuracy**

L'accuracy (taux de succès) est une mesure qui représente la proportion d'instances correctement classifiées par rapport au total des instances [59].

$$Accuracy = \frac{VP + VN}{VP + VN + FP + FN} \tag{3.1}$$

- **La Précision**

La précision (precision) est une mesure qui représente le pourcentage de prédictions positives qui sont correctes parmi toutes les prédictions positives effectuées par un modèle [59].

$$Precision = \frac{VP}{VP + FP} \tag{3.2}$$

- **Le Rappel**

Le rappel représente la capacité du modèle à prédire correctement les points positifs à partir des points positifs réels qui existent dans un ensemble de données [59].

$$Rappel = \frac{VP}{VP + FN} \quad (3.3)$$

- **Le Score F1**

Le score F1 est une métrique qui combine la sensibilité et la précision en utilisant la moyenne harmonique pour fournir une mesure unique de la performance du modèle.

$$F1Score = \frac{2 \times (Recall \times Precision)}{Recall + Precision} \quad (3.4)$$

- **La courbe Précision-Rappel**

La courbe précision-rappel (Precision-Recall Curve) est une courbe qui représente l'évolution de la précision en fonction du rappel (sensibilité) à mesure que le seuil de décision varie. Pour résumer cette courbe, on utilise souvent l'aire située en dessous de celle-ci, ce qui est abrégé par le terme AUPR (Area Under the Precision-Recall Curve).

Lorsque le seuil de décision est très élevé, aucun exemple n'est étiqueté comme positif, ce qui signifie que la précision n'est pas définie. Pour remédier à cela, il est courant de définir une précision de 1 lorsque la première observation considérée est positive, et une précision de 0 dans le cas contraire. Cette convention permet d'inclure ces cas extrêmes dans le calcul de l'AUPR [59].

- **La courbe ROC**

La courbe ROC est obtenue en croisant le taux de vrais positifs (TVP) avec le taux de faux négatifs (TFN) pour différents seuils de classification. Pour établir une référence, on utilise un classificateur aléatoire qui agit comme une ligne de base. Lorsque les instances sont classées de manière aléatoire, la courbe ROC devrait correspondre à cette ligne. Un modèle qui présente une discrimination parfaite (sans chevauchement entre les classes), sera représenté par une courbe ROC qui passe par le coin supérieur gauche du graphique. En d'autres termes, lorsque la courbe ROC est plus proche du coin supérieur, cela indique une meilleure performance de classification [59].

## 3.7 Résultats et Discussion

### 1. Base de Données 1

Les résultats d'entraînement et du test obtenus à partir des différents modèles sont présentés dans ce qui suit.

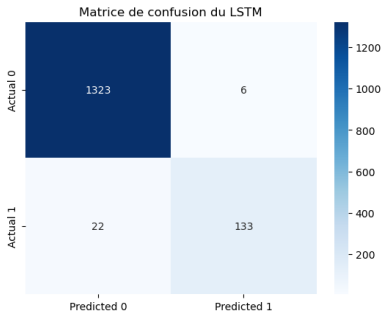


FIG. 3.9 : Matrice de confusion du LSTM-BDD1

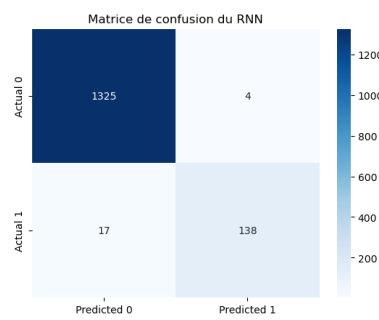


FIG. 3.10 : Matrice de confusion du RNN-BDD1

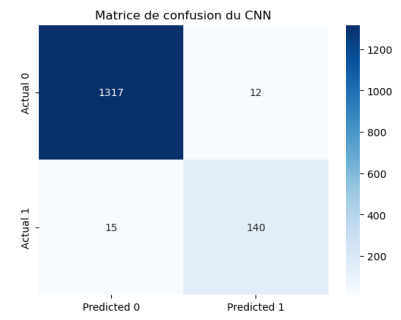


FIG. 3.11 : Matrice de confusion du CNN-BDD1

En analysant les trois matrices de confusion (3.9),(3.10),(3.11) correspondant à nos modèles, nous pouvons observer les performances de chaque modèle en termes de prédictions correctes et d’erreurs de classification pour chaque classe. Ces matrices fournissent une vue détaillée des résultats obtenus, permettant de comparer les performances de chaque modèle en identifiant les classes pour lesquelles ils sont plus ou moins précis.

Le tableau (3.5) présente les résultats des métriques de performance pour les trois modèles appliqués à la base de données BDD1. Les métriques incluent l’accuracy, la précision, le rappel et le F1 score pour chaque classe (classe 0 et classe 1).

TAB. 3.5 : Les métriques des différentes modèles BDD1

Modèle	Classe	Accuracy	Précision	Rappel	F1 score
LSTM	0	0.9811	0.98	1	0.99
	1		0.96	0.86	0.90
RNN	0	0.9858	0.99	1	0.99
	1		0.97	0.89	0.93
CNN	0	0.9818	0.99	0.99	0.99
	1		0.92	0.90	0.91

La figure (3.15) et le tableau (3.5) indiquent que les modèles présentent une très bonne performance en termes d’accuracy supérieure à (98%). Pour la classe 0, tous les modèles ont obtenu des performances élevées en termes, de précision, de rappel et de F1 score, avec des valeurs supérieures à 0.98. Cela suggère que les modèles sont capables de bien classifier les échantillons de cette classe.

En ce qui concerne la classe 1, les modèles RNN et CNN ont obtenu des performances relativement élevées, avec des valeurs, de précision, de rappel et de F1 score supérieur à 0.90. Cependant, le modèle LSTM a obtenu des performances légèrement inférieures, en particulier en termes de rappel et de F1 score.

En résumé, les trois modèles ont montré de bonnes performances globales pour la base de données BDD1, mais le modèle RNN semble avoir des performances légèrement supérieures pour la classe 1, avec des valeurs de Précision, Rappel et de F1 score plus élevées. Cependant, tous les modèles ont démontré leur capacité à classifier avec précision les échantillons des deux classes.

