REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE YAHIA FARES DE MEDEA



Faculté de **T**echnologie

Département du **G**énie **E**lectrique

Mémoire de fin d'études de Master

Filière : Télécommunications

Spécialité : Système de Télécommunications

Etude et implémentation d'un système intelligent d'aide à la diagnostic automatique des défauts de roulements dans les machines tournantes.

Présentés par :

BENREKIA Mustapha

MEKHTICHE Oussama

Proposés et Dirigés par :

BENGHERBIA Billel

Promotion 2022

Remerciement

Avant tout, nous rendons grâce d'ALLAH de nous avoir accordé la santé et la volonté et le courage pour réaliser ce travail.

Nous tenons à saisir nos profonds remerciements et nos profondes reconnaissances à chers parents pour leurs soutient et leur encouragements.

Nous désirons nos sincères remerciements à nos promoteurs
Dr. BENGHERBIA Billel et Mr. TOUMI Yacine
pour ses précieux conseils et son orientation ficelée tout au long de notre recherche.

Dédicaces

Je tiens c'est avec grand plaisir que je dédie ce modeste travail :

À l'être le plus cher de ma vie, ma mère.

À celui m'a fait un homme, mon père.

À mes chers frères Ismain et Riadh.

À mes chères sœurs Romaissa et Rofaida.

À l'âme de mon cher ami et frère ATSAMNIA Nouh que Dieu ait son âme.

À tous personne qui occupe une place dans mon cœur.

À mon meilleur Binôme MEKHTICHE Oussama.

Et mes très chers collègues sans exception.

À tous mes professeurs qui nous ont enseignés.

…BENREKIA Mustapha

Dédicaces

Avant tout, je rends grâce à DIEU de m'avoir accordé la volonté et le courage pour réaliser ce mémoire.

Je dédie ce modeste travail à :

Tout d'abord je dédie ce modeste travail A ma **chère mère**, A mon **cher père**.

Qui m'ont donné un magnifique modèle de labeur et persévérance, je ne trouverai jamais les

mots qu'il faut pour vous remercier que le bon dieu le tout puissant vos grande pour moi.

A mes sœur **MERIEM**, **MANAR**, **NOUR**, **NIHEL**

A Mon meilleure Binôme **BENRKIA MUSTAPHA**

A tous mes professeurs qui nous ont enseignés.

A mes très chers collègues sans exception (Benkhaoua Nadjet, Berkani rafik,Achir Zahra,Douik Nadjet)

A tous ceux qui ont participé à l'élaboration de ce modeste travail et tous ceux qui nous sont chères.

…MEKHTICHE Oussama

ملخص

يقوم عملنا هذا على استخراج خصائص اشارات الإهتزاز المنبثقة من الألات الدوارة و تحديدا تلك الصادرة من محامل الكريات, تتكون الطريقة المتبعة على القيام بمعالجة مسبقة للاشارات الإهتزازية ثم معرفة مكان الخلل على المحمل وذلك باستخدام الشبكة العصبية التلافيفية. تم تطبيق نموذج شبكة CNN على بطاقة Raspberry.

لقد برهن النموذج المقترح نجاعته و فعاليته و قدم نتائج تصنيف مرضية.

الكلمات المفتاحية : إشارات اهتزازية, محامل الكريات, الشبكة العصبية التلافيفية,CNN, Raspberry

Résumé

Notre travail s'appuie sur l'extraction des caractéristiques des signaux vibratoires issus des machines tournantes, notamment celles des roulements à billes. La méthode utilisée consiste à prétraitement sur les signaux vibratoires puis à connaître la localisation du défaut sur le roulement à l'aide d'un réseau de neurones convolutif (CNN). Le modèle CNN a été implémenté sur la carte Raspberry.

Le modèle proposé a démontré son efficacité et a fourni des résultats de classification satisfaisants.

Mots clé : signaux vibratoires, roulements à billes, réseau de neurones convolutif, CNN, Raspberry.

Abstract

Our work is based on the extraction of the characteristics of vibratory signals from rotating machines, especially those from ball bearings. The method used is to pretreat the vibration signals and then to know the location of the defect on the bearing ball using a convolutional neural network (CNN). The CNN model has been implemented on the Raspberry card.

The proposed model demonstrated its effectiveness and provided satisfactory classification results.

Keywords: vibration signals, bearing ball, convolutional neural network, CNN, Raspberry.

Table des matières

Résumé v

Table des matières vi

Liste des figures ix

Liste des tableaux xi

Liste des abréviations xii

Introduction générale 14

Chapitre I: Diagnostic des défauts dans les machines tournantes par analyse vibratoire

I.1 Introduction 3

I.2 La maintenance 3

I.2.1 Définition de la maintenance 3

I.2.2 But de la maintenance 3

I.2.3 Les types de maintenance 4

I.3 Choix de types d'entretien 6

I.4 Analyse vibratoire 6

I.4.1 Définition d'une vibration 7

I.4.2 Caractéristiques des vibrations 7

I.4.3 Analyse temporelle et analyse fréquentielle 8

I.5 Les défauts des roulements 10

I.5.1 Structure mécanique d'un roulement 11

I.5.2 Causes des défauts des roulements à billes 12

I.5.3 Les types de défauts 14

I.5.4 Les fréquences des défauts 16

I.6 Traitement des signaux vibratoires 16

I.6.1 Segmentation 17

I.6.2 Transformée de Fourier 17

I.6.3 Définition de TEO 18

I.6.4 Historique de la TEO 19

I.6.5 Notion de Base 19

I.7 Transformer d'Hilbert 20

I.8 Conclusion 20

Chapitre II: Technique d'intelligence artificiel

II.1 Introduction 21

II.2 Deep Learning 21
II.3 Détection des anomalies 22
II.4 Type d'apprentissage 22
II.4.1 Apprentissage supervisé 23
II.4.2 Apprentissage non-supervisé 23
II.4.3 Différence entre apprentissage supervisé et non supervisé 23
II.5 Réseau Neurones 24
II.5.1 Neurone biologique 24
II.5.2 Architecture des réseaux de neurones artificiels 24
II.5.3 Le perceptron 25
II.5.4 Fonction d'activation 26
II.6 Types de réseaux de neurones 28
II.6.1 Réseaux de neurones artificiels (ANN) 28
II.6.2 Réseau de neurones à convolution (CNN) 28
II.7 Conclusion 35
Chapitre III: Mise en marche, test et implémentation
III.1 Introduction 36
III.2 Base de données choisie 36
III.3 Présentation des outils utilisés 38
III.3.1 Matlab 38
III.3.2 Python 38
III.4 Prétraitement des signaux 41
III.4.1 Application de la méthode Hilbert sur les signaux vibratoires 41
III.4.2 Application de la méthode TEO FFT TEO sur les signaux vibratoires 42
III.4.3 Comparaison entre (TEO FFT TEO) et Hilbert 43
III.5 Préparation des données 43
III.5.1 Décomposition des signaux 43
III.5.2 Remplissage des signaux 44
III.5.3 Enregistrement 44
III.5.4 Distribution de base de données 45
III.6 Modèle CNN 46
III.6.1 Architecture utiliser 46
III.6.2 Méthode d'optimisation 49

III.7 Résultats de classification 51
III.7.1 Graphe de précision et d'erreur 51
III.7.2 Rapport de classification 52
III.7.3 Matrice de Confusion 52
III.8 Comparaison 53
III.9 Résultats du test de modèle CNN 54
III.9.1 Chargée (Load) modèle CNN 54
III.9.2 Préparation Signaux vibratoire 55
III.9.3 Classement des Signaux 55
III.10 Système d'implémentation 56
III.10.1 Raspberry Pi 56
III.10.2 Raspberry Pi 3 B+ 56
III.11 Conclusion 60
Conclusion générale 61
Bibliographie 62

Liste des figures

Figure I-1: Les différents types de maintenance. 4
Figure I-2: Maintenance corrective. 4
Figure I-3: Maintenance préventive systématique. 5
Figure I-4: Maintenance préventive conditionnelle. 5
Figure I-5: Les différentes méthodes de diagnostic des machines tournantes. 7
Figure I-6: Exemple d'une vibration. 7
Figure I-7: Caractéristique d'une vibration (cas d'un signal sinusoïdal). 8
Figure I-8: Exemple de signal dans le domaine temporel. 9
Figure I-9: Spectre d'amplitude du signal x(t) résultant du calcul de la DFT. 10
Figure I-10: Les sources de défaillances les plus rependues dans les moteurs. 11
Figure I-11: Structure mécanique d'un roulement à billes rigide. 11
Figure I-12: Choc électrique qui a endommagé un élément roulant. 12
Figure I-13: Corrosion sur la bague externe d'un roulement à billes. 12
Figure I-14: Défaut d'usure sur un roulement à billes. 13
Figure I-15: Défaut de bavure sur un roulement à billes. 13
Figure I-16: Défaut d'écaillage sur la bague interne d'un roulement à billes. 14
Figure I-17: Défaut de dentelure sur la bague interne d'un roulement à billes. 14
Figure I-18: Défaut au niveau de la bague externe d'un roulement avec son spectre. 15
Figure I-19: Défaut au niveau de la bague externe d'un roulement avec son spectre. 15
Figure I-20: Défaut au niveau de la bille d'un roulement avec son spectre. 16
Figure I-21: exemple de segmentation A: signal complet, B: signal segmenté. 17
Figure II-1: Les domaines de l'intelligence artificielle. 22
Figure II- 2: exemple explicatif de la déférence entre apprentissage. 23
Figure II-3: Structure d'un neurone biologique. 24
Figure II-4: Architecture d'un réseau de neurones artificiels. 25
Figure II-5: Neurone artificiel simple 'Le perceptron' qui prend une somme pondérée à l'entrée et détermine si la sortie est active ou non. 26
Figure II-6: Forme des fonctions d'activations les plus utilisées 28
Figure II-7: Un exemple de configuration CNN 1D avec 3 couches CNN et 2 couches MLP. 29
Figure II-8: Opération de convolution. 30
Figure II-9: opération convolution avec padding. 31
Figure II-10: Application de la fonction d'activation ReLU. 31
Figure II-11: Application de MaxPooling. 32
Figure II-12: Application d'AveragePooling. 33
Figure II-13: Introduction du Dropout dans un réseau ANN, la moitié des neurones sont désactivés. 35
Figure III- 1: Banc d'essai des défauts des roulements. 36
Figure III- 2: Logo TensorFlow. 39
Figure III- 3: Logo Keras. 40
Figure III- 4: Logo Sklearn. 40
Figure III- 5: Logo Matplotlib. 41

Figure III- 6: Application de Hilbert. 41
Figure III- 7: Application de la TEO FTT TEO. 42
Figure III- 8: Organigramme de traitement des données. 43
Figure III- 9: Décomposition des signaux. 44
Figure III- 10: Remplissage des signaux. 44
Figure III- 11: Enregistrement sous forme csv. 45
Figure III- 12: Distribution de la Base de données. 45
Figure III- 13: Scatter plot du Base de données. 46
Figure III- 14: Diagramme schématique de procédure d'entrainement des réseaux CNN. 46
Figure III- 15: L'Architecture utilisée. 49
Figure III- 16: K-Fold Cross Validation. 50
Figure III- 17: A- Graphe de Loss, B-Graphe de l'accuracy. 51
Figure III- 18: Rapport de Classification. 52
Figure III- 19: Matrice de confusion de validation. 53
Figure III- 20: A- Graphe de Loss sans KFold, B-Graphe de l'accuracy sans KFold. 54
Figure III- 21: Test expérimental. 54
Figure III- 22: Classement des signaux. 55
Figure III- 23: Matrice de confusion de test. 55
Figure III- 24: Raspberry Pi 3 B+. 57
Figure III- 25: Préparation des données (Raspberry). 58
Figure III- 26: résultat d'implémentation avec Raspberry. 59

Liste des tableaux

Tableau I-1: Comparaison entre la DFT et la FFT en terme de complexité de calcul. 18
Tableau II-1: déférence entre apprentissage supervisé et non supervisé. 23
Tableau III- 1: Fréquences des défauts de roulements. 37
Tableau III- 2: Les couches utilisent dans cette architecture. 47
Tableau III- 3: Meilleur Résultat obtenu avec KFold. 53
Tableau III- 4: Meilleur Résultat obtenu sans KFold. 53
Tableau III- 5: les résultats de test expérimental. 56
Tableau III- 6: Caractéristiques du Raspberry Pi 3 B+. 57
Tableau III- 7: Résultats des implémentations 59

Liste des abréviations

ANN: Réseaux Neurones Artificiel (Artificial Neural Network)

AM: Modulation d'amplitude (Amplitude modulation)

ARM: Advanced RISC Machines

BN: BatchNormalization (Normalisation par lots)

CNN: Réseaux Neurones à Convolution (Convolutional Neural Network).

CRF: Facteur de Crête (Crest factor)

CSV: Comma-Separated values

CPU: Central Processing Unit

CWRU: Case Western Reserve University

DL: deep learning (l'apprentissage en profondeur)

DE: Drive End

EDM: L'Electronic Miscellaneous Document

ELU: Unité Linéaire Exponentielle (Exponential Linear Units).

EN: Extrait de Norme

FM: Modulation de Fréquence (Frequency modulation.)

FE: Fan End

GPU: Graphics Processing Unit

IA: Intelligence Artificielle (Artificial intelligence)

K: Kurtosis

LAN: Local Area Network

LReLu: Leaky Rectified Linear Unit

Matlab: MATrix LABoratory

ML: Machine Learning (apprentissage automatique)

NF: Norme Afnor

OS: Operating System

ReLu: Unité Linéaire Rectifiée (Rectified Linear Unit).

RMS: la valeur efficace de signale (Root-Mean-Square)

RPM: Révolutions Per Second

SD: Secure Digital

Sklearn: Scikit-Learn.

SoC: System on a Chip

Tanh: Tangente Hyperbolique.

TEO: Opérateur Energétique de Teager (teager energy operator)

TF: Transformé de Fourier (Fourier transform)

TFD: Transformée de Fourier Discrète

USB: Universal Serial Bus

WIFI: Wireless Fidelity

Introduction générale

Introduction générale

Dans l'industrie le gain représente un objectif majeur pour les entreprises. Puisque le coût d'entretien contribue avec une portion substantielle au coût de production, les entreprises doivent avoir un budget optimum d'entretien. Les arrêts d'une machine de production coûtent souvent très cher et il est donc souhaitable de les limiter. Par exemple, les machines tournantes à l'image d'une turbine dans une centrale électrique, elles font partie de ces équipements qu'on ne peut pas se permettre de panne à tout moment. Alors il est nécessaire d'appliquer une stratégie de maintenance adéquate. On doit donc appliquer une maintenance préventive, afin d'alerter l'exploitant à un stade plus ou moins précoce sur l'existence d'un début d'anomalie, source potentielle d'une rupture ou d'un accident grave et pour apprécier l'urgence d'une intervention corrective.

Ceci est indispensable pour prévenir les défaillances et éviter ses arrêts de production imprévus et les pertes économiques qui en découlent. Une solution s'impose : la surveillance par le suivi des défauts de composants mécaniques.

Puisque le fonctionnement d'une machine est lié à chaque instant à son état. Ce genre de stratégie a l'inconvénient d'avoir d'énormes enjeux économiques à cause du suivi quotidien de l'état des machines par les équipes de maintenance.

Cependant, les phénomènes de dégradation d'une machine tournante se traduisent par des valeurs mesurables. L'expérience a montré que l'analyse vibratoire représente un paramètre fiable pour le suivi d'une machine tournante. L'analyse vibratoire est la plus connue et la plus employée dans l'analyse des défauts. Il faut dire qu'elle permet de détecter pratiquement tous les défauts. Ceci permet de définir le juste nécessaire au niveau des interventions de maintenance, la machine est alors révisée seulement si son état le nécessite, ce qui a pour effet de réduire considérablement les temps d'arrêts de production les coûts de maintenance.