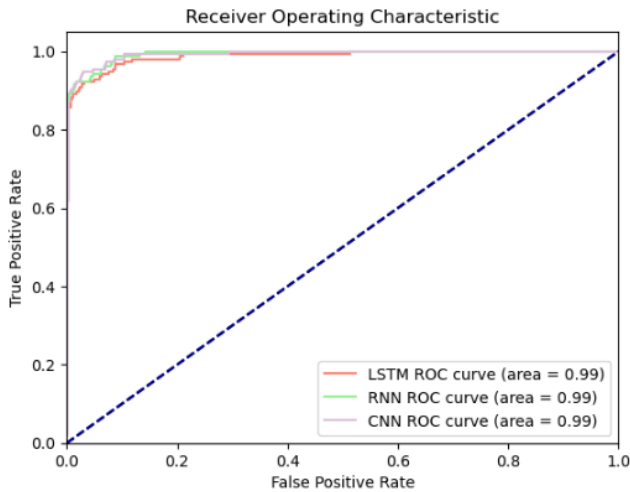


FIG. 3.12 : Courbes ROC des différents modèles-BDD1

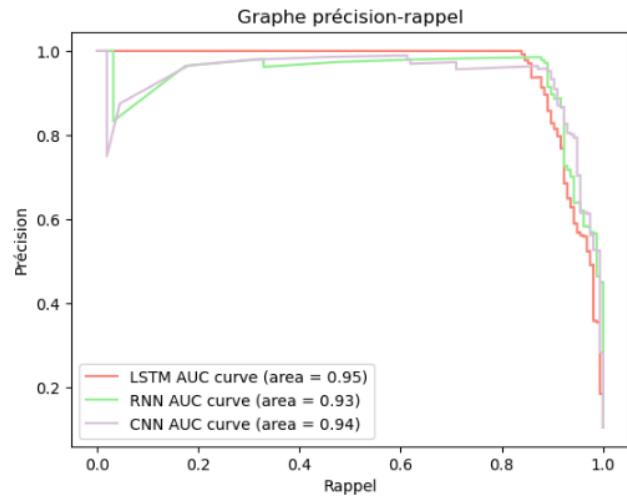


FIG. 3.13 : Courbes Précision-Rappel des différents modèles-BDD1

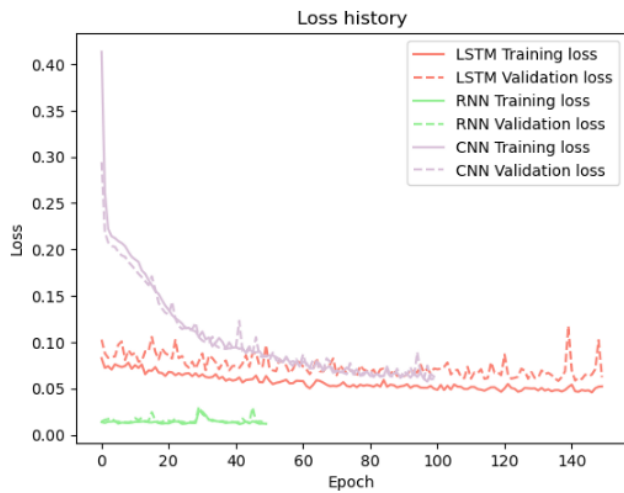


FIG. 3.14 : Courbes LOSS des différents modèles-BDD1

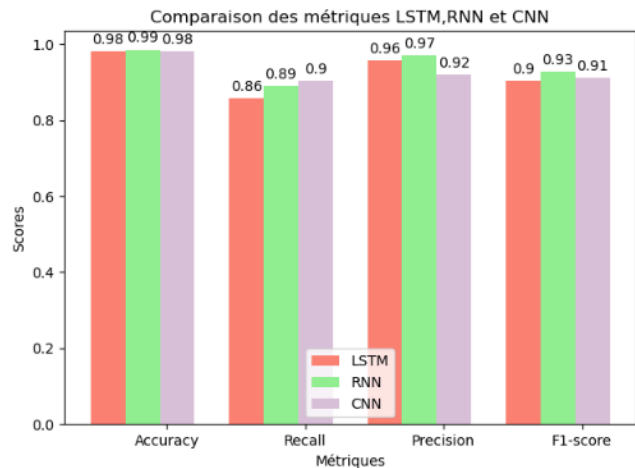


FIG. 3.15 : Les métriques des différents modèles-BDD1

En examinant les courbes ROC (3.12), on peut observer que les trois modèles se situent près du coin supérieur gauche. Cette position indique un meilleur compromis entre le taux de vrais positifs (TVP) et le taux de faux positifs (TFP), ce qui traduit une performance améliorée des modèles.

En analysant les courbes Précision-Rappel(3.13), nous constatons que les valeurs de l'aire sous la courbe (AUC) pour les trois modèles sont relativement proches, bien que le modèle LSTM obtienne la valeur la plus élevée. Cela suggère que les trois modèles sont capables de discriminer efficacement entre les différentes classes et de fournir des prédictions précises.

En se basant sur l'observation de les courbes LOSS (3.14), nous remarquons une évolution positive des pertes d'entraînement et de validation pour les modèles LSTM, CNN et RNN. Ces modèles affichent une bonne performance et convergent vers des valeurs de pertes basses et constantes. Ces résultats confirment leur capacité à apprendre efficacement et à s'adapter aux données pour effectuer des prédictions précises.

## 2. Base de Données 2

La deuxième base de données se caractérise par deux cibles distinctes. L'une décrit l'état de l'équipement (Panne ou Non Panne), tandis que l'autre concerne les modes de défaillance (Heat Dissipation Failure, Power Failure, Overstrain Failure, Tool Wear Failure, Random Failures ou No Failure). Pour cela, les algorithmes sélectionnés seront mis en œuvre avec cette base de données pour les deux scénarios possibles.

- **Cas 01: Classification Binaire**

Les figures (3.16), (3.17) et (3.18) illustrent les matrices de confusion associées à nos modèles. Elles révèlent les prédictions correctes ainsi que les erreurs de classification pour la classe «0=Non Panne» et la classe «1=Panne».

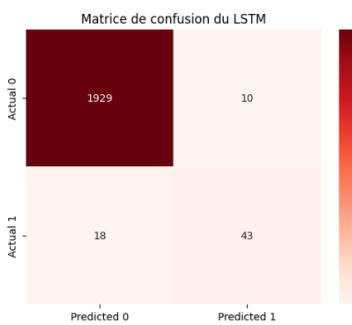


FIG. 3.16 : Matrice de confusion du LSTM-BDD2-1

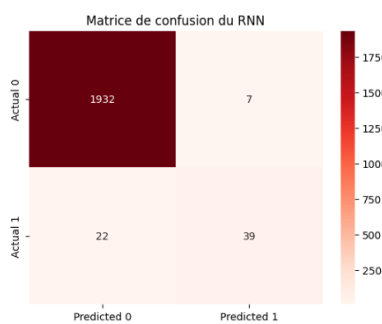


FIG. 3.17 : Matrice de confusion du RNN-BDD2-1

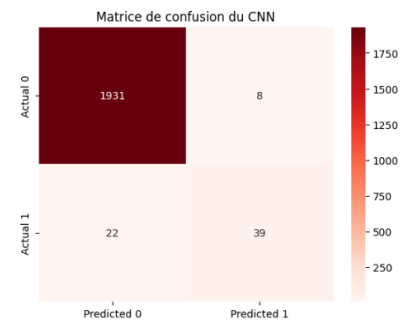


FIG. 3.18 : Matrice de confusion du CNN-BDD2-1

À partir de ces matrices, nous pouvons déduire les différentes mesures de performance pour la base de données 2, associées à chaque classe pour chaque modèle. Ces mesures sont présentées dans le tableau (3.6) suivant :

TAB. 3.6 : Les métriques des différents modèles BDD2-cas1

Modèle	Classe	Accuracy	Précision	Rappel	F1 score
LSTM	0	0.9860	0.99	0.99	0.99
	1		0.81	0.70	0.75
RNN	0	0.9855	0.99	1	0.99
	1		0.85	0.64	0.73
CNN	0	0.9850	0.99	1	0.99
	1		0.83	0.64	0.72

Selon la figure (3.22) et le tableau (3.6), les modèles ont une excellente performance en termes d'accuracy (plus de 98 %). Pour la classe 0, tous les modèles affichent des performances exceptionnelles en matière de précision, de rappel et de score F1, avec des valeurs supérieures à 0,99. Cela confirme que les modèles sont capables de bien classer les échantillons de cette classe.

En ce qui concerne la classe 1, le modèle LSTM obtient des performances relativement élevées en termes de précision, de rappel et de score F1, supérieures à 0,70.

Cependant, les modèles RNN et CNN ont des performances légèrement inférieures, en particulier en termes de rappel et de F1 score.

En résumé, les trois modèles ont globalement de bonnes performances pour la base de données BDD2, mais le modèle LSTM semble avoir des performances légèrement supérieures pour la classe 1, avec des valeurs de rappel et de score F1 plus élevées. Cependant, tous les modèles ont démontré leur capacité à classifier avec précision les échantillons des deux classes.