L'objectif de ce projet de fin d'étude est d'implémenter une méthode d'extraction des informations utiles d'un signal vibratoire par des techniques d'intelligence artificielle et le développement d'une plateforme embarquée capable de détecter et classifier les défauts, et déduire l'origine du défaut dans une machine tournante. Pour parvenir à cet objectif, Notre travail comporte trois chapitres:

Le premier chapitre sera consacré aux différentes stratégies de maintenance et la présentation des généralités sur la maintenance par analyse vibratoire, porter le plus d'attention aux défectuosités des roulements à billes et de recherche de pré traitements visant à illustrer les analyses vibratoires.

Le deuxième chapitre est consacré à un aperçu de l'apprentissage profond et ces types d'apprentissages et la structure d'un réseau neurone.

Finalement, on élabore un système à base d'intelligence artificielle capable d'identifier et de suivre un défaut de roulement et les classer.

De plus, une implémentation de la méthode proposée sur Raspberry Pi et les résultats des tests effectués pour valider l'efficacité du traitement proposé sont représentés sur le troisième chapitre.

Chapitre I: Diagnostic des défauts dans les machines tournantes par analyse vibratoire

I.1 Introduction

Lorsque les machines sont en bon fonctionnement elles vibrent avec une fréquence bien déterminée. En outre, dès l'apparition des phénomènes d'usure, de fatigue et de vieillissement, ces fréquences changent. L'analyse de vibration permet de détecter les perturbations du système. Ainsi elle devient un élément important de la maintenance conditionnelle. Lors de l'analyse de vibration plusieurs paramètres peuvent être mesurés et sont porteurs de renseignement, tels que l'amplitude de vibration, la vitesse et l'accélération.

Dans ce chapitre, nous présentons différentes méthodes de maintenance et essayons de démontrer la pertinence de l'entretien conditionnel prédictif pour le suivi et la surveillance des machines tournantes.

La surveillance peut se faire par des mesures vibratoires, ces signaux permettent de donner un avis sur l'état de la machine, et l'aspect du signal vibratoire mesuré dépend du type de défauts rencontrés sur les composants mécanique particulièrement intéressant pour les roulements car ils représentent une source commune de défaillance.

L'analyse vibratoire est un des moyens utilisés pour suivre l'état des machines tournantes en fonctionnement. Cela s'inscrit dans le cadre d'une politique de maintenance prévisionnelle de l'outil de production industrielle.

À partir des vibrations régulièrement recueillies sur une machine tournante, l'analyse vibratoire consiste à détecter d'éventuels dysfonctionnements et à suivre leur évolution dans le but de planifier ou reporter une intervention mécanique.

I.2 La maintenance

I.2.1 Définition de la maintenance

La norme Afnor NF X 60-010 définie la maintenance comme suit : « La maintenance est l'ensemble des actions permettant de maintenir ou de rétablir un bien dans un état spécifié ou en mesure d'assurer un service déterminé » [1].

La maintenance dispose d'une panoplie de méthodes, chacune d'entre elles correspond à un concept particulier.

I.2.2 But de la maintenance

- Réduire le nombre de pauses.
- Fiabiliser les outils de production.
- Améliorer son taux de disponibilité.
- Meilleure gestion des stocks de pièces détachées etc.

I.2.3 Les types de maintenance

La présence ou l'absence de panne sur le matériel permet de scinder la maintenance en deux types principaux (figure I-1). La maintenance corrective, qui s'applique après la défaillance de la machine, et la maintenance préventive qui est effectuée avant que la panne ne se manifeste.

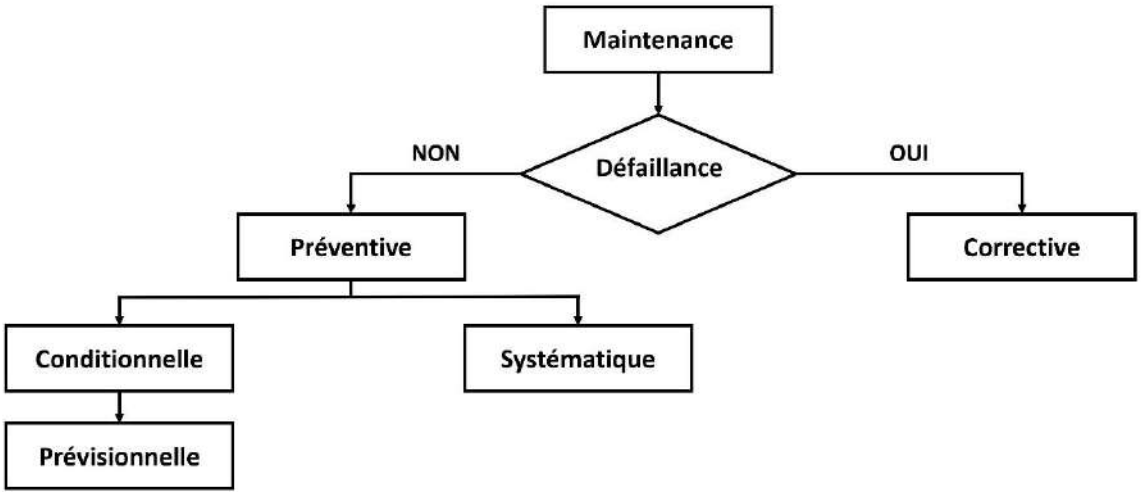


Figure I-1: Les différents types de maintenance.

I.2.3.1 Maintenance corrective

« Maintenance exécutée après détection d'une panne et destinée à remettre un bien dans un état dans lequel il peut accomplir une fonction requise. » (Figure I-2). (Extrait de la norme NF EN 13306 X 60-319).

Ce concept recouvre simplement les actions à entreprendre après une panne.

Ce type de maintenance prend trois formes d'intervention : Dépannage, Réparation et Remplacement.

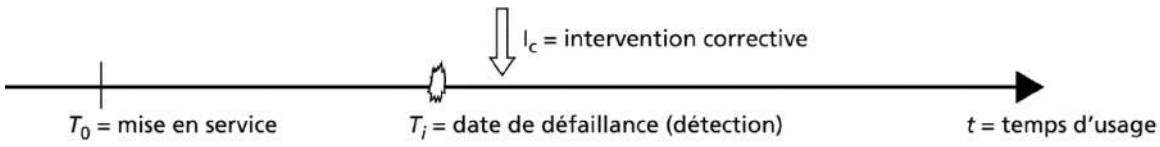


Figure I-2: Maintenance corrective.

I.2.3.2 Maintenance préventive

« Maintenance exécutée à des intervalles prédéterminés ou selon des critères prescrits et destinés à réduire la probabilité de défaillance ou la dégradation du fonctionnement d'un bien».

Il existe deux formes particulières de maintenance préventive :

- ***Maintenance systématique :***

« 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» (figure I-3).

Il s'agit d'un entretien préventive effectué selon un calendrier fixe basé sur la durée de vie la plus courte des composants mécaniques. Les coûts de maintenance de ce type sont élevés, surtout avec des pièces de rechange régulières.

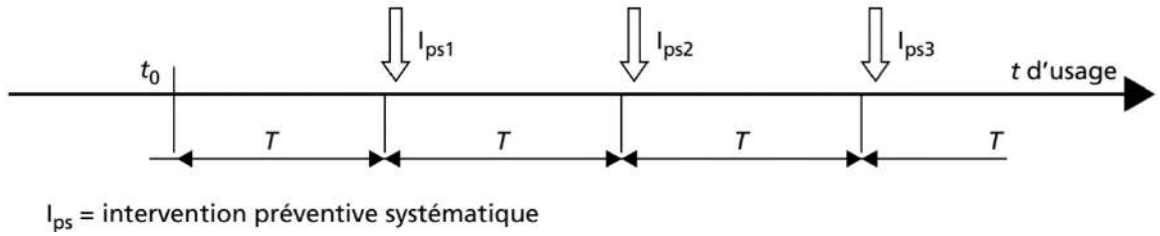


Figure I-3: Maintenance préventive systématique.

- ***Maintenance conditionnelle :***

« 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. »

Pour les équipements importants, la maintenance préventive systématique est de plus en plus remplacée par la maintenance préventive conditionnelle. Dans ce cas la maintenance déclenchée uniquement lorsque certains paramètres mesurables atteignent des seuils limites (figure I-4) [1].

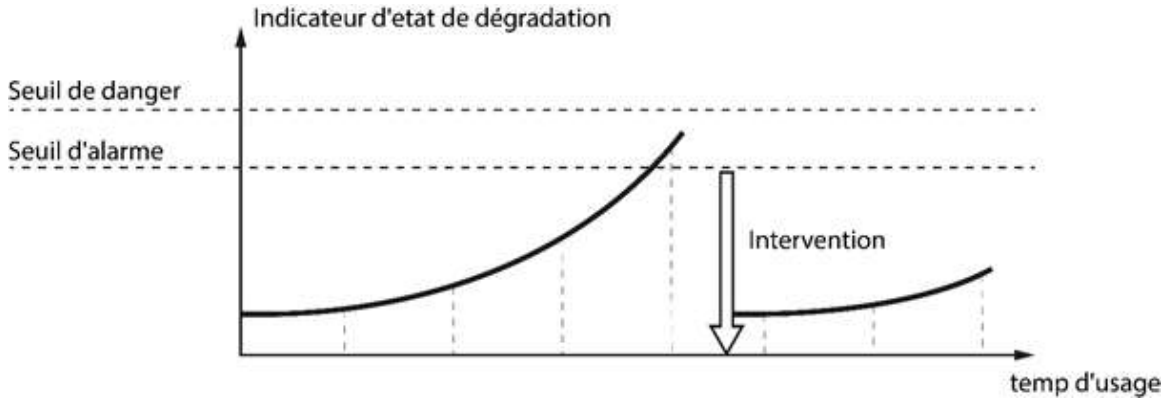


Figure I-4: Maintenance préventive conditionnelle.

- ***Maintenance prévisionnelle :***

« La maintenance prévisionnelle est une maintenance conditionnelle exécutée en suivant les prévisions extrapolées de l'analyse et de l'évaluation des paramètres significatifs de la dégradation du bien. » La maintenance prévisionnelle est une variante de la maintenance

conditionnelle qui est réalisé par prédiction extrapolée après analyse paramétrique et évaluation dégâts matériels importants [1].

I.3 Choix de types d'entretien

Actuellement, chaque entreprise choisit le type de maintenance qui convient le mieux à ses activités, et les grandes entreprises industrielles ont immédiatement adopté une stratégie en qui leur permettant de garantir la maintenance de leurs outils de production comme les systèmes de maintenance préventive systématique. Mais il ne faut pas oublier que cette approche systématique coûte cher, car dans la plupart des cas, les organes mécaniques Changé alors qu'il est encore loin d'être critique, ce qui signifie que la machine peut fonctionner plus longtemps sans intervention et sans aucun problème [2].

Les traitements sont toujours disponibles et chers. De ce constat émerge une nouvelle approche, comme système de la maintenance, qui veut intervenir avant les pannes, mais basée sur des éléments plus fiables pour réparer les outils de production : Il s'agit d'une maintenance conditionnelle (prédictive).

I.4 Analyse vibratoire

Pendant le fonctionnement, les machines tournantes génèrent diverses forces telles que le couple, les chocs ou les frottements provoquant l'usure de divers composants constituant la machine. Ensuite, nous pouvons juger de l'état de dégradation de la machine selon certaines observations telles que la chaleur dégagée, l'analyse des débris dans l'huile, et même le bruit et les vibrations observé pendant le fonctionnement. La vibration dépend directement des forces générées par les différents éléments mobiles internes, pondérés par la fonction de transfert des connexions (paliers, membranes huile, film d'air, boulons, supports élastiques, ressorts) et présente de grands avantages En raison de la richesse des informations qu'ils véhiculent, ce sont des transmissions quasi instantanées et en relation directe avec le mouvement de la machine.

Par conséquent, la vibration occupe une place privilégiée dans les paramètres à considérer pour assurer le suivi et l'efficacité de fonctionnement des machines tournantes. Par conséquent, la plupart les méthodes modernes de maintenance des machines utilisent l'analyse vibratoire (Figure I-5), désormais très courantes dans l'industrie [3].

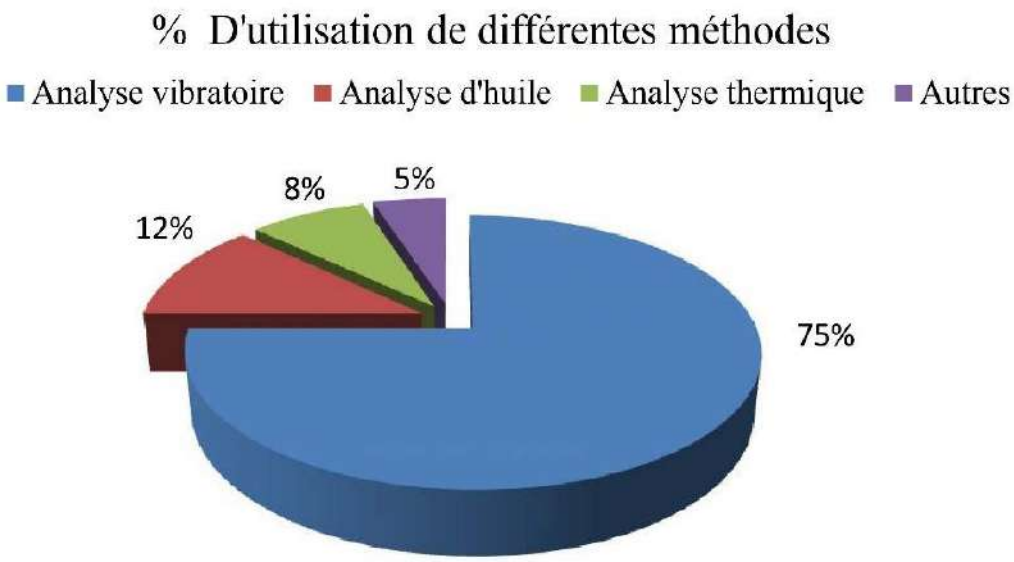


Figure I-5: Les différentes méthodes de diagnostic des machines tournantes [37].

I.4.1 Définition d'une vibration

Un système mécanique est dit en vibration lorsqu'il est animé d'un mouvement de va-et-vient rapide autour d'une position moyenne appelée « position d'équilibre », (figure I-6) [2].

La transmission du mouvement vibratoire peut se faire par deux moyens : soit par un gaz ou un fluide, généralement de l'air, dans ce cas on parle de « son » ou signal acoustique. Ou bien à travers d'un solide, comme la structure de la machine, et on parle alors de vibration dite « solidienne » ou plus couramment de signal vibratoire.

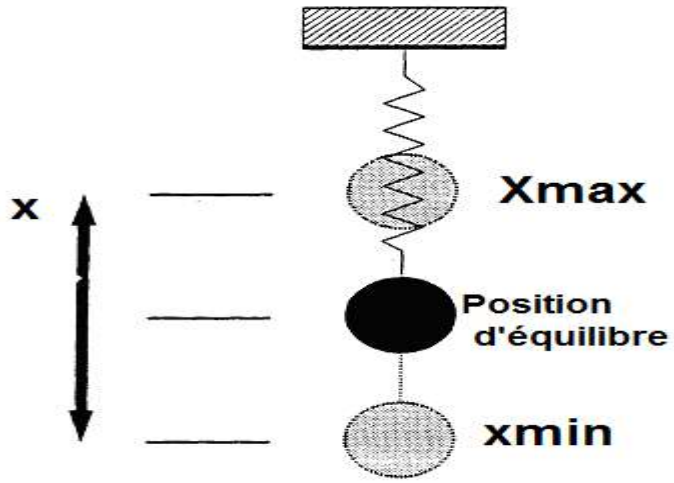


Figure I-6: Exemple d'une vibration.

I.4.2 Caractéristiques des vibrations

La vibration est caractérisée par son amplitude et sa fréquence (Figure I-7). La fréquence f indique le taux de répétition ou le nombre de récurrences d'un phénomène dans le temps imparti. Lorsque l'unité de temps sélectionnée est la seconde, la fréquence est exprimée en

Hertz (Hz) [4]. La période T est l'intervalle de temps entre deux motifs identiques, elle est Elle est liée à la fréquence en inversant la relation T = 1/ f. Nous appelons la grandeur comme La valeur par laquelle le mouvement vibratoire s'écarte de sa position d'équilibre [4], qui est habituellement exprimé par A.

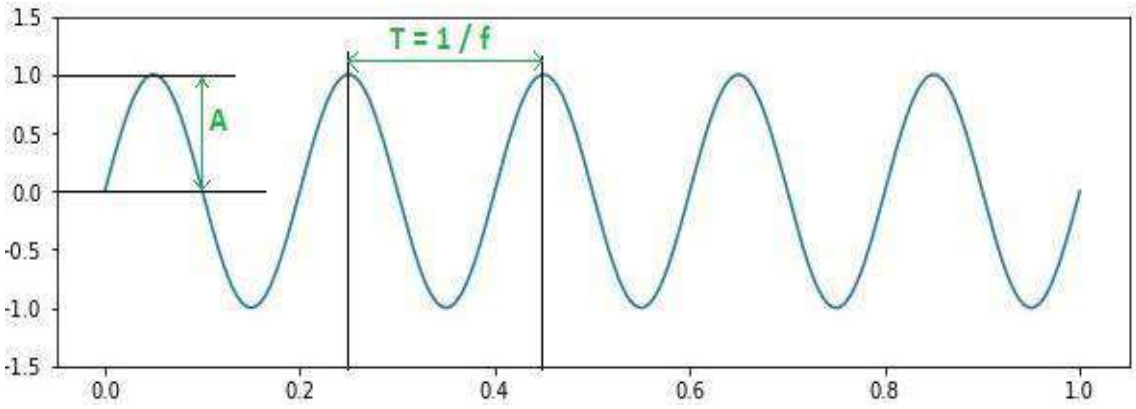


Figure I-7: Caractéristique d'une vibration (cas d'un signal sinusoïdal).

Une vibration se caractérise par trois grandeurs cinématiques : le déplacement en [µm], la vitesse en [mm/s] et l'accélération en [m/s²]. A ces grandeurs sont associées une période temporelle et une phase. Ainsi, dans le cas d'une machine tournante, une vibration élémentaire peut se mettre sous la forme :

$$x(t) = A.\ \mathrm{Sin}\ (2\pi f t + \theta) \qquad \text{Éq 1}$$

Une propriété de base relie trois représentations : déplacement D, vitesse V et accélération a :

$$D = \frac{v}{2\pi f} = \frac{a}{(2\pi f)^2} \qquad \text{Éq 2}$$

I.4.3 Analyse temporelle et analyse fréquentielle

Une méthode appropriée doit être choisie pour analyser les signaux vibratoires émis par la machine pendant son fonctionnement. Les plus couramment utilisées sont l'analyse temporelle et l'analyse fréquentielle.

- ***Analyse temporelle :***

Dans cette méthode, l'analyse ou l'interprétation du signal vibratoire est perçue à travers divers paramètres ou indicateurs [5], les plus connus sont :

- La valeur efficace du signal (RMS) :

$$RMS = \sqrt{\frac{1}{N}\sum_{i=1}^{N} {x_i}^2} \qquad \text{Éq 3}$$

- Le facteur de crête :

$$CRF = \frac{|x|_{peak}}{RMS} \quad \text{Éq 4}$$

- Le facteur d'impulsion :

$$IMF = \frac{|x|_{peak}}{\frac{1}{N}\sum_{i=1}^{N}|x_i|} \quad \text{Éq 5}$$

- Le kurtosis :

$$K = \frac{\frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})^4}{RMS^4} \quad \text{Éq 6}$$

Où :

x : valeurs successives du signal x(t) échantillonné

$|x|_{peak}$: Valeur pique du signal x

$\overline{x}$: Valeur moyenne du signal x, avec $\overline{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$

La figure (I-8) présente un exemple de signal temporel représenté par l'équation :

$$x(t) = \frac{1}{2}\sin(2.\pi.t.500) + \sin(2.\pi.t.1000) + \frac{1}{2}\sin(2.\pi.t.2000) \quad \text{Éq 7}$$

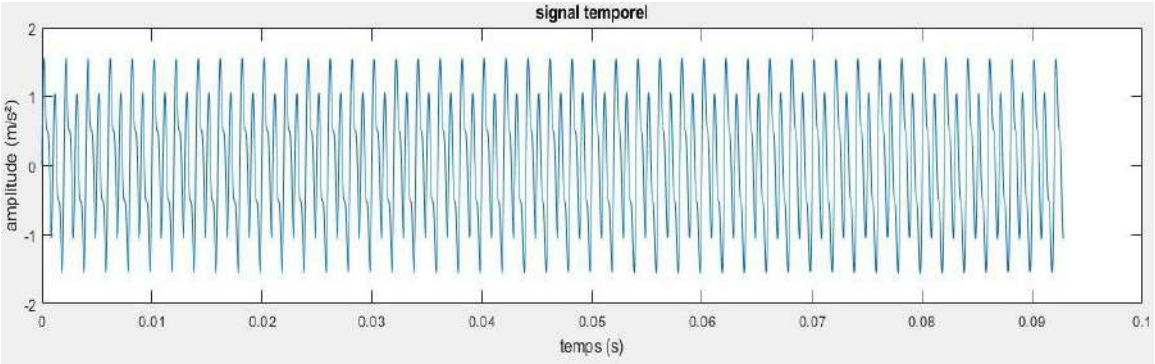


Figure I-8: Exemple de signal dans le domaine temporel.

- ***Analyse fréquentielle :***

On peut aussi représenter l'information contenue dans un signal dans le domaine fréquentiel, et qui consiste à représenter des fréquences ou des harmoniques contenues dans un signal. Ce type de représentation s'appelle un spectre. Le spectre du signal est obtenu par transformée de Fourier du signal temporel [6].

Pour le signal continu (TF) :

$$X(f) = \int_{-\infty}^{+\infty} x(t)\, e^{-2j\pi ft}.dt \quad \text{Éq 8}$$

Où :

- t : représente la variable temporelle
- f : représente la variable fréquentielle

Et dans le cas des signaux discrets (DFT) :

$$X(K) = \sum_{n=0}^{N-1} x[n].e^{-2j\pi k\frac{n}{N}}, k \in \{0,1,\dots,N-1\} \quad \text{Éq 9}$$

Où :

- N : est le nombre de points temporels (i.e. nombre d'échantillons)
- n : indice de temps
- k : indice de fréquence

En effectuant cette transformée sur le même signal précédant on obtient le spectre représenté sur la (figure I-9) :

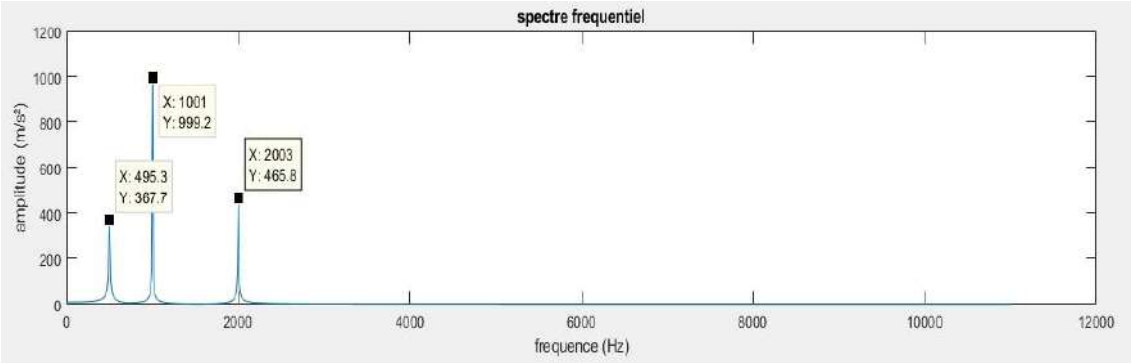


Figure I-9: Spectre d'amplitude du signal x(t) résultant du calcul de la DFT.

I.5 Les défauts des roulements

Les études statistiques montrent que dans les machines tournantes, les roulements sont les éléments qui tombent en panne le plus fréquemment. Les chercheurs ont également observé une nette augmentation des défauts de roulements en comparant deux études, l'une établie en 1999 et l'autre en 2008, (figure I-10).

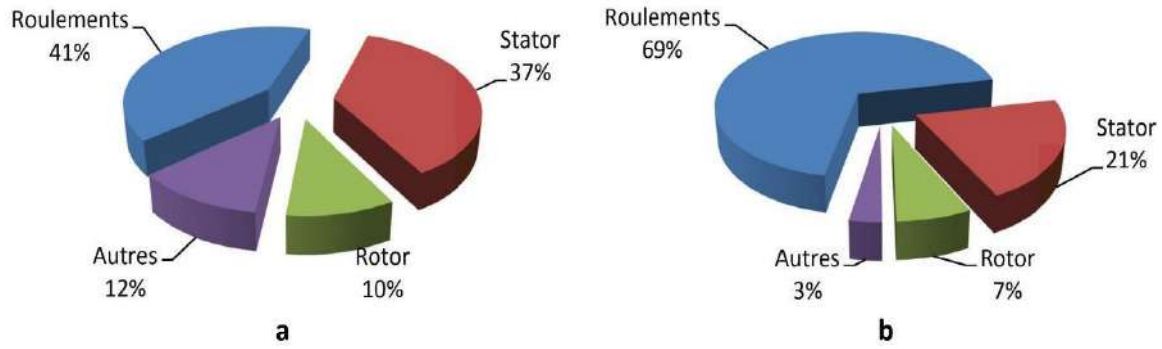


Figure I-10: Les sources de défaillances les plus rependues dans les moteurs [38].

a : étude établies en 1999, b : étude établie en 2008.

I.5.1 Structure mécanique d'un roulement

Le roulement à billes rigide, (appelé aussi roulement à billes à contact radial) est un composant mécanique formé par deux bagues concentriques appelées bagues intérieure et extérieure comportant des chemins de roulement entre lesquelles existent des corps roulants uniformément espacés par une cage. Cette cage est placée entre les bagues et a pour rôle de réduire les frottements en empêchant les éléments de se frotter les uns contre les autres (figure I-11-a\b).

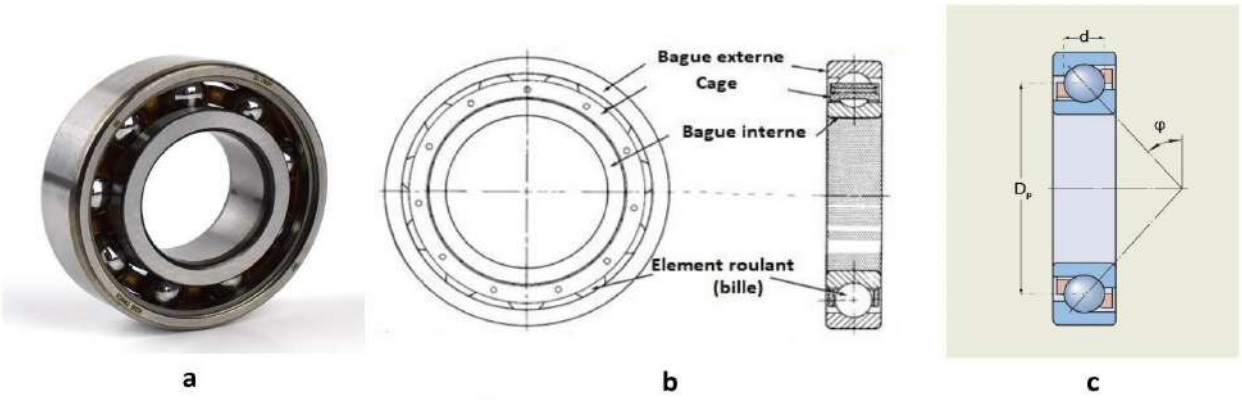


Figure I-11: Structure mécanique d'un roulement à billes rigide.

a : photographie d'un roulement réel, b : dessin technique d'un roulement et ses composants, c : caractéristiques géométriques d'un roulement.

Les dimensions et les spécifications d'un roulement à billes peuvent être visualisées sur la figure (I-11-c), où :

- d : est le diamètre de l'élément roulant (bille)
- D_p : est le diamètre moyen de rotation (moyenne des diamètres de la bague interne et externe)

- φ : est l'angle de contact

Un autre paramètre géométrique important est le nombre d'éléments qui compose le roulement à billes, et qui est généralement noté N.

I.5.2 Causes des défauts des roulements à billes

Les défauts des roulements commencent par une perte locale de matériau sur une surface mate.

I.5.2.1 Le courant électrique

Le passage du courant électrique à travers le roulement en rotation peut endommager ses différents composants (figure I-12) [7].

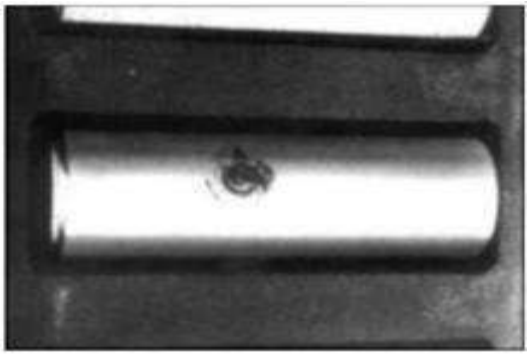


Figure I-12: Choc électrique qui a endommagé un élément roulant.

I.5.2.2 Corrosion

Apparaît sous forme de points d'oxydation sur le diamètre intérieur du roulement, le diamètre extérieur et la surface du roulement. Il apparaît après une présence prolongée d'eau ou d'humidité dans le roulement. La corrosion peut entraîner la rupture des bagues ou des éléments roulants. Pour le corriger, vous devez utiliser un lubrifiant antirouille de meilleure qualité, (figure I-13) [7].

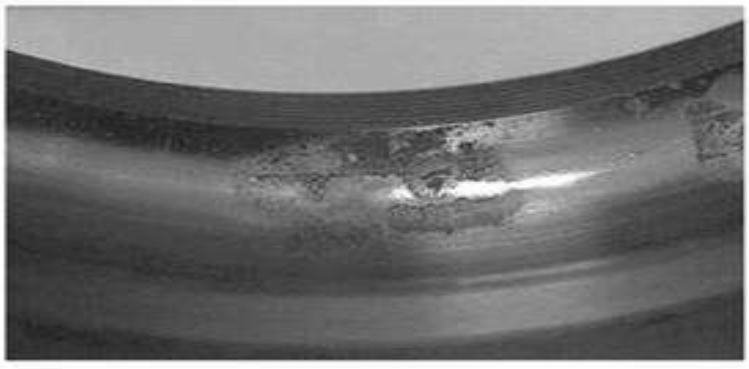


Figure I-13: Corrosion sur la bague externe d'un roulement à billes.

I.5.2.3 L'usure

L'usure provient généralement de trois raisons principales. L'usure peut provenir les particules pénètrent entre les éléments roulants et les bagues intérieure et extérieure c'est se manifeste généralement par un manque de propreté lors du montage des roulements ou si le lubrifiant est contaminé par des particules. Vous pouvez toujours vous présenter habillé due à la lumière ou au manque de lubrification. La dernière raison à cela l'usure provient de vibrations excessives atteignant le roulement sans rotation, (figure I-14) [7].