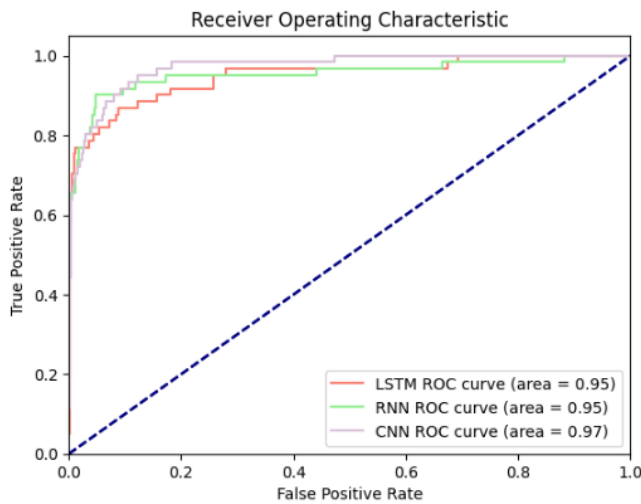


FIG. 3.19 : Courbes ROC des différents modèles-BDD2

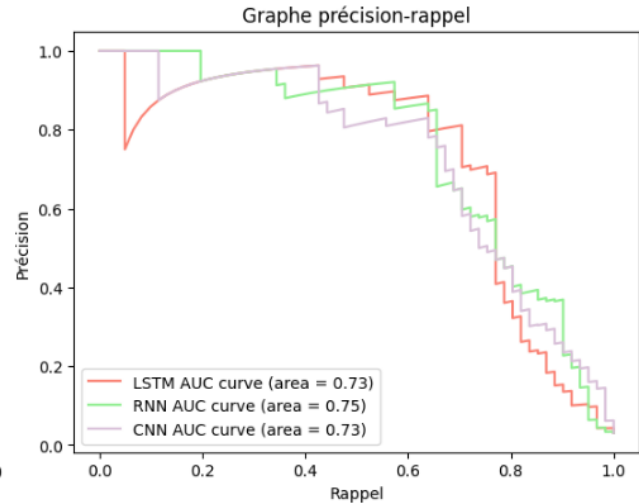


FIG. 3.20 : Courbes Précision-Rappel des différents modèles-BDD2

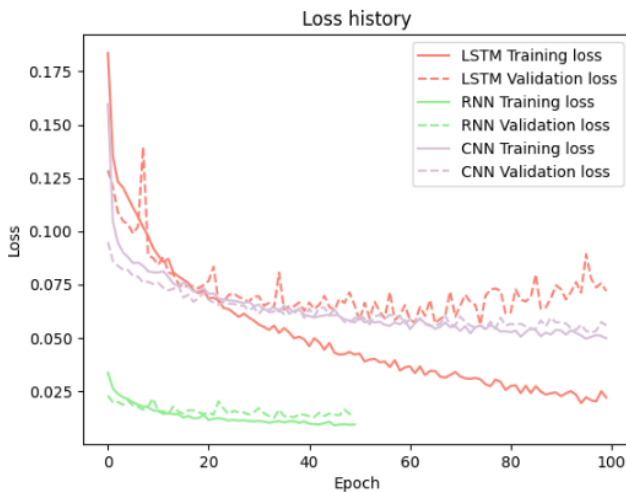


FIG. 3.21 : Courbes LOSS des différents modèles-BDD2

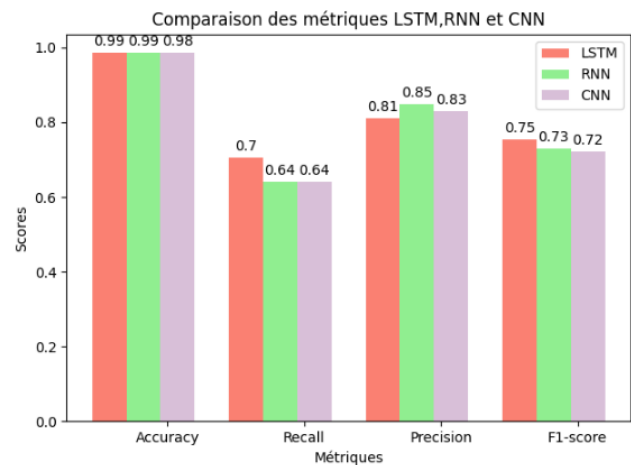


FIG. 3.22 : Les métriques des différents modèles-BDD2

Les courbes ROC représentées dans la figure (3.19), se rapprochent du coin supérieur gauche du graphique, indiquant une sensibilité et une spécificité élevée. De plus, l'aire sous la courbe ROC (AUC-ROC) est supérieure à 0.95 pour les différents modèles signifie que le modèle a une forte capacité à distinguer entre les classes positives et négatives. Il est capable de classifier correctement 95% des paires d'observations positives et négatives.

Les courbes Précision-Rappel (3.20), indiquent que les valeurs de l'aire sous la courbe

(AUC) pour les trois modèles sont relativement proches. Cela signifie que les trois modèles sont capables d'identifier correctement un grand nombre d'échantillons positifs tout en maintenant un faible nombre de faux positifs.

Selon les courbes LOSS (3.21), on remarque que les pertes des modèles diminuent à la fois sur les données d'entraînement et de validation, cela peut indiquer que les modèles sont en train de trouver un bon équilibre entre l'ajustement aux données d'entraînement et la généralisation aux données de validation. Par ailleurs, la courbe de la perte de validation du modèle LSTM est considérablement plus élevée que celle d'entraînement, cela peut indiquer un sous-apprentissage du modèle.

- **Cas 02: Classification Multi-Classes**

Les figures (3.23), (3.24) et (3.25) illustrent les matrices de confusion associées à nos modèles. Elles révèlent les prédictions correctes ainsi que les erreurs de classification pour chaque classe.

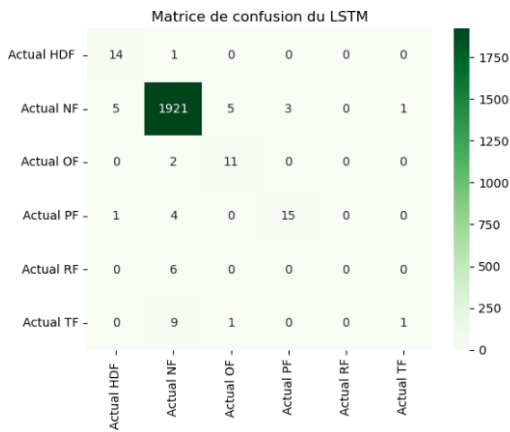


FIG. 3.23 : Matrice de confusion du LSTM-BDD2-2

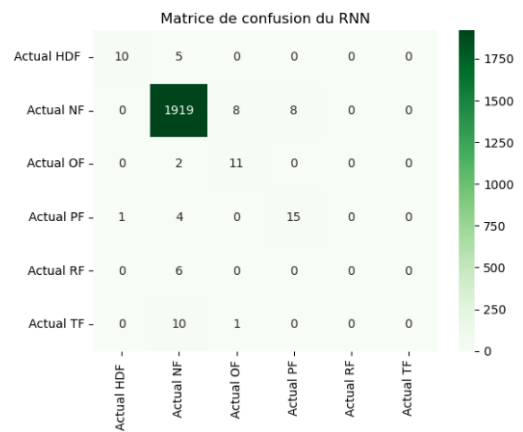


FIG. 3.24 : Matrice de confusion du RNN-BDD2-2

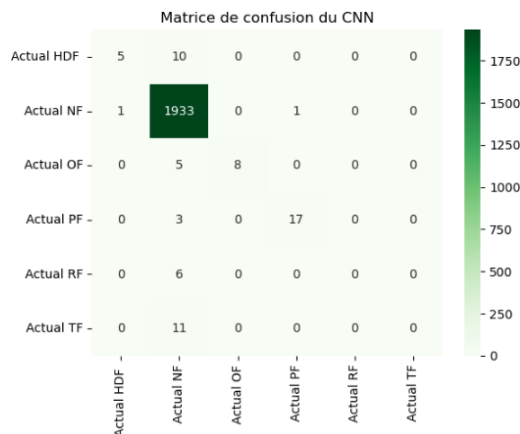


FIG. 3.25 : Matrice de confusion du CNN-BDD2-2

Les résultats des différentes mesures de performance associées a ce cas sont présentés dans le tableau suivant (Tableau 3.7) :

TAB. 3.7 : Les métriques des différents modèles BDD2-cas2

Modèle	Classe	Accuracy	Précision	Rappel	F1 score
LSTM	Heat Dissipation Failure	0.9810	0.70	0.93	0.80
	No Failure		0.99	0.99	0.99
	Overstrain Failure		0.65	0.85	0.73
	Power Failure		0.83	0.75	0.79
	Random Failures		0	0	0
	Tool Wear Failure		0.5	0.09	0.15
RNN	Heat Dissipation Failure	0.9775	0.91	0.67	0.77
	No Failure		0.99	0.99	0.99
	Overstrain Failure		0.55	0.85	0.67
	Power Failure		0.65	0.75	0.75
	Random Failures		0	0	0
	Tool Wear Failure		0	0	0
CNN	Heat Dissipation Failure	0.9815	0.83	0.33	0.48
	No Failure		0.98	1	0.99
	Overstrain Failure		1	0.62	0.76
	Power Failure		0.94	0.85	0.89
	Random Failures		0	0	0
	Tool Wear Failure		0	0	0

Les résultats présentés dans les matrices (3.23), (3.24), (3.25) et dans le tableau (3.7) ci-dessus, ont permis de déterminer que le modèle LSTM se démarque avec des performances élevées par rapport aux deux autres modèles. En effet, le modèle LSTM a réussi à prédire presque toutes les classes avec un rappel supérieur à 0.8 et une précision moyenne pour la plupart des classes, à l'exception d'une classe qu'il n'a pas réussi à la prédire. Il est important de noter que le nombre d'échantillons pour cette classe était considérablement plus petit par rapport aux autres, ce qui a eu pour conséquence que tous les modèles ont eu du mal à s'entraîner correctement sur cette classe en particulier.

### Synthèse

D'après les résultats des différentes implémentations des algorithmes avec chaque base de données, nous pouvons conclure que pour la première base de données, caractérisée par un nombre restreint de features, le modèle RNN affiche des performances légèrement supérieures. En revanche, pour la deuxième base de données, où la quantité de données est supérieure à celle de la première, le modèle LSTM s'avère être le meilleur classificateur parmi les trois algorithmes considérés pour les deux cas de classification.

Les résultats obtenus mettent en évidence qu'il n'existe pas d'algorithme clairement supérieur aux autres. La sélection de l'algorithme le plus approprié ne se base pas uniquement sur les critères que nous avons utilisés dans notre étude comparative, mais également sur d'autres critères tels que la capacité de l'algorithme à traiter efficacement de grandes quantités de données et à résoudre des problèmes plus complexes, ainsi que l'objectif de la PdM. Il convient de noter que les résultats de notre étude sont fondés sur les métriques



employées ainsi que sur les caractéristiques spécifiques des bases de données évaluées. Par conséquent, il est envisageable que d'autres algorithmes ou approches puissent produire de meilleurs résultats pour d'autres tâches ou critères d'évaluation.

### 3.8 Comparaison avec des Études Similaires

Dans cette partie du travail, nous présenterons des études récentes sur l'ensemble de données BDD2 qui ont utilisé différentes approches pour résoudre le même problème de détection de panne, ces travaux sont : N.Sharma et al. (2022) [60] en utilisant (Random Forest) et I.Assagaf et al. (2023) [61] en utilisant (Support Vector Machines).

Comme le montre le tableau (3.8) ci-dessous, le modèle LSTM surpasse les modèles construits dans les études citées précédemment en terme d'accuracy.

TAB. 3.8 : Comparaison avec les résultats d'études similaires

Travail	Notre étude	N.Sharma et al. [60]	I.Assagaf et al. [61]
Approche	LSTM	Random Forest	Support Vector Machine
Accuracy	98.6%	98.4%	93%

Les résultats du tableau démontrent l'efficacité et les performances supérieures des techniques du Deep Learning par rapport aux techniques du Machine Learning traditionnelles. Cela s'explique par la capacité des techniques de DL à capturer des motifs complexes dans les données. Toutefois, il est crucial de prendre en compte d'autres mesures d'évaluation et de considérer le contexte spécifique de la tâche ainsi que les éventuelles limitations des données ou des paramètres expérimentaux. Bien que le modèle LSTM se démarque en termes de performance, il est important de prendre en compte les exigences et les considérations spécifiques à la tâche, ce qui pourrait justifier l'utilisation d'autres modèles tels que Random Forest ou SVM.

### 3.9 Conclusion

Ce chapitre a été consacré à l'implémentation des algorithmes sélectionnés dans notre étude comparative réalisée dans le chapitre précédent, ainsi qu'à la discussion des résultats obtenus afin de déterminer le meilleur algorithme DL pour la PdM. Nous avons évalué ces algorithmes sur plusieurs bases de données, en tenant compte de plusieurs métriques.

Les résultats de cette étude ont démontré qu'aucun modèle d'apprentissage ne peut être considéré comme étant meilleur à tous les autres. Dans le domaine de Deep Learning pour la maintenance prédictive (PdM), le choix de l'algorithme dépend de la taille et de la complexité de la base de données ainsi que du type de problème. Pour les bases de données volumineuses et complexes, le modèle LSTM s'est révélé être la meilleure option, que ce soit pour des problèmes de classification binaire ou multi-classes. En revanche, pour les bases de données plus simples, le modèle RNN peut être suffisant et approprié.

# Conclusion et Perspectives

Ce projet de recherche s'est focalisé sur l'utilisation de l'intelligence artificielle dans le domaine de la Maintenance Prédictive. L'objectif était de sélectionner l'algorithme de Deep Learning le plus approprié aux problèmes de la Maintenance Prédictive. Pour atteindre cet objectif, une revue de l'état de l'art a été réalisée dans un premier temps, afin d'examiner les différents types de maintenance, ainsi que les avantages et les inconvénients de la Maintenance Prédictive et ses approches. De plus, une exploration approfondie du concept du Deep Learning et de ses techniques a été entreprise. Enfin, une étude comparative des algorithmes du DL a été effectuée à l'aide de critères couramment utilisés dans les articles de recherche, dans le but d'identifier les algorithmes les plus efficaces pour résoudre les tâches de la PdM.

Le travail réalisé nous a permis de consulter de nombreux articles de recherche portant sur le domaine de Deep Learning, un sujet extrêmement en vogue ces dernières années, avec une application majeure dans l'industrie. Parallèlement, la Maintenance Prédictive est devenue le sujet de recherche le plus courant dans le domaine de la maintenance industrielle. Les nouvelles technologies de PdM disponibles sur le marché renforcent davantage sa position par rapport aux approches de maintenance traditionnelles.

Par conséquent, Le Deep Learning est une approche qui vise à améliorer la Maintenance Prédictive en facilitant les tâches et les calculs de prédictions. Cependant, il présente ses propres limites, notamment en ce qui concerne la disponibilité de données adéquates et représentatives pour l'entraînement, ainsi que les coûts élevés associés aux ressources computationnelles nécessaires et à la collecte/annotation des données requises.

En effectuant des implémentations avec des données d'entraînement réelles et de simulation, il a été constaté que le choix de l'algorithme le plus approprié dépend de la taille et de la complexité de la base de données ainsi que le type de problème (binaire/multi-classes). Dans le cas de bases de données volumineuses et complexes, le modèle LSTM s'avère plus adapté pour les deux types de problèmes, tandis que pour des bases de données plus simples, le modèle RNN peut suffire.

Bien que les modèles sélectionnés et implémentés aient donné de bons résultats, il est important de noter que les critères utilisés pour choisir l'algorithme le plus approprié sont indépendants de la PdM. Il est donc nécessaire de prendre en considération d'autres critères afin d'obtenir des résultats plus solides et plus en accord avec les objectifs de l'entreprise en matière de PdM. De plus, cette étude réside dans le fait qu'elle a uniquement pris en compte un type de problème, à savoir la classification. Par conséquent, il est difficile de généraliser les résultats obtenus.

### Perspectives

Notre projet présente plusieurs possibilités d'évolution, parmi lesquelles figurent les suivantes :

- Le développement d'un algorithme de Deep Learning dédié à la Maintenance Prédictive, en utilisant les principes d'algorithmes préalablement sélectionnés. Par exemple, on pourrait envisager un algorithme hybride capable de gérer à la fois la classification binaire et multi-classes. Cet algorithme serait capable de tirer parti des performances des RNN sur des ensembles de données de petite taille, tout en utilisant des LSTM sur des ensembles de données plus volumineux, afin de générer des prédictions fiables.
- Étudier l'utilisation du Deep Learning pour résoudre d'autres problèmes liés à la Maintenance Prédictive (PdM). Par exemple, il serait possible de développer un modèle de régression basé sur le Deep Learning afin de prédire des mesures quantitatives pertinentes dans le contexte de la PdM. Ces mesures pourraient inclure la durée de vie restante d'un équipement, le temps estimé jusqu'à la prochaine défaillance ou encore les coûts de maintenance anticipés. En obtenant des prédictions précises, ce modèle permettrait aux gestionnaires de maintenance de planifier de manière proactive les activités de maintenance, d'optimiser l'utilisation des ressources disponibles et de réduire les coûts opérationnels.

En conclusion, nous espérons que ce mémoire pourra servir de point de départ à d'autres travaux futurs portant sur l'application de Deep Learning dans le domaine de la Maintenance Prédictive.