Figure I-14: Défaut d'usure sur un roulement à billes.

I.5.2.4 Bavure

Ce phénomène se produit sous forme de transfert de matière d'une surface à une autre cela peut être causé par plusieurs facteurs : manque de lubrification, pénétration dans le corps roulement dans la zone de charge, par rapport à la vitesse de rotation de la charge légère chaque cas un traitement doit être fait pour éliminer la cause, le développement de la bavure conduit à une cassure totale de la bague ou du corps roulant, (figure I-15) [7].



Figure I-15: Défaut de bavure sur un roulement à billes.

I.5.2.5 Ecaillage

Il se caractérise par des traces de fissuration et d'arrachement de fragments de matière. Il apparaît normalement à cause du vieillissement du roulement. Mais l'écaillage arrive aussi à cause d'une grande différence de température entre la bague intérieure et extérieure. Également, si le boitier qui porte le roulement possède une forme ovale, le même résultat est observé, il faut alors changer le boîtier immédiatement. D'autres causes de l'écaillage, peuvent être dues à un montage incorrect du roulement ou une liberté axiale légère qui n'est pas suffisante pour le mouvement de la dilatation, (figure I-16) [7].



Figure I-16: Défaut d'écaillage sur la bague interne d'un roulement à billes.

I.5.2.6 Dentelure

Ce défaut se produit généralement lorsque le roulement n'est pas installé correctement ou lorsque le roulement est soumis à une charge importante lorsqu'il est à l'arrêt. En dents de scie apparaît, la distance entre les fissures est égale à la distance entre les éléments roulants, (figure I-17) [7].

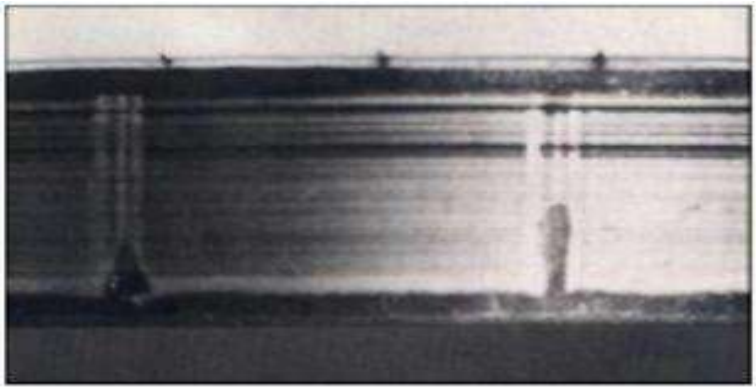


Figure I-17: Défaut de dentelure sur la bague interne d'un roulement à billes.

I.5.3 Les types de défauts

Nous différencions les types de défauts en fonction du degré d'arrachement du matériau sur le composant faire défiler.

I.5.3.1 Défaut au niveau de la bague externe

Un défaut de type écaillage affectant la bague externe d'un roulement a pour image vibratoire un peigne de raies dont le pas correspond à la fréquence du défaut (Figure I-18). Sous fortes charges dynamiques, chaque composante de ce peigne est associée à une paire de bandes latérales séparées par la fréquence de rotation [8].

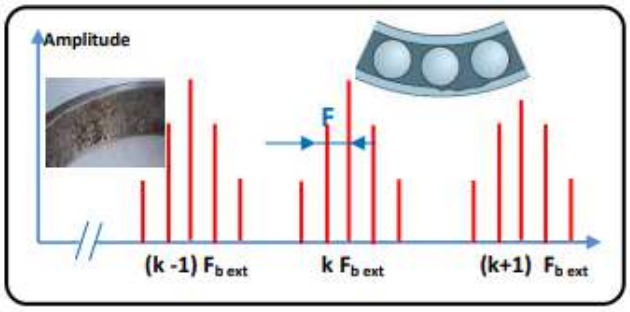


Figure I-18: Défaut au niveau de la bague externe d'un roulement avec son spectre.

I.5.3.2 Défaut au niveau de la bague interne

Un défaut de type écaillage affectant la bague interne de roulement a pour image vibratoire un peigne de raies (figure I-19).A chaque composante de ce peigne, sont associées plusieurs paires de bandes latérales espacées de la fréquence de rotation [8].

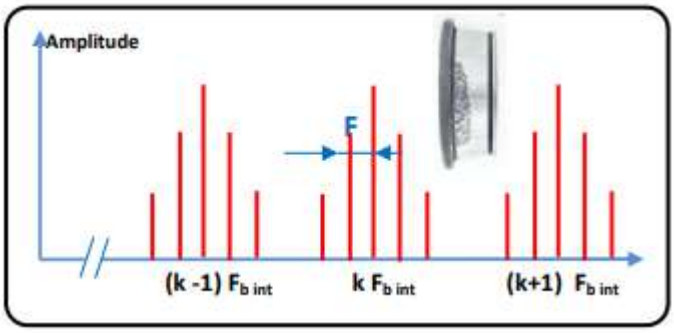


Figure I-19: Défaut au niveau de la bague externe d'un roulement avec son spectre.

I.5.3.3 Défaut au niveau de l'élément roulant

Un défaut de type écaillage sur un élément roulant (bille, rouleau ou aiguille) a pour image vibratoire un peigne de raies (figure I-20). A chaque composante de ce peigne, sont associées plusieurs paires de bandes latérales espacées de la fréquence de la cage [8].

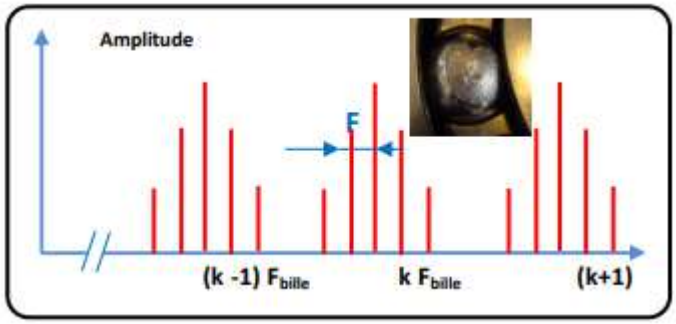


Figure I-20: Défaut au niveau de la bille d'un roulement avec son spectre.

I.5.4 Les fréquences des défauts

Le choc se produit à la fréquence correspondante donnée par les formules ci-dessous [4], ces fréquences dépendent à la fois des caractéristiques géométriques du roulement ($d, D_p, \varphi \; et \; N$) mais aussi de vitesse de rotation de l'axe qui entraîne la bague interne F_{rot}

- La fréquence de passage des billes sur la bague externe :

$$F_{b_ext} = \frac{N}{2}[1 - \frac{d\cos\varphi}{\mathrm{D_p}}]F_{rot} \qquad \text{Éq 10}$$

- La fréquence de passage des billes sur la bague interne :

$$F_{b_int} = \frac{N}{2}[1 + \frac{d\cos\varphi}{\mathrm{D_p}}]F_{rot} \qquad \text{Éq 11}$$

- La fréquence de rotation de l'élément roulant (la bille) :

$$F_{bille} = \frac{\mathrm{D_p}}{2d}[1 - \frac{d^2(\cos\varphi)^2}{\mathrm{D_p^2}}]F_{rot} \qquad \text{Éq 12}$$

Mais comme le choc dû au défaut se produit à chaque rotation de l'élément altéré, à la fois sur la bague interne et sur la bague externe, la fréquence du défaut d'élément roulant correspond généralement au double de leur fréquence de rotation sur eux-mêmes.

I.6 Traitement des signaux vibratoires

Il y a plusieurs prétraitements sur les signaux vibratoires.

Dans cette partie nous avons faire une étude et comparons quelle est la meilleure entre la transformer d'Hilbert, et l'application de notre approche.

I.6.1 Segmentation

Les méthodes d'analyse du contenu fréquentiel opèrent généralement sur des signaux stationnaires, alors que le signal vibratoire que l'on veut traiter change perpétuellement au cours du temps. La solution est donc d'effectuer l'analyse sur des trames successives de durées relativement courtes où le signal peut être considéré comme quasi-stationnaire. Les trames peuvent aussi être chevauchées pour ne pas perdre l'information contenue aux bordures (figure I-21).

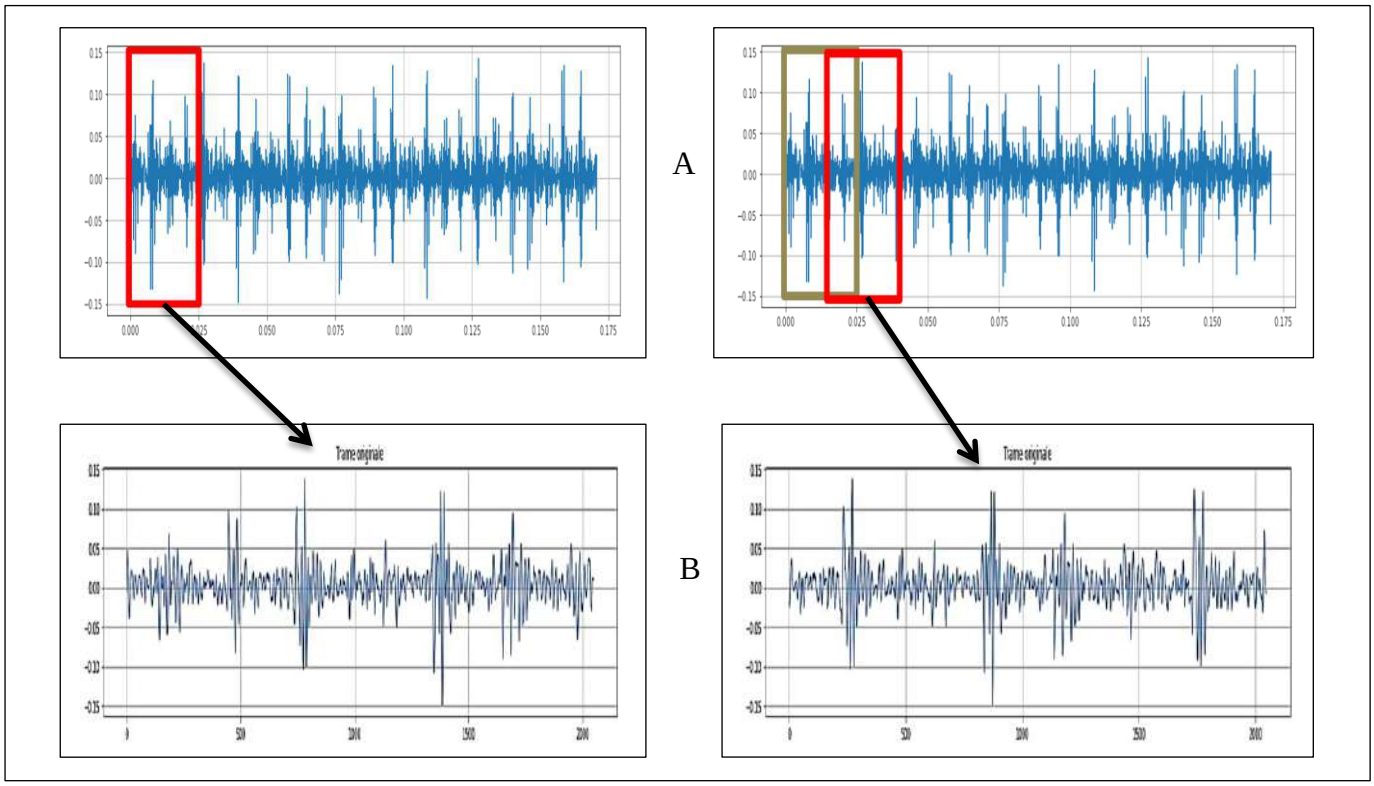


Figure I-21: exepmle de segmentation A: signal complet, B: signal segmenté.

I.6.2 Transformée de Fourier

La transformée de Fourier TF permet de représenter un signal temporel à travers ses composantes fréquentielles. La transformée de Fourier dans les cas des signaux continues est définie par :

$$X(f) = \int_{-\infty}^{+\infty} x(t) e^{-j2\pi ft}\, dt \qquad \text{Éq 13}$$

Où (f) est le spectre du signal temporel (t).

La transformée de Fourier Discrète (TFD), est utilisée quand il s'agit de calculer la transformée de Fourier d'une fonction à l'aide d'un calculateur numérique. En effet les machines ne peuvent traiter que des nombres finis car elles sont limitées par la taille de leurs

mémoires. Pour aller de la TF à la TFD il faut remplacer d'une part le signal (t) par des nombres (nt) qui représentent un échantillonnage de ce signal et d'autre part en limitant l'ensemble des nombres sur lesquels portent les calculs à une valeur finie N [9].

$$X[k] = \sum_{n=0}^{N-1} x(n) e^{-j2\pi f \frac{nk}{N}} \quad \text{Éq 14}$$

$avec\ k = 0,1,2, \dots N-1$

Le coût de calcul de la TFD est donc de N^2 multiplications complexes, il a donc une complexité de (N^2). Le nombre de multiplications augmente alors de façon exponentielle avec l'augmentation de la taille du signal à traiter. Ce nombre peut être grandement réduit en utilisant la transformée de Fourier rapide (FFT)

La FFT est simplement un algorithme permettant de réduire le nombre d'opérations, en particulier le nombre de multiplications. Pour calculer la TFD, il existe différents algorithmes de transformée de Fourier rapide. Le plus connu et le plus utilisé est celui de Cooley-Tukey (appelé aussi « à entrelacement temporel »). Le nombre des multiplications complexes réduit par cet algorithme ce $(N + N\ log_2 N)$, ce qui réduit la complexité à $(N\ log_2 N)$.

Le (tableau I-1) montre l'amélioration des temps de calcul d'une DFT par utilisation d'un algorithme de FFT.

Nombre d'échantillons	Nombre de multiplications DFT	Nombre de multiplications FFT	Rapport du nombre de multiplications
8	64	24	2.5
128	16384	896	18.95
512	262144	4608	56.88
1024	1048578	10240	102.4
2048	4194304	22528	186.18

Tableau I-1: Comparaison entre la DFT et la FFT en terme de complexité de calcul.

I.6.3 Définition de TEO

L'algorithme de séparation d'énergie utilise TEO pour séparer $x(t)$ dans son enveloppe d'amplitude $|a(t)|$ et le signal IF $f(t)$ pour accomplir la démodulation du signal AM-FM mono composant [10] :

$$f(t) = \frac{1}{2\pi} \sqrt{\frac{\psi[x'(t)]}{\psi[x(t)]}} \quad \text{Éq 15}$$

I.6.4 Historique de la TEO

En 1983, Herbet M. Teager et Shushan M. Teager ont publié l'article suivant : « Phenomenological Model For Vowel Production in the Vocal Tract ». Estime que ce modèle de parole actuel (à l'époque) principalement cette modélisation utilisant la théorie des filtres linéaires [11].

En 1990 il décrit un autre modèle processus non linéaire et publié l'article suivant : « Evidence of Nonlinear Sound Production Mechanisms in the Vocal Tract » [12].

En 1990 aussi JF Kaiser a été attiré par la méthode de calcul d'énergie du générateur de son et publié son article intitulé « A Simple Algorithm for Calculating the Energy of a signal» [13].

La première application des opérateurs dans « Speech Nonlinearity, Modulation and Energy operators » présenté par Petros Maragos, Thomas F et Kaiser, c'est la première technique de démodulation AM-FM [14].

Il y a plusieurs articles publiés sur la TEO opérateur énergétique de Teager lors de la conférence internationale sur l'acoustique, la parole et le traitement du signal ICASSP.

Le premier article de kaiser a été soumis à l'ICASSP-90 et deux autres ont été publiés en 1992 et au moins sept autres articles publiés en 1993.