# Bibliographie

- [1] Liang GUO et al. “A recurrent neural network based health indicator for remaining useful life prediction of bearings”. In : *Neurocomputing* 240 (2017), p. 98-109.
- [2] Haiyue WU, Aihua HUANG et John W SUTHERLAND. “Avoiding environmental consequences of equipment failure via an LSTM-based model for predictive maintenance”. In : *Procedia Manufacturing* 43 (2020), p. 666-673.
- [3] Woo-Seung YUN, Won-Hwa HONG et Hyuncheol SEO. “A data-driven fault detection and diagnosis scheme for air handling units in building HVAC systems considering undefined states”. In : *Journal of Building Engineering* 35 (2021), p. 102111.
- [4] Abdul WAHID, John G BRESLIN et Muhammad Ali INTIZAR. “Prediction of Machine Failure in Industry 4.0: A Hybrid CNN-LSTM Framework”. In : *Applied Sciences* 12.9 (2022), p. 4221.
- [5] MADANI et AL. “Mise en place d’une maintenance prédictive pour un système industriel de production.” Thèse de doct. Université Ibn Khaldoun-Tiaret-, 2021.
- [6] X FD. “60-000, Norme Française”. In : *Fonction Maintenance. AFNOR* (2002).
- [7] Abdelfettah OUADAH, Leila ZEMMOUCHI-GHOMARI et Nedjma SALHI. “Selecting an appropriate supervised machine learning algorithm for predictive maintenance”. In : *The International Journal of Advanced Manufacturing Technology* (2022).
- [8] HASSEN MAGHLOUT. “Application de l’analyse en ondelettes discrète dans la maintenance prédictive des paliers à roulements”. In : (2010).
- [9] Fatoumata et AL. “Machine Learning Pour La Maintenance Prédictive.” In : (2021).
- [10] Rachid NOUREDDINE. “Implémentation de la maintenance prédictive dans les systèmes de production”. In : *University of Oran 2* (2008).
- [11] Muhammad et al SOHAIB. “Deep learning for data-driven predictive maintenance”. In : *Vision, Sensing and Analytics : Integrative Approaches*. Springer, 2021, p. 71-95.
- [12] ARAB MUSTAPHA. “Contribution à l’amélioration de la maintenance prédictive dans l’industrie”. Thèse de doct. UNIVERSITÉ BADJI MOKHTAR-ANNABA, 2019.
- [13] Salah Eddine BENCHETTOUH. “Elaboration d’un système de prédiction des pannes et de planification des maintenances”. Thèse de doct. UNIVERSITE MOHAMED BOUDIAF-M’SILA-FACULTE MATHEMATIQUES ET DE L’INFORMATIQUE, 2019.
- [14] HOUD.Mohamed BACHIR et DJEBABLIA.ABDELMOTALIB. “Design and implementation of a predictive maintenance support tool based on supervised machine learning”. Thèse de doct. National Higher School of Technology *Department of Industrial Engineering Et Maintenance* 2021.

- [15] Omar DJEBILI. “Contribution à la maintenance prédictive par analyse vibratoire des composants mécaniques tournants. Application aux butées à billes soumises à la fatigue de contact de roulement.” Thèse de doct. Reims, 2013.
- [16] Manel SOUALHI et Oualid BERKANI. “Diagnostic des défaillances des roulements par analyse vibratoire”. In : (2022).
- [17] Cao et AL. “KSPMI : a knowledge-based system for predictive maintenance in industry 4.0”. In : *Robotics and Computer-Integrated Manufacturing* 74 (2022), p. 102281.
- [18] Jimenez et AL. “Towards multi-model approaches to predictive maintenance : A systematic literature survey on diagnostics and prognostics”. In : *Journal of Manufacturing Systems* 56 (2020), p. 539-557.
- [19] Es-sakali et AL. “Review of predictive maintenance algorithms applied to HVAC systems”. In : *Energy Reports* 8 (2022), p. 1003-1012.
- [20] Sayyad et AL. “Estimating remaining useful life in machines using artificial intelligence : A scoping review”. In : *Libr. Philos. Pract* 2021 (2021), p. 1-26.
- [21] Shreyas GAWDE et al. “A scoping review on multi-fault diagnosis of industrial rotating machines using multi-sensor data fusion”. In : *Artificial Intelligence Review* (2022), p. 1-54.
- [22] Kabouche ABDALLAH. “Techniques de Maintenance Prédictive pour l’Amélioration de la disponibilité des Installations”. Thèse de doct. Université de Annaba-Badji Mokhtar, 2007.
- [23] A KATONA, Peter PANFILOV et B KATALINIC. “Building predictive maintenance framework for smart environment application systems”. In : *Proceedings of the 29th DAAAM international symposium*. 2018, p. 0460-0470.
- [24] Yongyi RAN et al. “A survey of predictive maintenance : Systems, purposes and approaches”. In : *arXiv preprint arXiv :1912.07383* (2019).
- [25] Filipe ALVES et al. “Deployment of a smart and predictive maintenance system in an industrial case study”. In : *2020 IEEE 29th International Symposium on Industrial Electronics (ISIE)*. IEEE. 2020, p. 493-498.
- [26] Alberto JIMENEZ-CORTADI et al. “Predictive maintenance on the machining process and machine tool”. In : *Applied Sciences* 10.1 (2019), p. 224.
- [27] Kicky G van LEEUWEN et al. “Artificial intelligence in radiology : 100 commercially available products and their scientific evidence”. In : *European radiology* 31 (2021), p. 3797-3804.
- [29] Marizel B VILLANUEVA et Ma Louella M SALENGA. “Bitter melon crop yield prediction using machine learning algorithm”. In : *International Journal of Advanced Computer Science and Applications* 9.3 (2018).
- [30] Davy CIELEN et Arno MEYSMAN. *Introducing data science : big data, machine learning, and more, using Python tools*. Simon et Schuster, 2016.
- [31] Batta MAHESH. “Machine learning algorithms-a review”. In : *International Journal of Science and Research (IJSR).[Internet]* 9 (2020), p. 381-386.

- [32] manisha more vanita tonge buradkar manisha more. “machine learning and soft computing.introduction to machine learning and its applications : a survey”. In : *journal of artificial intelligence* (2020).
- [33] Michael PALUSZEK et Stephanie THOMAS. *MATLAB machine learning*. Apress, 2016.
- [34] BD SMART. “Reinforcement learning : A user’s guide”. In : *Tutorial at International Conference on Autonomic Computing (ICAC 2005)*. 2005.
- [35] Christophe SALPERWYCK. “Apprentissage incrémental en ligne sur flux de données”. Thèse de doct. Université Charles de Gaulle-Lille III, 2012.
- [36] Jiasi CHEN et Xukan RAN. “Deep learning with edge computing : A review”. In : *Proceedings of the IEEE* 107.8 (2019), p. 1655-1674.
- [37] Hocine TOLGUI. “Deep Learning pour Reconnaissance du Visage”. In : ().
- [39] Hiba KHELIL et Abdelkader BENYETTOU. “Application du Réseau de Neurones Gamma à la Reconnaissance de la Parole”. In : *SETIT 2007 4th International Conference : Sciences of Electronic, Technologies of Information and Telecommunications*. 2007.
- [41] Andrej KRENKER, Janez BEŠTER et Andrej KOS. “Introduction to the artificial neural networks”. In : *Artificial Neural Networks : Methodological Advances and Biomedical Applications. InTech* (2011), p. 1-18.
- [42] Oludare Isaac ABIODUN et al. “Comprehensive review of artificial neural network applications to pattern recognition”. In : *IEEE Access* 7 (2019), p. 158820-158846.
- [48] Oscar SERRADILLA et al. “Deep learning models for predictive maintenance : a survey, comparison, challenges and prospects”. In : *Applied Intelligence* 52.10 (2022), p. 10934-10964.
- [49] Ahlem BOUGUETTAYA. “Une Approche Basée Deep Learning Pour La Classification des Sentiments”. Thèse de doct. Universite laarbi tebessi tebessa, 2019.
- [50] Daniele RAVÌ et al. “Deep learning for health informatics”. In : *IEEE journal of biomedical and health informatics* 21.1 (2016), p. 4-21.
- [51] Rabah BENZITOUNI, Linda MERABET et Soumia ZERTAL. “La Proposition d’une Architecture basée Deep Learning pour la prédiction des maladies cardiaques dans un environnement IoT”. In : (2022).
- [52] Oussama Akram BENSIAH et Mohamed BERKANE. “La Proposition d’une nouvelle approche basée Deep learning pour la prédiction du cancer du sein”. In : (2020).
- [53] Hao YE et al. “Deep learning-based end-to-end wireless communication systems with conditional GANs as unknown channels”. In : *IEEE Transactions on Wireless Communications* 19.5 (2020), p. 3133-3143.
- [54] Qingmao ZENG et al. “Gans-based data augmentation for citrus disease severity detection using deep learning”. In : *IEEE Access* 8 (2020), p. 172882-172891.
- [55] Zhipeng CAI et al. “Generative adversarial networks : A survey toward private and secure applications”. In : *ACM Computing Surveys (CSUR)* 54.6 (2021), p. 1-38.
- [56] Ruhul Amin KHALIL et al. “Deep learning in the industrial internet of things : Potentials, challenges, and emerging applications”. In : *IEEE Internet of Things Journal* 8.14 (2021), p. 11016-11040.

- [57] Khanh TP NGUYEN et Kamal MEDJAHER. “A new dynamic predictive maintenance framework using deep learning for failure prognostics”. In : *Reliability Engineering & System Safety* 188 (2019), p. 251-262.
- [58] Bingxu LI et al. “A semi-supervised approach to fault detection and diagnosis for building HVAC systems based on the modified generative adversarial network”. In : *Energy and Buildings* 246 (2021), p. 111044.
- [59] Romane SCHERRER et al. “Détection de Bateaux de Plaisance dans une Marina par Deep Learning.” In : *EGC*. 2021, p. 47-58.
- [60] Sharma et AL. “Predictive Maintenance : Comparative Study of Machine Learning Algorithms for Fault Diagnosis”. In : *Available at SSRN 4143868* (2022).
- [61] Idrus ASSAGAF et al. “Machine Predictive Maintenance by Using Support Vector Machines”. In : *Recent in Engineering Science and Technology* 1.01 (2023), p. 31-35.

# Webographie

- [28] Nicolas P ROUGIER. *Intelligence artificielle et singularité*. [Consulté le 3 mars 2023].
- [38] *The Concept of Artificial Neurons (Perceptrons) in Neural Networks*. <https://towardsdatascience.com/the-concept-of-artificial-neurons-perceptrons-in-neural-networks-fab22249cbfc>. [consulté le 20 avril 2023].
- [40] *Yandex*. <https://yandex.com/>. [Consulté le 25 avril 2023].
- [43] *Difference Between Machine Learning and Deep Learning*. <https://levity.ai/blog/difference-machine-learning-deep-learning>. [consulté le 08 mars 2023].
- [44] *Deep Learning vs Machine Learning*. <https://nixustechnologies.com/deep-learning-vs-machine-learning/>. [consulté le 8 mars 2023].
- [45] *Machine Learning vs Deep Learning*. <https://www.scs.org.sg/articles/machine-learning-vs-deep-learning>. [consulté le 8 mars 2023].
- [46] *Machine Learning Applications*. <https://www.apriorit.com/dev-blog/472-machine-learning-applications>. [consulté le 20 mars 2023].
- [47] *Machine Learning vs Deep Learning : What's the Difference ?* [https://dev.to/bpb\\_online/machine-learning-vs-deep-learning-what-s-the-difference-2bm7](https://dev.to/bpb_online/machine-learning-vs-deep-learning-what-s-the-difference-2bm7). [Consulté le 8 mars 2023].



# Annexes

# Annexe A

## Présentation de la Raffinerie d'Alger

SONATRACH est une société nationale pour la recherche, la production, le transport, la transformation et la commercialisation des hydrocarbures.

C'est la première entreprise du continent Africain. Elle est classée 12ème parmi les compagnies pétrolières mondiales, 2ème exportateur de GPL et 3ème exportateur de gaz naturel, ses missions sont :

- Le raffinage des hydrocarbures et dérivés de toutes les opérations qui sont liées.
- Le développement de toutes activités conjointes en Algérie et hors Algérie.
- Le respect des spécifications et de la quantité des produits raffinés.
- Toutes opérations industrielles, commerciales, financières, et immobilières.

### A.1 Historique de la raffinerie

La raffinerie d'Alger fait partie des trois raffineries appartenant au groupe NAFTEC, avec celles de SKIKDA et d'ARZEW, sa construction remonte à l'année 1961 par la société française FOSTER WHEELER, et sa production a débuté en 1964. Elle était sous la gestion des français, gérée par la société de la raffinerie d'Alger, et ce jusqu'à la nationalisation des hydrocarbures en 1971 où l'activité de raffinage fut affectée à la SONATRACH.

En 1998, le raffinage et la distribution des produits sont passés à L'EDRP-NAFTEC, puis, en 1987, le raffinage est à son tour séparé de la distribution et affecté à NFTEC. Depuis 1988, les actions de NAFTEC sont détenues à 100% par la SONATRACH. En juillet 2009 NAFTEC a été dissoute et devenu carrément SONATRACH (Activité Aval, Division Raffinage).

### A.2 Capacité de production

Cette raffinerie a pour objectif de traiter le pétrole brut de HASSI-MESSOUAD seul ou mélange avec du condensat provenant des champs de HASSI-R'MEL, afin d'obtenir des coups pétroliers. L'unité a une capacité de traiter 2.700.000 tonnes par an de brut, soit une allure de 7700 tonnes par jours, pour 350 jours de marche.

### A.3 Organigramme de la raffinerie d'Alger

Le schéma (A.1) suivant présente l'organigramme de la raffinerie d'Alger.