L'article de l'ICASSP-91 c'est nouveau type de filtrage est un filtre non linéaire pour l'amélioration des images [15].

$$|a(t)| = \frac{\psi[x(t)]}{\sqrt{\psi[x(t)]}} \qquad \text{Éq 16}$$

Quand on a aussi :

$$\psi[x(t)] = [x''(t)]^2 - x'(t)x'''(t) \qquad \text{Éq 17}$$

I.6.5 Notion de Base

Les méthodes de traitement des signaux continus et des signaux discrets sont différentes, pour cela on utilise la notation suivante :

- Les signaux continus s'écrivent $x(t)$.
- Les signaux discrets s'écrivent $x[n]$.

Selon la complexité de l'expression.

Cela devrait également ressortir clairement du contexte dans la plupart des cas.

x' : C'est la première dérivée utilisée pour désigner $\frac{dx}{dt}$

$$\frac{dx}{dt} = x[n] - x[n-1] \quad \text{Éq 18}$$

$$\frac{dx}{dn} = x[n+1] - x[n] \quad \text{Éq 19}$$

Car on ne peut pas garantir que l'une des deux est plus correct que l'autre.

x'' : C'est la deuxième dérivée utilisée pour désigner $\frac{d^2x}{dt^2}$

$$x'' = x[n-1] - 2x[n] + x[n+1] \quad \text{Éq 20}$$

I.7 Transformer d'Hilbert

L'enveloppe de signal donne une représentation plus simple et claire. Dans cette partie, une technique pour trouver l'enveloppe d'un signal est décrite. Le concept est similaire à la détection d'enveloppe utilisée dans la démodulation d'amplitude. L'enveloppe d'un signal est considérée comme une information transportée par le signal porteur.

Dans cette étude, nous utilisons une technique d'extraction d'enveloppe parmi les plus représentative, à savoir la transformée de l'enveloppe de Hilbert [16] qui sont fréquemment utilisées pour extraire les courbes d'enveloppe sur les signaux.

I.8 Conclusion

Dans ce chapitre, nous avons montré différentes stratégies de maintenance Adopté dans l'industrie, nous avons présenté également l'utilisation de l'analyse vibratoire et leur rôle dans la détection des défauts des machines tournantes. Enfin, nous avons vu comment utiliser cette approche pour se concentrer sur le diagnostic des défauts de roulements.

Finalement nous avons présenté des différents types de prétraitement sur les signaux vibratoires.

Chapitre II: Technique d’intelligence artificiel

II.1 Introduction

Les données traitées par les machines informatiques sont aujourd'hui de plus en plus complexes, et il est difficile voire impossible d'écrire manuellement des programmes informatiques capables d'extraire des informations utiles de ces données. L'avènement des méthodes d'apprentissage automatique ML et d'apprentissage profond DL ouvre la possibilité d'éduquer les ordinateurs à partir d'un grand nombre d'exemples et d'appliquer les connaissances qu'ils acquièrent à des domaines très spécifiques.

Dans ce chapitre nous allons parler sur les différents réseaux neurones en général, ensuit nous allons décrit les réseaux CNNs et ANNs en détail.

En fin nous allons définir les différentes couches CNN à une dimension avec des organigrammes explicatifs.

II.2 Deep Learning

Deep Learning est un sous-domaine de machine Learning qui à son tour est un sous-domaine de l'intelligence artificielle (Figure II-1). L'objectif principal de l'intelligence artificielle est de fournir un ensemble d'algorithmes et de techniques qui peuvent être utilisés pour résoudre les problèmes auxquels sont confrontés les humains. Intuitivement et efficacement, presque automatiquement [17].

Le Deep Learning ou DL est une branche du Machine Learning entièrement basée sur les Réseaux de Neurones Artificiels.

Le concept de Deep Learning existe depuis quelques années, mais a été abandonné faute de moyens nécessaires.

Au milieu des années 2000, le machine Learning battait son plein dans les concours de reconnaissance visuelle, en 2012 deep mind une startup dans le domaine de l'IA gagnait gros avec des algorithmes de Deep Learning, et l'année suivante, tous les concurrents se tournaient vers le Deep Learning.

Les avancées du DL sont dues à la croissance exponentielle des machines en puissance de calcul et de stockage, et à la disponibilité de quantités massives de données (big data), ses 3 éléments sont nécessaires pour développer des DL potentiels qui n'étaient pas envisageables dans les années 90 [18].

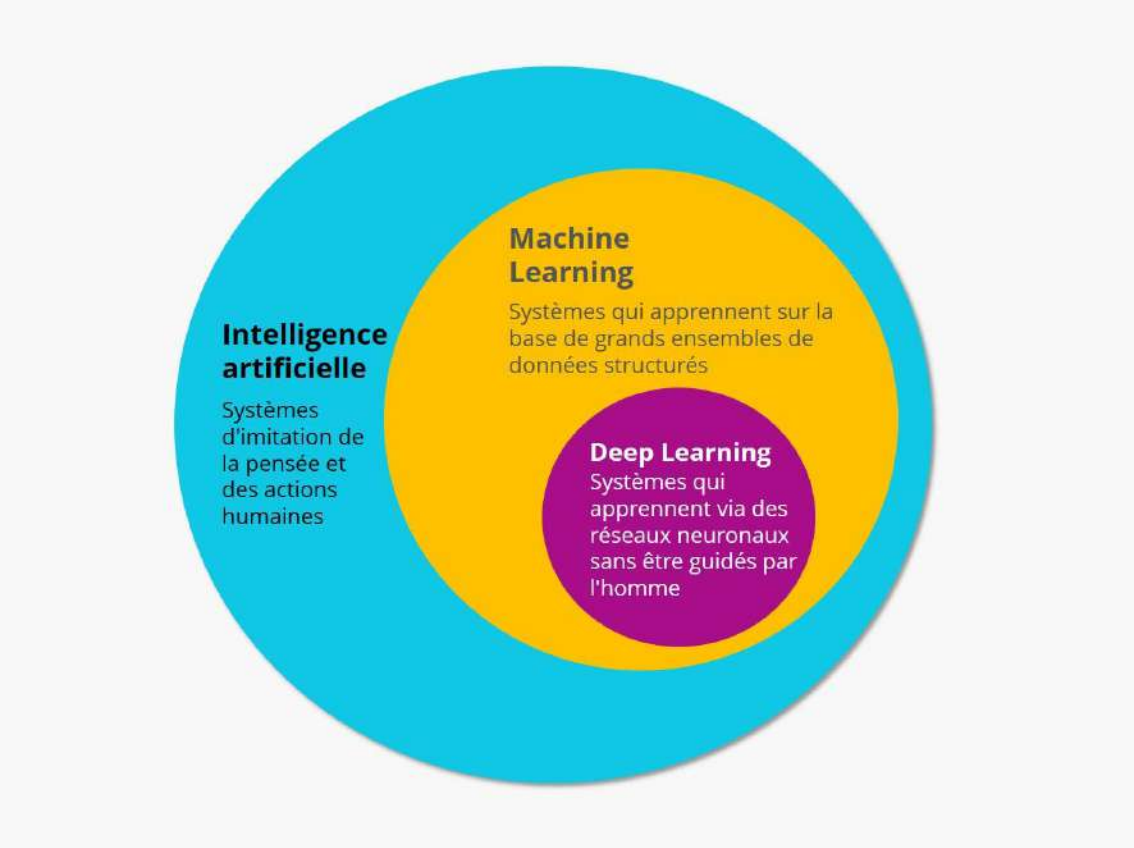


Figure II-1: Les domaines de l'intelligence artificielle.

II.3 Détection des anomalies

La détection des anomalies dans la donnée est une habilité importante à apprendre aux machines. Les hommes peuvent détecter les anomalies présentent dans leurs environnements pour se protéger d'un possible danger ou pour découvrir de nouvelles connaissances. La détection des anomalies et leurs classifications est appliquée aujourd'hui dans le domaine de l'IA pour détecter des fraudes de cartes de crédit, des cybers intrusions ou bien dans la maintenance prédictive des équipements industriels. Cette tâche est une variante de la classification où l'objectif est de dire si l'information présentée à la machine relève d'une situation ordinaire ou bien d'une situation irrégulière et peut même aller plus loin en nous informant sur la nature de cette irrégularité [19].

II.4 Type d'apprentissage

Les algorithmes de Deep Learning se répartissent en deux types principaux, apprentissage supervisé et apprentissage non-supervisé [17].

II.4.1 Apprentissage supervisé

L'apprentissage supervisé des données étiquetées (possédant des étiquettes). Ces étiquettes nous renseignent sur le résultat souhaité et utilisent leurs données pour créer un modèle qui peut faire des prédictions qui se généralisent à de nouvelles données (figure II-2).

II.4.2 Apprentissage non-supervisé

L'apprentissage non supervisé est un algorithme d'apprentissage automatique utilisé pour faire des inférences à partir d'un ensemble de données composé de données d'entrée sans réponses étiquetées (figure II-2).

Ce type d'apprentissage est principalement utilisé lorsque l'on souhaite étudier un ensemble de données non étiquetées.

II.4.3 Différence entre apprentissage supervisé et non supervisé

La principale différence entre ces deux types d'apprentissage est la disposition des données d'entrée, mais d'autres différences peuvent être trouvées dans le tableau ci-dessous [18].

Le tableau suivant résume la différence entre l'apprentissage supervisé et non supervisé :

	L'apprentissage supervisé	**L'apprentissage non supervisé**
Les données d'entrée	Utilise des données connues et étiquetées en tant qu'entrée	Utilise des donnes inconnues qu'entrée non étiquetées
Nombre des classes	Connues	Inconnues
Précision des résultats	Résultats précis et fiables	Précision et fiabilité modère

Tableau II-1: déférence entre apprentissage supervisé et non supervisé.

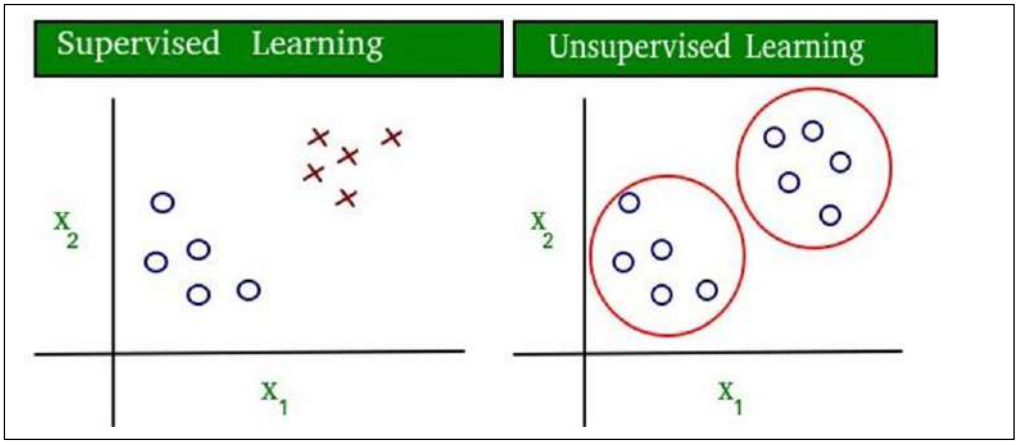


Figure II-2: exemple explicatif de la déférence entre apprentissage.

II.5 Réseau Neurones

II.5.1 Neurone biologique

Comme beaucoup d'invention ont été inspirées par la nature, les réseaux de neurones artificielles ANN inspirés par l'architecture du cerveau humain ou neurone biologique. Notre cerveau est composé d'environ 10 milliards de neurones, chacun connecté à environ 10 000 autres neurones, le neurone est composé de corps cellulaire contenant le noyau, et de nombreuses extensions ramifiées appelées dendrites, plus une très longue extension appelée axone. À l'extrémité de l'axone se trouvent de minuscules structures appelées terminaisons de l'axone, qui sont connectées aux dendrites des autres neurones (figure II-3) [20].

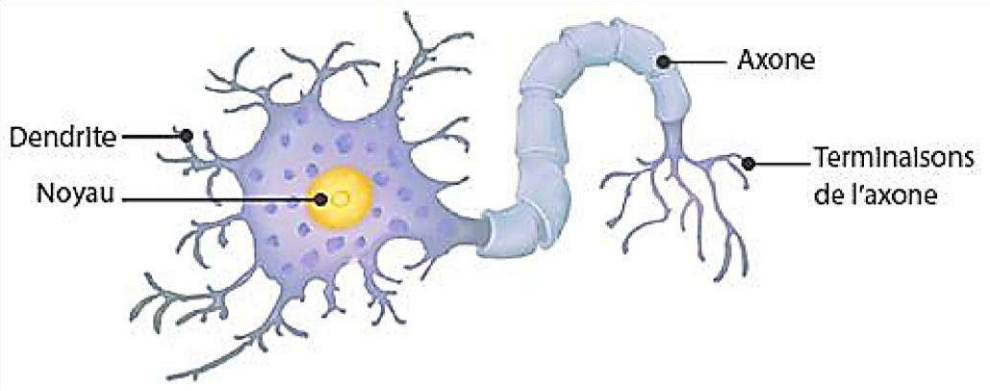


Figure II-3: Structure d'un neurone biologique.

Chaque neurone reçoit dans ses dendrites des intrants électrochimiques d'autres neurones. Si ces entrées électriques sont suffisamment puissantes pour activer le neurone, le neurone est activé et transmet le signal sur son axone, et il le transmet ensuite aux dendrites des autres neurones. Ces neurones attachés peuvent aussi se déclencher, ce qui prolonge le processus de transmission du message [17].

II.5.2 Architecture des réseaux de neurones artificiels

Les réseaux neuronaux sont des structures complexes constituées de neurones artificiels qui prennent de multiples entrées pour produire une ou plusieurs sorties (figure II-4) [18].

Il se compose de :

- **Couche Entrée (Input Layer) :** c'est l'ensemble de neurones qui porte le signal d'entré du réseau, et par la suite tous les neurones de cette couche sont reliés à la couche suivante.
- **Couche cachée (Hidden Layer) :** elles peuvent être une ou plusieurs, c'est ici où les relations entre les variables vont être mises en exergue. Le choix du nombre de couches et de neurones est intuitif et nécessite de l'expérience venant de l'expert.

- **Couche sortie (Output Layer) :** elle représente le résultat du réseau de neurones c'est ce qu'on appelle la prédiction.

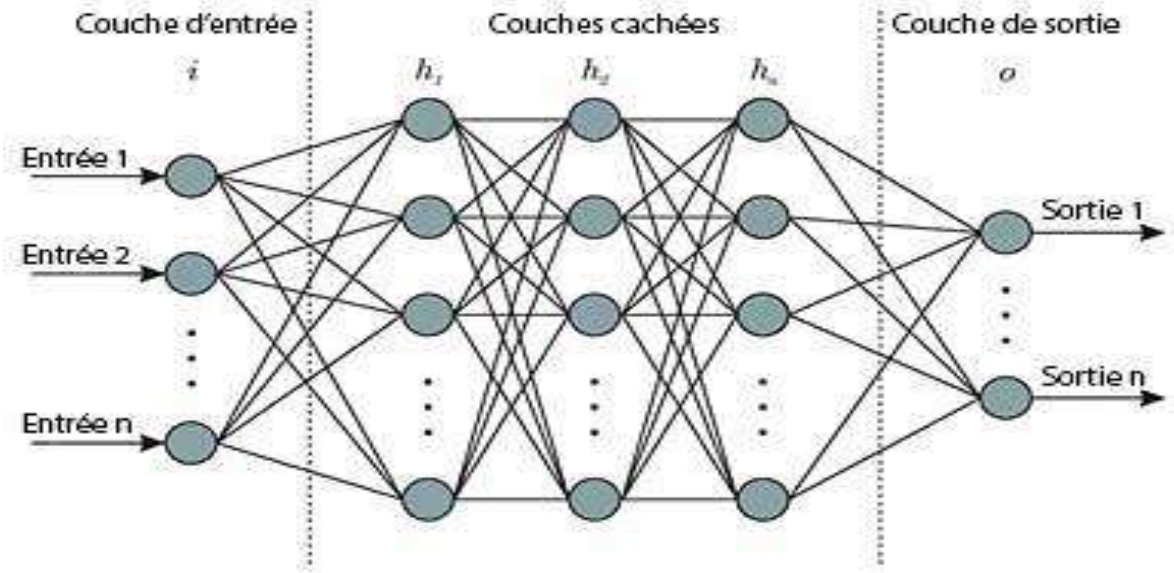


Figure II-4: Architecture d'un réseau de neurones artificiels.

II.5.3 Le perceptron

Le perceptron est une unité primaire dans un réseau de neurones artificiels, c'est un neurone artificiel qui accomplit une fonction algébrique non linéaire. Le neurone artificiel reçoit une ou plusieurs entrées $\mathbf{x_i}$, et chaque entrée est pondérée séparément par $\mathbf{w_i}$, la somme des entrées est ajoutée au biais de neurone **b**, puis utilisé comme argument d'une fonction d'activation **f** (figure II-5). La prédiction avec un réseau de neurones se fait en appliquant les données comme entrées à la première couche, puis en calculant toutes les couches suivantes jusqu'aux sorties finales. C'est ce qu'on appelle souvent la propagation vers l'avant [17].

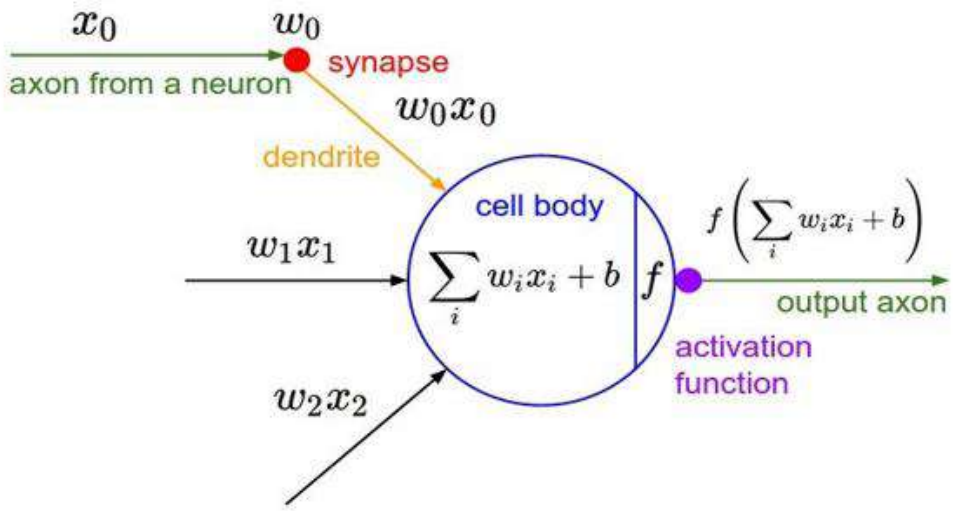


Figure II-5: Neurone artificiel simple 'Le perceptron' qui prend une somme pondérée à l'entrée et détermine si la sortie est active ou non.

II.5.4 Fonction d'activation

Afin de pouvoir exprimer des relations non linéaires entre l'entrée et la sortie, des fonctions d'activation sont appliquées. Lorsque la non-linéarité est utilisée, un réseau de neurones devient un approximateur de fonction universelle. La sortie du neurone n'a pas de limite, elle varie de $-\infty$ à $+\infty$, le neurone ne connaît donc pas vraiment de limites. Pour dire si un neurone est excité ou non, on introduit une fonction d'activation [17].

II.5.4.1 La fonction seuil (échelon unité)

C'est la fonction d'activation la plus simple, son équation est donnée par :

$$f(x) = \begin{cases} 1 \; si \; x \geq 0 \\ 0 \; si \; x < 0 \end{cases} \qquad \text{Éq 13}$$

Comme on peut le voir dans l'équation ci-dessus, si la valeur de la somme pondérée est supérieure à 0 la sortie est 1, sinon la sortie est 0.

II.5.4.2 La fonction sigmoïde

Une fonction d'activation plus couramment utilisée dans l'histoire de la littérature NN est la fonction sigmoïde, qui suit l'équation :

$$f(x) = \frac{1}{1 + e^{-x}} \qquad \text{Éq 14}$$

L'avantage de cette fonction d'activation est quelle est différentiable en tous point. La sortie de cette fonction d'activation sera en comparaison de la fonction linéaire, toujours dans l'intervalle [0,1].

II.5.4.3 La fonction Tangente Hyperbolique

Une autre fonction d'activation qui est utilisée est la fonction tangente hyperbolique (Tanh), Son équation est donnée par :

$$f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \quad \text{Éq 15}$$

La fonction Tangente hyperbolique est centrée sur zéro, donc la sortie de la fonction sera toujours comprise dans l'intervalle [-1, 1].

II.5.4.4 La fonction ReLU

Cette fonction d'activation Rectified Linear Unit (ReLU) est définie comme suit :

$$f(x) = \begin{cases} \text{x } si\ x\ \geq 0 \\ 0\ si\ x < 0 \end{cases} \quad \text{Éq 16}$$

ReLU est également appelé fonction rampe, en raison de sa représentation graphique. La sortie de la fonction est nulle pour les entrées négatives, mais augmente linéairement pour les valeurs positives. Empiriquement, la fonction d'activation ReLU a tendance à surpasser à la fois les fonctions sigmoïde et Tanh dans presque toutes les applications [22].

II.5.4.5 La fonction Leaky ReLU

Une variante de ReLU, appelée Leaky ReLU, elle est représentée par la forme algébrique suivante :

$$f(x) = \begin{cases} \text{x } si\ x\ \geq 0 \\ \text{ax } si\ x < 0 \end{cases} \quad \text{Éq 17}$$

D'après la formule ci-dessus, et la représentation graphique de cette fonction, nous pouvons voir que la fonction est bien autorisée à prendre une valeur négative, contrairement à ReLU qui met la sortie de la fonction à une valeur nulle.

II.5.4.6 La fonction Unité exponentielle linéaire (ELU)

La fonction Unité Exponentielle Linéaire (ELU) est définie comme suit :

$$f(x) = \begin{cases} \text{x} \quad si\ x\ \geq 0 \\ \text{a}(e^{x} - 1)\ si\ \ x < 0 \end{cases} \quad \text{Éq 18}$$

L'hyper-paramètre ELU α contrôle la valeur à laquelle la fonction ELU sature pour les entrées négatives. ELU obtient souvent une précision de classification plus élevée que ReLU

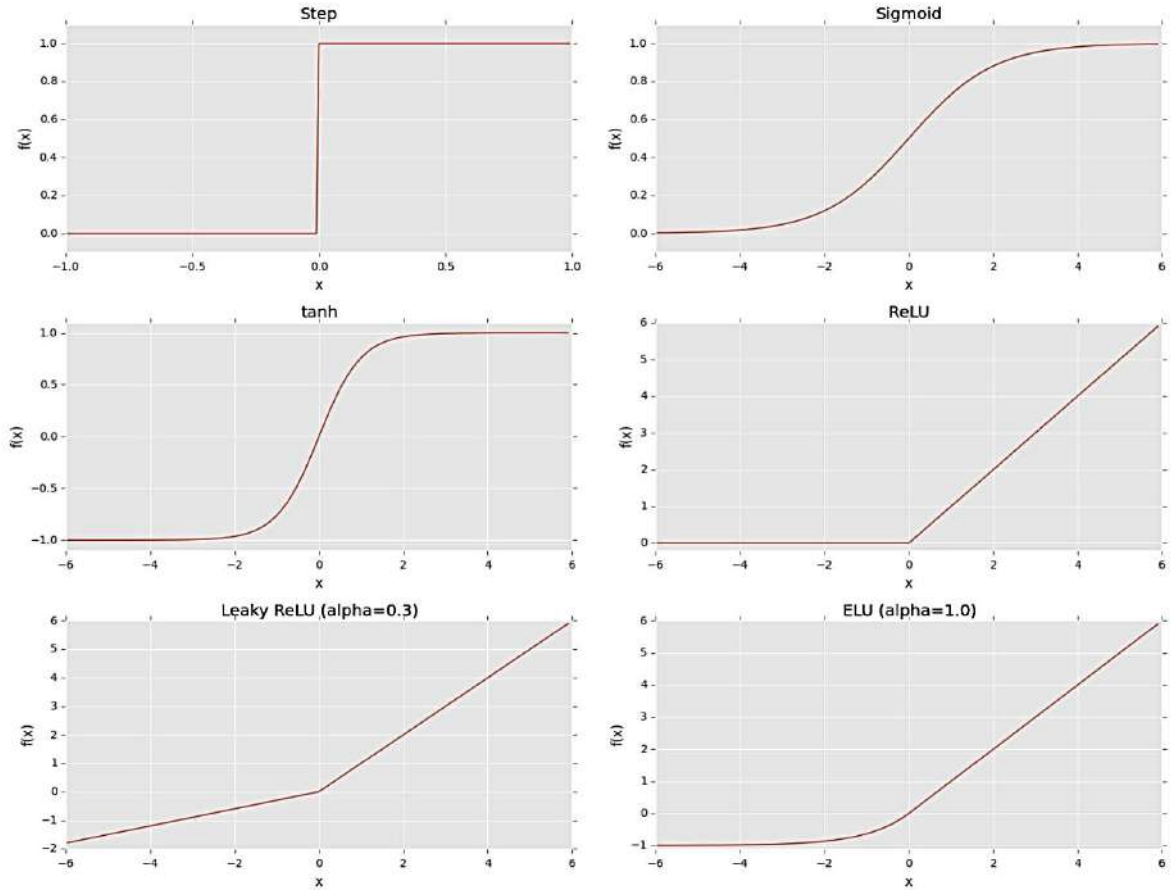


Figure II-6: Forme des fonctions d'activations les plus utilisées.

II.6 Types de réseaux de neurones

Plusieurs architectures de réseaux de neurones sont utilisées dans la pratique. Les plus exploitées sont :

- Les réseaux de neurones artificiels (ANN)
- Les réseaux de neurones à convolution (CNN)

II.6.1 Réseaux de neurones artificiels (ANN)

Les réseaux de neurones artificiels sont des réseaux fortement connectés de processeurs élémentaires fonctionnant en parallèle. Chaque processeur élémentaire calcule une sortie unique sur la base des informations qu'il reçoit. Toute structure hiérarchique de réseaux est évidemment un réseau.

Un ANN se compose de 3 types de couches, une couche d'entrée, des couches cachées et une couche de sortie.

II.6.2 Réseau de neurones à convolution (CNN)

Un réseau neuronal convolutionnel où CNN est une extension des réseaux de feed forward traditionnels (FFN) dans le cadre de l'inspiration des facteurs biologiques [23]. Ceux-ci ont été initialement étudiés pour le traitement d'images dans lesquelles des motifs

répétitifs peuvent être trouvés - par exemple, une image avec des bords répétitifs et d'autres motifs. Les CNNs surpassent tous les autres algorithmes ML classiques et fait un grand succès dans les tâches de traitement de vision par ordinateur (Computer Vision Tasks), ils ont des larges applications dans le traitement d'images et vidéo, le traitement du langage naturel (NLP), les systèmes de recommandation etc.

Les réseaux convolutifs sont particulièrement efficaces grâce à plusieurs types de couches spéciales : des couches de convolution, des couches groupement (Pooling) et de couches entièrement connectées [24], la figure (II-7) illustre un modèle d'un réseau conventionnel unidimensionnel (1D CNN).

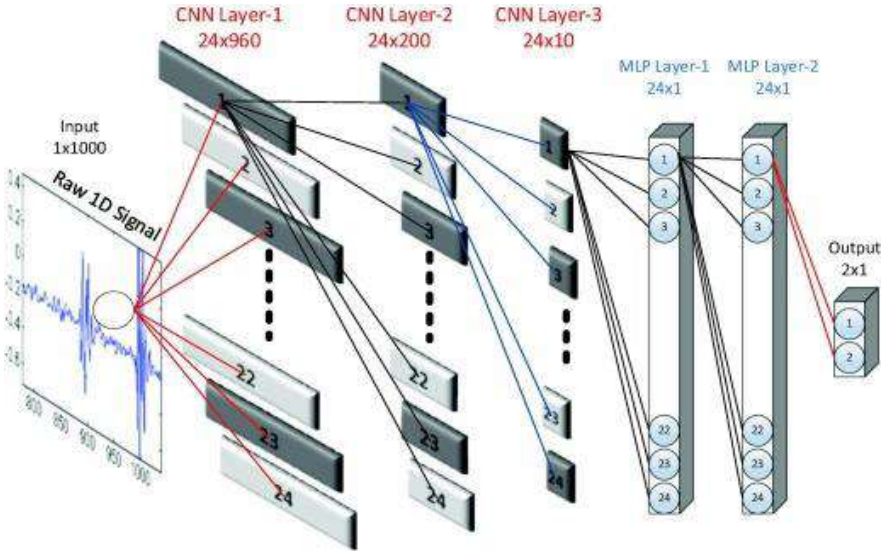


Figure II-7: Un exemple de configuration CNN 1D avec 3 couches CNN et 2 couches MLP.

Il existe de nombreux types de couches utilisées pour constituer des réseaux de neurones convolutifs, mais les plus utilisées sont: convolution, fonction activation, pooling, entièrement connectée et dropout.

II.6.2.1 Couche de convolution (CONV)

Les couches CONV sont les éléments de base des réseaux de neurones convolutifs.

Les paramètres de la couche CONV consistent en un ensemble de filtres (noyaux) qui peuvent appris, où chaque filtre a une largeur et une hauteur. Dans la plupart des applications, le noyau est un vecteur de côtés impairs.

Pour effectuer la convolution, un filtre de taille fixe balaye un vecteur 1D. Pour chaque position, multipliez les données d'entrée par les poids du filtre, ensuite, nous les ajoutons pour obtenir une valeur unique, qui devient la valeur de sortie (figure II-8).

Il convient de mentionner qu'une couche de convolution est spécifiée par les hyperparamètres suivants :

- La taille du filtre.
- Le nombre de filtre.
- Le pas que les filtres effectuent durant le balayage (Stride).

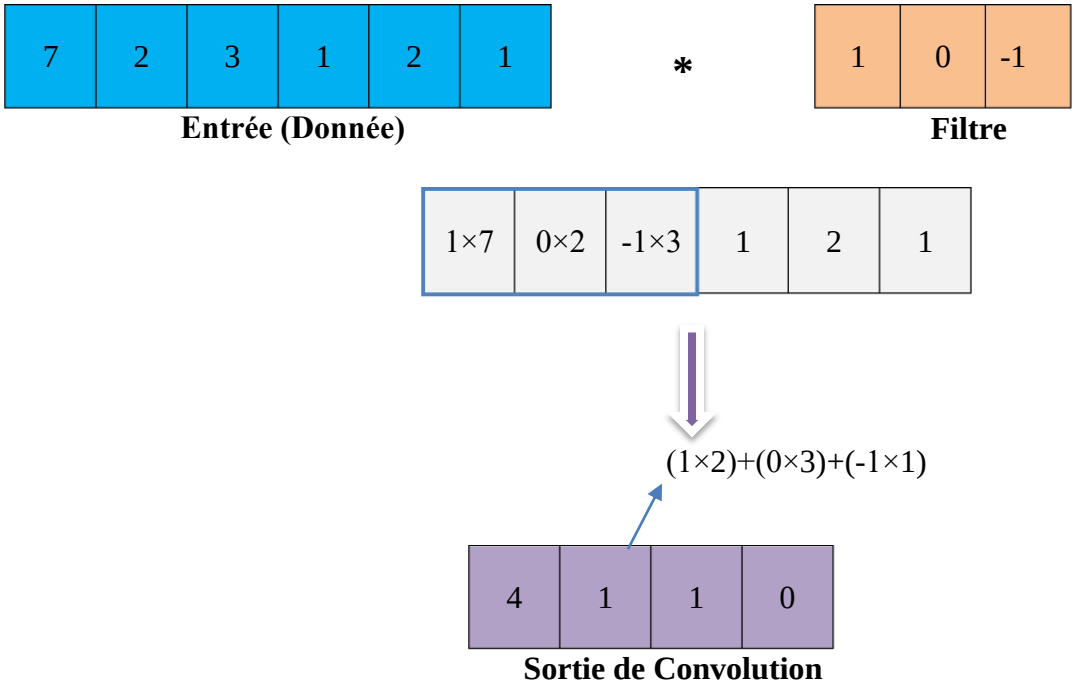


Figure II-8: Opération de convolution.