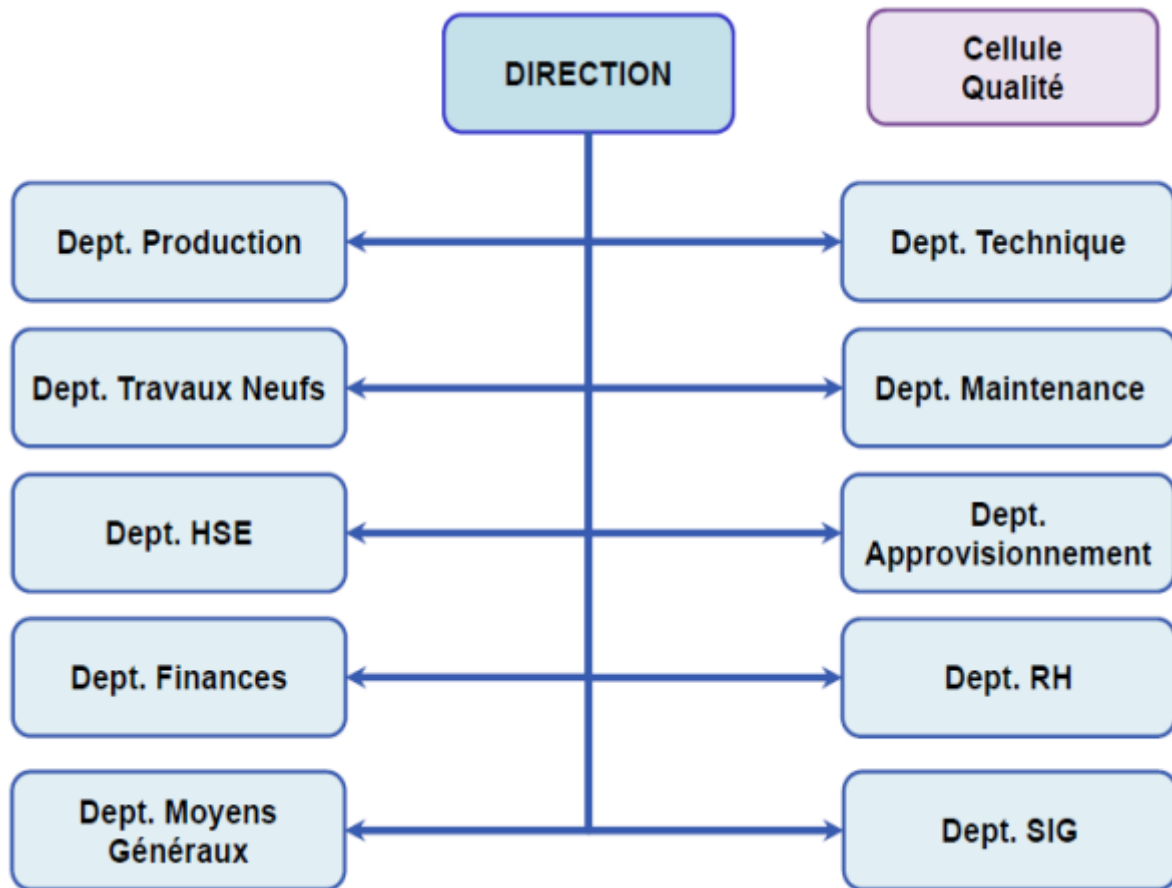


FIG. A.1 : Organigramme de la raffinerie d'Alger

# Annexe B

## Le compresseur 520-k-052A/B

Le compresseur 520-k-052A/B est compresseur à double piston qui est un dispositif mécanique utilisé pour augmenter la pression d'un fluide, généralement de l'air ou un gaz. Il est équipé de deux pistons qui se déplacent en sens opposé l'un par rapport à l'autre. Lorsque l'un des pistons se déplace vers le bas, il aspire le fluide dans la chambre de compression. En remontant, il comprime le fluide, augmentant ainsi sa pression. Pendant ce temps, le deuxième piston effectue un mouvement opposé pour assurer un fonctionnement continu. Les compresseurs à double piston offrent des avantages tels qu'une pression élevée, une efficacité énergétique et une durabilité, et ils sont largement utilisés dans diverses applications industrielles.

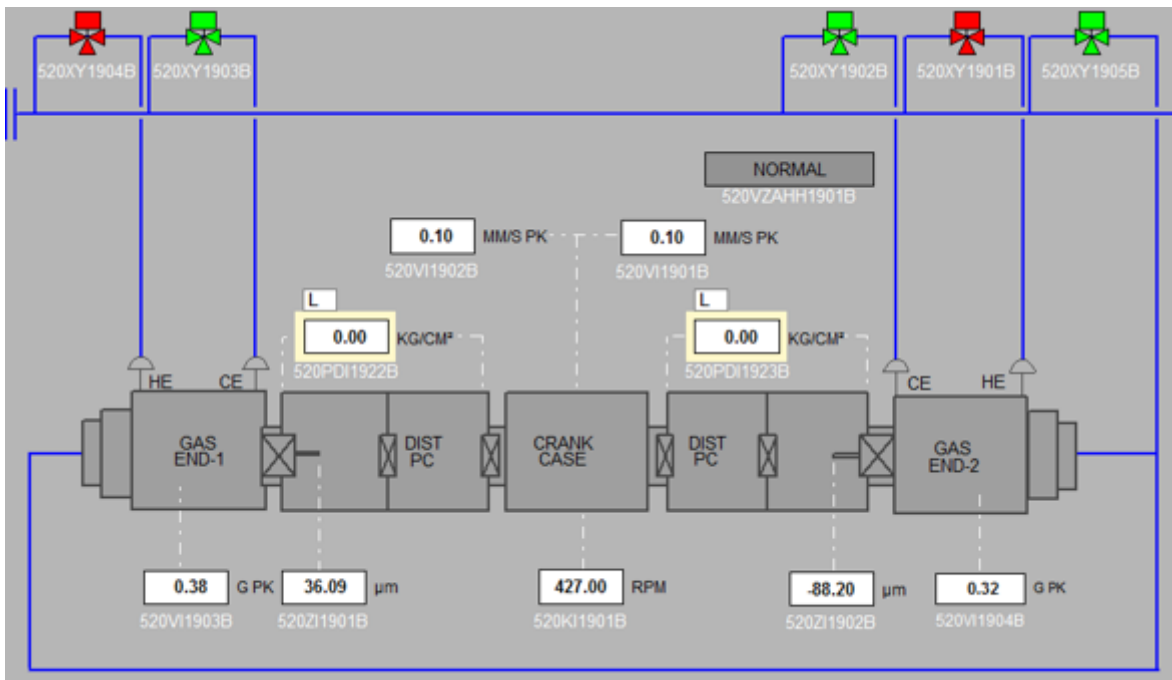


FIG. B.1 : Compresseur 520-k-052A/B

# Annexe C

## Les codes de simulation

### C.1 LSTM

```
#Importer les bibliothèques
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, Dropout
from sklearn.model_selection import RandomizedSearchCV
from sklearn.metrics import accuracy_score, f1_score
from tensorflow.keras.wrappers.scikit_learn import KerasClassifier
import numpy as np
from sklearn.metrics import accuracy_score, roc_auc_score,
    confusion_matrix, classification_report

# Définir la fonction de création du modèle LSTM
def create_lstm_model(hidden_layer_sizes=(50,), activation='relu',
    solver='adam', alpha=0.0001, learning_rate='constant', epochs
    =10, batch_size=32, dropout_rate=0.0):
    model = Sequential()
    for i, units in enumerate(hidden_layer_sizes):
        if i == 0:
            model.add(LSTM(units, activation=activation,
                input_shape=(X_train.shape[1], 1)))
        else:
            model.add(LSTM(units, activation=activation))
        if dropout_rate > 0.0:
            model.add(Dropout(dropout_rate))
    model.add(Dense(1, activation='sigmoid'))
    model.compile(loss='binary_crossentropy', optimizer=solver,
        metrics=['accuracy'])
    return model

# Paramètres de recherche aléatoire
param_grid = {
```

```
'hidden_layer_sizes' : [(100,), (110,), (50, 50), (100, 50)],
'activation' : ['relu', 'sigmoid'],
'solver' : ['adam', 'sgd'],
'alpha' : [0.0001, 0.001, 0.01],
'learning_rate' : ['constant', 'adaptive'],
'epochs' : [50, 64, 100],
'batch_size' : [32, 64],
'dropout_rate' : [0.2, 0.3, 0.4]
}

# Créer un modèle KerasClassifïer à partir de la fonction LSTM
lstm_model = KerasClassifïer(build_fn=create_lstm_model, verbose=0)

# Recherche aléatoire des hyperparamètres
random_search = RandomizedSearchCV(estimator=lstm_model,
    param_distributions=param_grid, scoring='accuracy', cv=5, n_iter
    =10)
random_search.fit(X_train, y_train)

best_params = random_search.best_params_
best_score = random_search.best_score_

# Créer un nouveau modèle LSTM avec les meilleurs paramètres
model = create_lstm_model(**best_params)
model.fit(X_train, y_train, epochs=best_params['epochs'],
    batch_size=best_params['batch_size'], verbose=0)

# Prédiction sur les données de test
y_pred_lstm = np.round(model.predict(X_test)).flatten()

#Calculer les métriques d'évaluation
lstm_accuracy = accuracy_score(y_test, y_pred_lstm)
LSTM_cm = confusion_matrix(y_test, y_pred_lstm)
LSTM_report = classification_report(y_test, y_pred_lstm)

# Afficher les résultats
print('LSTM_accuracy :%.2f%%' % (lstm_accuracy* 100))
print('LSTM_Confusion Matrix :')
print(LSTM_cm)
print('LSTM_Classification Report :')
print(LSTM_report)
```