II.6.2.2 Remplissage zéro (Padding)

C'est une technique pour résoudre le problème de cette convolution qui ne permet pas au centre de chaque filtre de chevaucher les éléments les plus externes de données d'entrée, donc pour résoudre ce problème on utilise cette technique de remplissage zéro, où des lignes et des colonnes de zéros sont ajoutées de chaque côté de la donnée d'entrée, de manière pour ajuster le centre de filtre sur l'élément le plus externe et à conserver la même dimension dans le plan grâce à l'opération de de convolution [25].Les architectures de CNN modernes utilisent généralement :

Zéro rembourrage : cette méthode utiliser pour conserver les dimensions dans le plan afin d'appliquer plus de couches.

Si n'y a pas de remplissage zéro (nul) (padding), chaque carte d'entités (caractéristique) successive deviendrait plus petite après l'opération de convolution (Figure II-9)

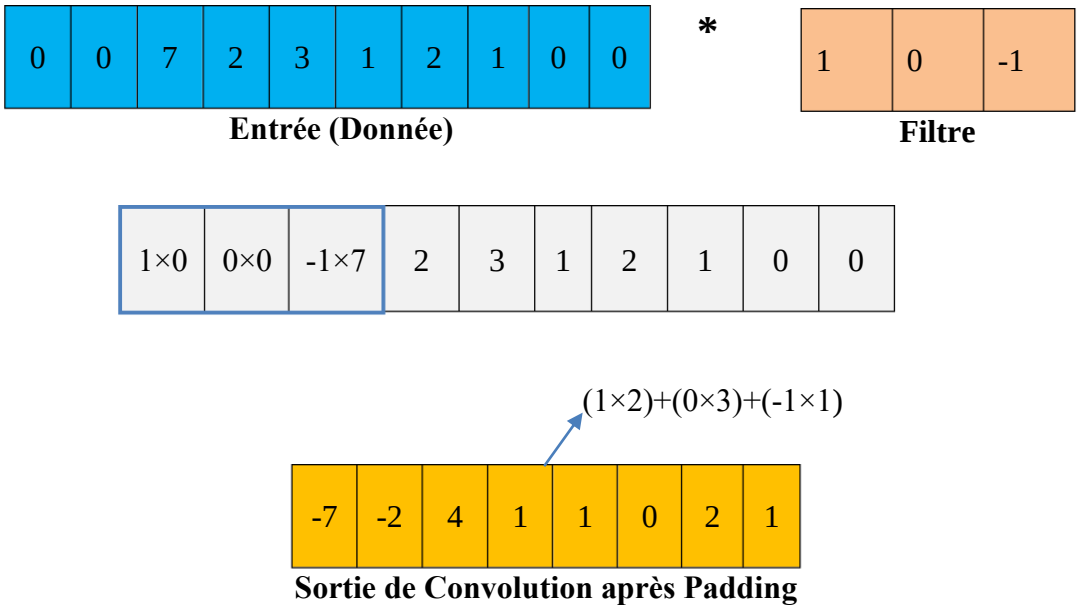


Figure II-9: opération convolution avec padding.

II.6.2.3 Couche de correction (ReLU)

Après chaque couche CONV dans un CNN, il existe une fonction d'activation non linéaire tel que ce soit pour appliquer ReLU, ELU ou Leakey ReLU [17]. Nous spécifions généralement des couches La correction est effectuée par ReLU dans l'architecture réseau car la fonction d'activation de ReLU est Plus utilisée. La couche de correction ReLU remplace toutes les valeurs négatives avec zéro en entrée (figure II-10).

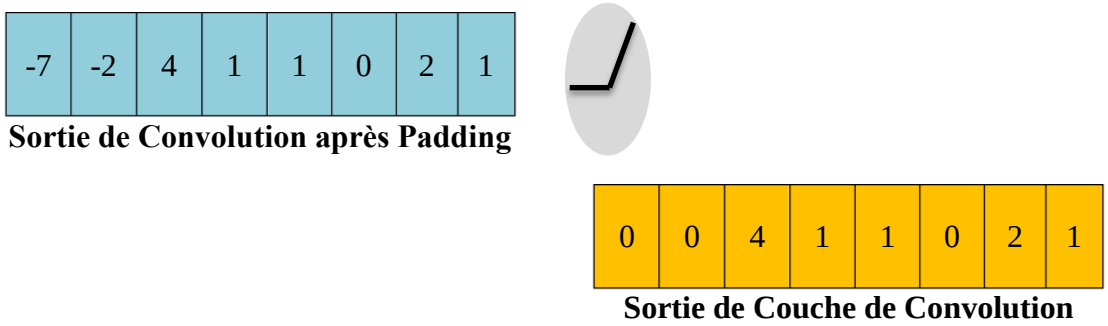


Figure II-10: Application de la fonction d'activation ReLU.

II.6.2.4 Couche de Pooling (POOL)

La couche de regroupement fournit une opération de sous-échantillonnage typique qui réduit la dimensionnalité dans le plan des cartes de caractéristiques pour introduire une

invariance de translation à de petits déplacements et distorsions, et réduire la quantité de formation ultérieure.

Dans l'une des couches de regroupement, la taille du filtre, la foulée et padding sont des hyper paramètres dans les opérations de mise en commun, similaires à opérations de convolution.

- ***Max Pooling***

La forme la plus populaire d'opération de pooling est le max pooling, qui extrait les patchs des cartes de caractéristiques d'entrée, produit la valeur maximale dans chaque patch, et rejette toutes les autres valeurs. Un max pooling avec un filtre de taille 2×2 avec une stride de deux est couramment utilisé dans la pratique. Cela sous-échantillonne la dimension en plan des cartes des caractéristiques par un facteur de deux. Contrairement à la hauteur et à la largeur [26], la dimension en profondeur des cartes d'entités ré-secteur inchangé (figure II-11).

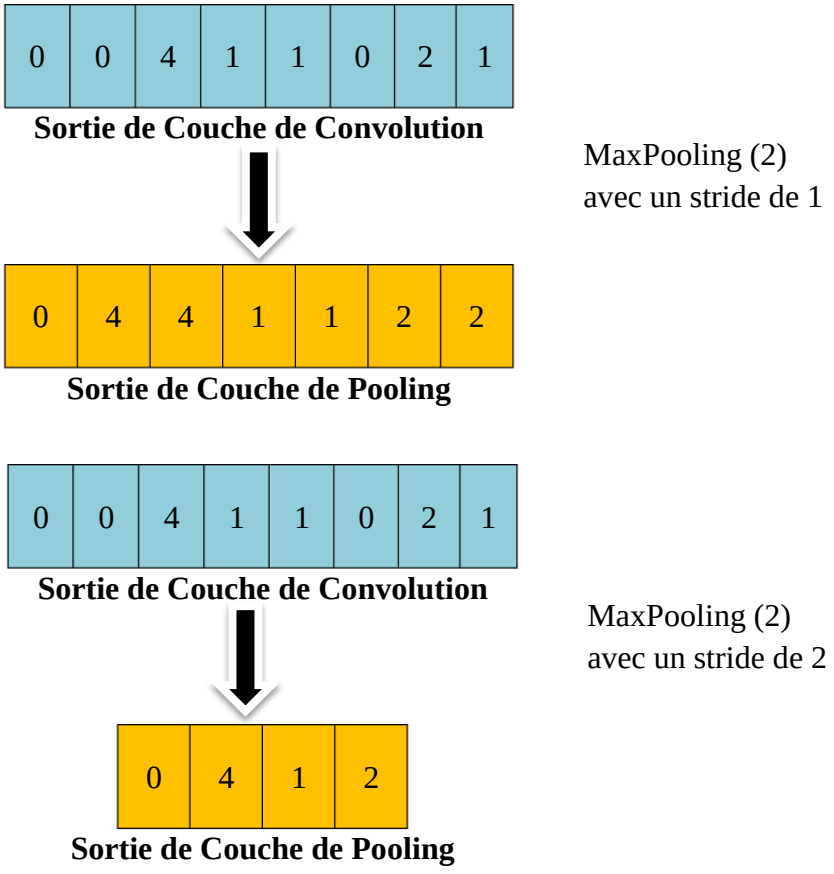


Figure II-11: Application de MaxPooling.

- ***Global Average Pooling***

Une autre opération de mutualisation digne de mention est la mutualisation moyenne mondiale [27]. Un regroupement global moyen effectue un type extrême de sous-échantillonnage, où une carte d'entités avec une taille de largeur de hauteur est réduite en un tableau de 1×1 en prenant simplement la moyenne de tous les éléments de chaque carte d'entités, tandis que la profondeur des cartes d'entités est conservée. Cette opération n'est généralement appliquée qu'une seule fois avant les couches entièrement connectées (figure II-12).

Les avantages de l'application de la mise en commun moyenne globale sont les suivants :

- Réduit le nombre de paramètres apprenables .
- Permet au CNN d'accepter des entrées de taille variable.

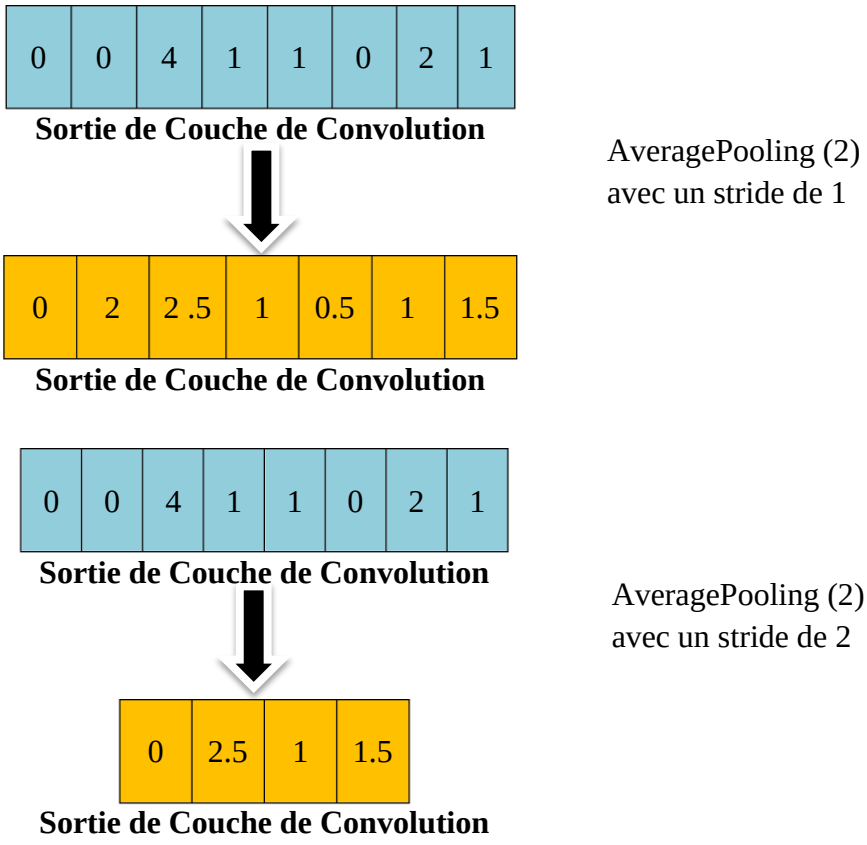


Figure II-12: Application d'AveragePooling.

II.6.2.5 Couche entièrement connectée (FC)

Les neurones de la couche FC sont entièrement connectés à toutes les sorties des fonctions d'activation de la couche précédente, tout comme un réseau de neurones ANN.

Les couches FC sont toujours placées à la fin du réseau, c'est-à-dire après la dernière couche convolution. Une ou deux couches FC sont généralement utilisées avant d'appliquer le classificateur softmax.

II.6.2.6 La fonction d'activation softmax

La fonction d'activation appliquée à la dernière couche entièrement connectée est généralement différente des autres couches. La fonction d'activation doit être choisie en fonction de chaque tâche. Une fonction d'activation appliquée aux tâches de classification multi-classes est une fonction softmax qui normalise la valeur de sortie réelle de la dernière couche entièrement connectée à la probabilité de classe cible, où chaque valeur est comprise entre 0 et 1, la somme de toutes les valeurs est 1[28].

Mathématiquement, Softmax est défini comme :

$$s(y_i) = \frac{e^{y_i}}{\sum_{j=1}^{n} e^{y_j}} \qquad \text{Éq 19}$$

Où n est le nombre de classes et yi les éléments de vecteur d'entrée.

II.6.2.7 Normalisation des lots (BN)

Présenté pour la première fois par Ioffe et Szgedy dans leur article de 2015, Batch Normalization (BN en abrégé), comme son nom l'indique, sont utilisés pour normaliser les activations d'un volume d'entrée donné avant de le passer dans la couche suivante du réseau [39].

Si nous considérons x comme notre mini-lot d'activations, alors nous pouvons calculer le x̂ normalisé via l'équation suivante :

$$\hat{x} = \frac{x_i - \mu_\beta}{\sqrt{\sigma^2{}_\beta + \varepsilon}} \qquad \text{Éq 20}$$

Lors de l'apprentissage, nous calculons les μ_β et σ_β sur chaque mini-lot β, talque :

$$\mu_\beta = \frac{1}{M}\sum_{i=1}^{m} x_i \rightarrow \sigma^2{}_\beta = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_\beta)^2 \qquad \text{Éq 21}$$

Nous mettons la valeur de $\varepsilon = 10^{-7}$, pour éviter de prendre la racine carrée de zéro.

II.6.2.8 Dropout

Le dernier type de couche dont nous allons parler est Dropout. Le Dropout est en fait une forme de régularisation qui vise à prévenir le débordement en augmentant la précision des tests, peut-être au détriment de la précision de l'entraînement. Pour chaque mini-lot de notre ensemble d'apprentissage, les couches Dropout, avec la probabilité p, déconnectent

aléatoirement les entrées de la couche précédente à la couche suivante de l'architecture réseau, (figure II-13) représente exemple d'application d'un Dropout de 50% il est important de remarquer que la moitié des connexions sont supprimées [39].

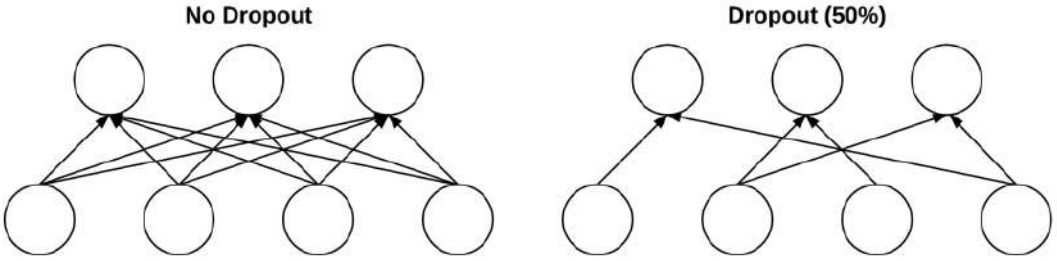


Figure II-13: Introduction du Dropout dans un réseau ANN, la moitié des neurones sont désactivés.