## C.2 RNN

```
#Importer les bibliothèques
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import SimpleRNN, Dense, Dropout
from sklearn.model_selection import RandomizedSearchCV
from sklearn.metrics import accuracy_score, f1_score
from tensorflow.keras.wrappers.scikit_learn import KerasClassifier
import numpy as np
from sklearn.metrics import accuracy_score, roc_auc_score,
    confusion_matrix, classification_report

# Définir la fonction de création du modèle RNN
def create_rnn_model(hidden_layer_sizes=(50,), activation='relu',
    solver='adam', alpha=0.0001, learning_rate='constant', epochs
    =10, batch_size=32, dropout_rate=0.0):
    model = Sequential()
    for i, units in enumerate(hidden_layer_sizes):
        if i == 0:
            model.add(SimpleRNN(units, activation=activation,
                input_shape=(X_train.shape[1], 1)))
        else:
            model.add(SimpleRNN(units, activation=activation))
        if dropout_rate > 0.0:
            model.add(Dropout(dropout_rate))
    model.add(Dense(1, activation='sigmoid'))
    model.compile(loss='binary_crossentropy', optimizer=solver,
        metrics=['accuracy'])
    return model

# Paramètres de recherche aléatoire
param_grid = {
    'hidden_layer_sizes': [(100,), (110,), (50, 50), (100, 50)],
    'activation': ['relu', 'sigmoid'],
    'solver': ['adam', 'sgd'],
    'alpha': [0.0001, 0.001, 0.01],
    'learning_rate': ['constant', 'adaptive'],
    'epochs': [50, 64, 100],
    'batch_size': [32, 64],
    'dropout_rate': [0.2, 0.3, 0.4]
}

# Créer un modèle KerasClassifier à partir de la fonction RNN
rnn_model = KerasClassifier(build_fn=create_rnn_model, verbose=0)

# Recherche aléatoire des hyperparamètres
```

```
random_search = RandomizedSearchCV(estimator=rnn_model,
    param_distributions=param_grid, scoring='accuracy', cv=5, n_iter
    =10)
random_search.fit(X_train, y_train)

best_params = random_search.best_params_
best_score = random_search.best_score_

# Créer un nouveau modèle RNN avec les meilleurs paramètres
model = create_rnn_model(**best_params)
model.fit(X_train, y_train, epochs=best_params['epochs'],
    batch_size=best_params['batch_size'], verbose=0)

# Prédiction sur les données de test
y_pred_rnn = np.round(model.predict(X_test)).flatten()

# Calculer les métriques d'évaluation
rnn_accuracy = accuracy_score(y_test, y_pred_rnn)
RNN_cm = confusion_matrix(y_test, y_pred_rnn)
RNN_report = classification_report(y_test, y_pred_rnn)

# Afficher les résultats
print('RNN_accuracy :%.2f%%' % (rnn_accuracy* 100))
print('RNN_Confusion Matrix :')
print(RNN_cm)
print('RNN_Classification Report :')
print(RNN_report)
```



## C.3 CNN

```
#Importer les bibliothèques
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv1D, MaxPooling1D, Flatten,
    Dense, Dropout
from sklearn.model_selection import RandomizedSearchCV
from sklearn.metrics import accuracy_score, f1_score
from tensorflow.keras.wrappers.scikit_learn import KerasClassifier
import numpy as np
from sklearn.metrics import accuracy_score, roc_auc_score,
    confusion_matrix, classification_report

# Définir la fonction de création du modèle CNN
def create_cnn_model(filters=32, kernel_size=3, pool_size=2,
    hidden_units=64, dropout_rate=0.0, epochs=10, batch_size=32,
    activation='relu', solver='adam') :
    model = Sequential()
    model.add(Conv1D(filters=filters, kernel_size=kernel_size,
        activation=activation, input_shape=(X_train.shape[1], 1)))
    model.add(MaxPooling1D(pool_size=pool_size))
    model.add(Flatten())
    model.add(Dense(hidden_units, activation=activation))
    if dropout_rate > 0.0 :
        model.add(Dropout(dropout_rate))
    model.add(Dense(1, activation='sigmoid'))
    model.compile(loss='binary_crossentropy', optimizer=solver,
        metrics=['accuracy'])
    return model

# Paramètres de recherche aléatoire
param_grid = {
    'filters' : [32, 64, 128],
    'kernel_size' : [3, 5],
    'pool_size' : [2, 3],
    'hidden_units' : [64, 110, 128],
    'dropout_rate' : [0.3, 0.4, 0.5],
    'epochs' : [50, 64, 100],
    'batch_size' : [32, 64],
    'activation' : ['relu', 'sigmoid'],
    'solver' : ['adam', 'sgd']
}

# Créer un modèle KerasClassifier à partir de la fonction CNN
cnn_model = KerasClassifier(build_fn=create_cnn_model, verbose=0)
```

```
# Recherche aléatoire des hyperparamètres
random_search = RandomizedSearchCV(estimator=cnn_model,
    param_distributions=param_grid, scoring='accuracy', cv=5, n_iter
    =10)
random_search.fit(X_train, y_train)

best_params = random_search.best_params_
best_score = random_search.best_score_

# Créer un nouveau modèle CNN avec les meilleurs paramètres
model = create_cnn_model(**best_params)
model.fit(X_train, y_train, epochs=best_params['epochs'],
    batch_size=best_params['batch_size'], verbose=0)

# Prédiction sur les données de test
y_pred_cnn = np.round(model.predict(X_test)).flatten()

# Calculer les métriques d'évaluation
cnn_accuracy = accuracy_score(y_test, y_pred_cnn)
CNN_cm = confusion_matrix(y_test, y_pred_cnn)
CNN_report = classification_report(y_test, y_pred_cnn)

# Afficher les résultats
print('CNN_accuracy :%.2f%%' % (cnn_accuracy* 100))
print('CNN_Confusion Matrix :')
print(CNN_cm)
print('CNN_Classification Report :')
print(CNN_report)
```

## ملخص

في الوقت الحاضر، يشهد مجال البحث اهتمامًا متزايدًا في استخدام التعلم العميق، وهو فرع من التعلم الآلي والذكاء الاصطناعي، في مجال الإنترنت الصناعي للأشياء. إن التنبؤ بفشل المعدات الصناعية قبل حدوثها، أو ما يسمى بالصيانة التنبؤية، هو أيضًا موضوع شائع جدًا، مدفوعًا بشكل أساسي بإمكانية توفير التكاليف. تقنيات التعلم العميق قادرة على توفير عدة فوائد في سياق الصيانة التنبؤية. لذلك، هناك مجموعة من الخوارزميات، لكل منها خصائص مميزة. لذلك، فإن اختيار الأفضل لحل المشكلات ليس سهلاً. يهدف هذا العمل إلى تحسين أداء الصيانة التنبؤية من خلال اختيار أنسب خوارزميات التعلم العميق. ولهذا، تم إجراء مقارنة تستند إلى عدة معايير، وتم اختيار ثلاث خوارزميات. تم تنفيذ هذه دراسة واختبارها باستخدام بيانات حقيقية وبيانات محاكاة من أجل تحديد الخوارزمية الأكثر ملاءمة للصيانة التنبؤية.

**كلمات مفتاحية :** التعلم العميق، الشبكات العصبية، التنبؤ، اختيار الخوارزمية.

## Résumé :

Actuellement, la recherche accorde un intérêt croissant à l'utilisation du Deep Learning (DL), une branche du Machine Learning (ML) et de l'Intelligence Artificielle (IA), dans le domaine de l'Internet industriel des objets. La Maintenance Prédictive (PdM), qui consiste à prédire les pannes des équipements industriels avant qu'elles ne se produisent, est également un sujet très populaire, principalement en raison de son potentiel d'économies. Les techniques DL offrent des avantages significatifs dans les applications de PdM. Cependant, étant donné la variété d'algorithmes DL disponibles, chacun ayant ses propres caractéristiques distinctes, il n'est pas facile de choisir le meilleur algorithme DL pour la PdM. Ce travail a pour objectif d'améliorer les performances de la PdM en sélectionnant les algorithmes DL les plus appropriés. Pour ce faire, une étude comparative basée sur plusieurs critères a été réalisée, qui a abouti à la sélection de trois algorithmes. Ces algorithmes ont été implémentés avec des données réelles et simulées afin d'identifier celui qui convient le mieux à la PdM.

**Mots clés :** Deep Learning, Réseaux de neurones, PdM, Sélection d'algorithmes.

## Abstract :

Presently, the realm of research is witnessing a burgeoning interest in the utilization of Deep Learning (DL), a branch of machine learning (ML) and Artificial Intelligence (AI), within the domain of the Industrial Internet of Things. The prediction of failures of industrial equipment before their occurrence, referred to as Predictive Maintenance (PdM), is also a very trendy, primarily driven by its potential for cost-savings. DL techniques have the potential to provide significant benefits in the context of PdM applications. However, there are a variety of DL algorithms available, and each each possessing distinct characteristics. Therefore, choosing the best DL algorithm to resolve PdM issues is not trivial. This work aims to improve the performance of PdM by selecting the most appropriate DL algorithms. To achieve this goal, a comparative study based on several criteria was carried out, which resulted in the selection of three algorithms. These have been implemented with real and simulation data in order to identify which DL algorithm is most appropriate for PdM.

**Keywords :** Deep Learning, Neural Networks, PdM, Algorithm Selection.