II.7 Conclusion

Dans ce chapitre, on a expliqué que les réseaux de neurones artificiels sont des systèmes calculatoires inspirés par les processus biologiques qui se produisent dans le cerveau humain. Nous avons montré aussi les types d'apprentissages, identifié la fonction d'activation de base ainsi que les architectures des réseaux neurones.

On a également présenté les types des réseaux de neurones les plus exploitées et les différents couches constituer des réseaux de neurones convolutifs.

Chapitre III: Mise en marche, test et implémentation

III.1 Introduction

Dans ce chapitre, on découvre les différentes méthodes de détection d'enveloppe et on montre quelle est la meilleure méthode.

Nous mettrons en avant notre conception et on va montrer comment réaliser un classifieur de défauts des roulements basant sur l'intelligence artificielle.

Pour cela, on va travailler avec le langage de programmation python et des bibliothèques comme Tensorflow et Keras pour l'apprentissage et la classification afin d'améliorer les performances du modèle réalisées.

En plus, le travail réalisait a pour but d'implémenter le système proposé de détection des défauts des roulements sur une plateforme embarquée en utilisant le Raspberry Pi 3 B+, afin d'obtenir un système qui répand aux exigences des applications industrielles en matière d'encombrante et consommation d'énergie.

III.2 Base de données choisie

Pour ce travail nous avons opté pour la base de données CWRU (Case Westerne). Les données de la base sont collectées par un bon d'essai qui consiste un moteur électrique de 2 chevaux (HP) (Reliance Electric Motor) qui entraine un axe relié à un capteur de couple et un capteur de vitesse angulaire. Un couple est appliqué au moteur via un dynamomètre qui joue le rôle d'une charge variable. Pour effectuer les tests, des défauts ont été implantés aux différentes parties des deux roulements, côté entrainement et côté ventilateur, par la méthode d'électroérosion (EDM), le diamètre des défauts introduits varie entre 0.18mm et 0.71mm. La maquette comporte aussi trois accéléromètres qui servent à capturer les signaux vibratoires du banc de test, et sont positionnés au niveau du roulement côté d'entrainement, du roulement côté ventilateur et de la base du banc de test (figure III-1).

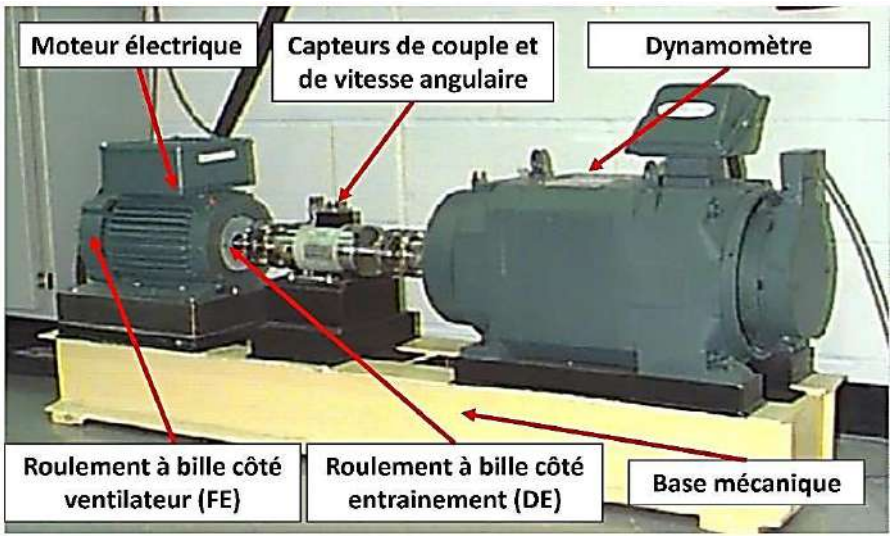


Figure III- 1: Banc d'essai des défauts des roulements.

Les signaux récoltés sur cette maquette ont été classées sur la base de cinq critères :

- Le roulement sur lequel le défaut a été implanté (côté entrainement ou ventilateur).
- La fréquence d'échantillonnage (48KHz et 12KHz).
- L'emplacement du défaut sur le roulement (aucun défaut, défaut sur la bague extérieure, défaut sur la bague intérieure, défaut sur un élément roulant) et ceci pour le roulement côté entrainement et côté ventilateur.
- Vitesse d'entrainement de l'axe qui dépend de la charge appliquée au moteur par le dynamomètre.
- Diamètre du défaut (0.18mm à 0.71mm avec un pas de 0.18mm).

Les signaux sont délivrés au format Matlab (fichiers .mat) et la plupart sont structurés de la façon suivante :

- **DE** (Drive End) : signal récolté à partir de l'accéléromètre côté entrainement
- **FE** (Fan End) : signal récolté à partir de l'accéléromètre côté ventilateur
- **BA** (Base) : signal récolté depuis la base de la maquette
- **RPM** (Révolutions Per Second) : vitesse de rotation du moteur durant le cas de test.

Origine du défaut	**Bague interne**	**Bague externe**	**Élément roulant**
Fréquence du défaut* (Hz)	157.91	104.53	137.46

Tableau III- 1: Fréquences des défauts de roulements.

* Les fréquences ont été calculées en utilisant une vitesse de rotation de l'axe du moteur de 1750 révolution par minutes, ce qui correspond à une fréquence de rotation de 29.16 Hz

Pour effectuer nos tests, nous avons travaillé sur les signaux vibratoires échantillonnés à 48 KHz pour les accéléromètres côté entrainement DE. Le modèle de roulement utilisé est le SKF 6205, les fréquences des défauts de ce type de roulements sont résumées sur le (tableau III-1).

Chaque signal de vibration de défaut de roulement dans la base de données sélectionnée a été enregistré pendant 10 secondes avec une fréquence d'échantillonnage de 48 KHz, ce qui nous donne environ 4 800 000 échantillons par structure dans le fichier Matlab, qui est un très grand échantillon. Afin de prétraiter nous devons diviser le signal en trames de 4096 points.

Les paramètres choisis pour le prétraitement sont :

- taille des trames N =48000 points.
- taille des segments L = 4096 points.

- taille des sauts de chevauchements entre chaque segment H = 128 points.
- nombre de point choisi pour effectuer notre approche L = 4096 points.

III.3 Présentation des outils utilisés

III.3.1 Matlab

Abrégés de (Matrix (Maths) LaBoratory), est un langage de programmation utilisé à des fins de calcul numérique, développé par la société The MathWorks.

MATLAB permet d'effectuer des opérations sur des matrices, d'afficher des courbes et des données, de mettre en œuvre des algorithmes. C'est le logiciel le plus utilisé dans le monde de l'ingénierie et la science. Il touche plusieurs domaines tels que les systèmes automobiles, les dispositifs de surveillance de la santé, les réseaux électriques intelligents, l'apprentissage automatique, la robotique, le traitement du signal, le traitement d'images, les systèmes de communications, les finances, et bien plus encore.

MATLAB peut s'utiliser tout seul ou avec des boites à outils (généralement connues sous le nom de 'toolbox') et à chaque domaine est associé une ou plusieurs boites à outils spécifiques, par exemple dans le domaine des Mathématiques, Statistiques, et d'Optimisation on trouve la : Partial Différentiel Equation Toolbox (boite à outils pour la résolution des équations différentiels), ou la : Curve Fitting Toolbox (boite à outils pour l'ajustement des courbe), ou encore la : Neural Network Toolbox (boite à outils pour l'utilisation des réseaux de neurones) qu'on aura l'opportunité d'expliquer ultérieurement.

Ainsi qu'en les intégrant à Simulink et à la Conception basée sur un modèle (Model Based Design) [36].

Nous avons utilisé la version R2018a.

III.3.2 Python

Python est un langage de programmation haut niveau, Python est un excellent langage de programmation orienté objet, interprété et interactif. Il est souvent comparé (favorablement bien sûr) à Lisp, Tcl, Perl, Ruby, Visual Basic, Visual Fox Pro, Scheme ou Java ... etc.

Python combine un pouvoir remarquable avec une syntaxe très claire. Il comporte des modules, des classes, des exceptions, des types de données dynamiques de très haut niveau et le typage dynamique. Il existe des interfaces vers de nombreux appels systèmes et bibliothèques, ainsi que vers différents systèmes de fenêtrage.

Les nouveaux modules intégrés sont faciles à écrire en C ou C ++ (ou dans d'autres langages, selon l'implémentation choisie). Python est également utilisable comme langage

d'extension pour les applications écrites dans d'autres langages nécessitant des interfaces de script ou d'automatisation facile à utiliser [29].

Nous avons utilisée Anaconda Pour l'utilisation, la programmation et l'installation des bibliothèques.

La dernière version du python est la version 3.10.4 c'est la même que nous avons utilisé.

Il est disponible au site web officiel du python : www.python.org

III.3.2.1 Les bibliothèques utilisé

- ***Tensorflow***

TensorFlow est une bibliothèque logicielle open source pour dédier au Machine Learning et Deep Learning.

Son architecture flexible permet un déploiement facile du calcul sur diverses plates-formes (CPU, GPU, TPU), et des ordinateurs de bureau aux clusters de serveurs, aux périphériques mobiles. Initialement développé par des chercheurs et des ingénieurs de l'équipe de Google Brain au sein de l'organisation de l'IA de Google, il s'appuie sur l'apprentissage automatique et l'apprentissage en profondeur (figure III-2) [30].



Figure III- 2: Logo TensorFlow.

- ***Keras***

Keras est une API de réseaux neuronaux de haut niveau, écrite en Python et capable de s'exécuter sur TensorFlow, CNTK ou Theano. Il a été développé dans le but de permettre une expérimentation rapide (figure III-3) [31].

Être en mesure de passer de l'idée au résultat le plus rapidement possible, est la clé pour faire de la recherche :

- Permet un prototypage facile et rapide (grâce à la convivialité, à la modularité et à l'extensibilité).
- Prend en charge les réseaux convolutionnels et les réseaux récurrents ainsi que les combinaisons des deux.
- Fonctionne de manière transparente sur le processeur et le processeur graphique.



Figure III- 3: Logo Keras.

- ***Sklearn***

Scikit-learn (Sklearn) est la bibliothèque la plus utile et la plus puissante pour la Machine Learning (Apprentissage automatique) en python. Il fournit une gamme d'outils efficaces pour l'apprentissage automatique et la modélisation statistique, notamment la classification, la régression, le regroupement et la réduction de la dimensionnalité via une interface cohérente en python. Cette bibliothèque est principalement écrite en python et repose sur Numpy, Scipy et Matplotlib (figure III-4) [32].



Figure III- 4: Logo Sklearn.

- ***Matplotlib***

Matplotlib est l'un des packages Python les plus populaires pour la visualisation de données. Il s'agit d'une bibliothèque multiplateforme permettant de créer des tracés 2D à partir de données dans des tableaux (figure III-5). Il fournit une API orientée objet qui facilite l'intégration du traçage dans les applications à l'aide de la boîte à outils Python GUI telle que :

- PyQt.
- WxPthonotTkinter.

Il peut également être utilisé dans :

- Les shells Python.
- Les shells IPython.
- Les ordinateurs portables Jupyter.
- Les serveurs d'applications web.

Figure III- 5: Logo Matplotlib.

III.4 Prétraitement des signaux

III.4.1 Application de la méthode Hilbert sur les signaux vibratoires

On applique d'abord la détection d'enveloppe par la méthode (Hilbert) résultats obtenus sont mentionnés sur la figure (III-6) une vue d'ensemble de l'état du signal avec les différents processus que nous appliquerons est donnée sur la figure (III-6) :

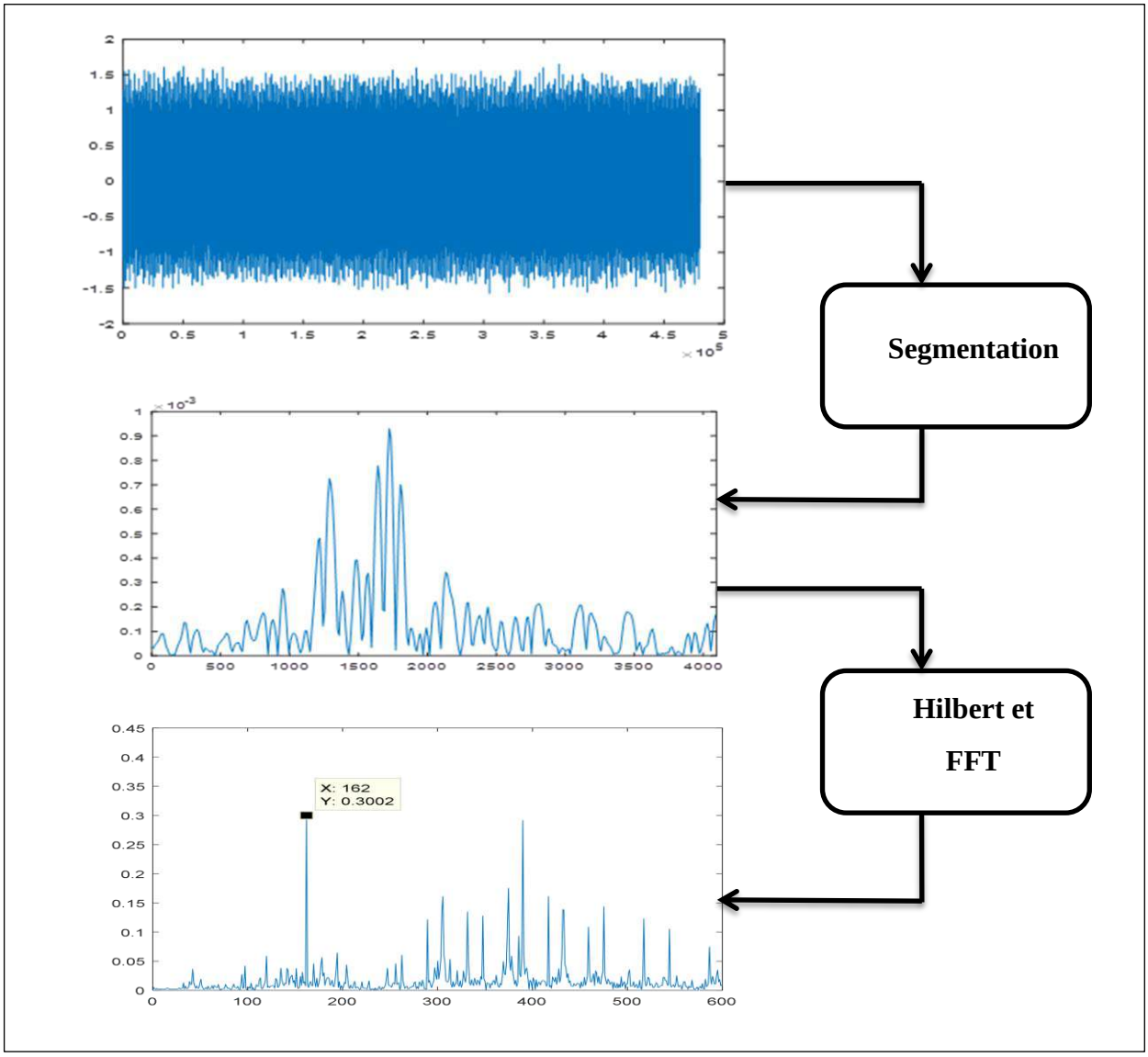


Figure III- 6: Application de Hilbert.

III.4.2 Application de notre approche sur les signaux vibratoires

On applique d'abord la détection d'enveloppe par notre approche résultats obtenus sont mentionnés sur la figure (III-7) une vue d'ensemble de l'état du signal avec les différents processus que nous appliquerons est donnée sur la figure (III-7) :

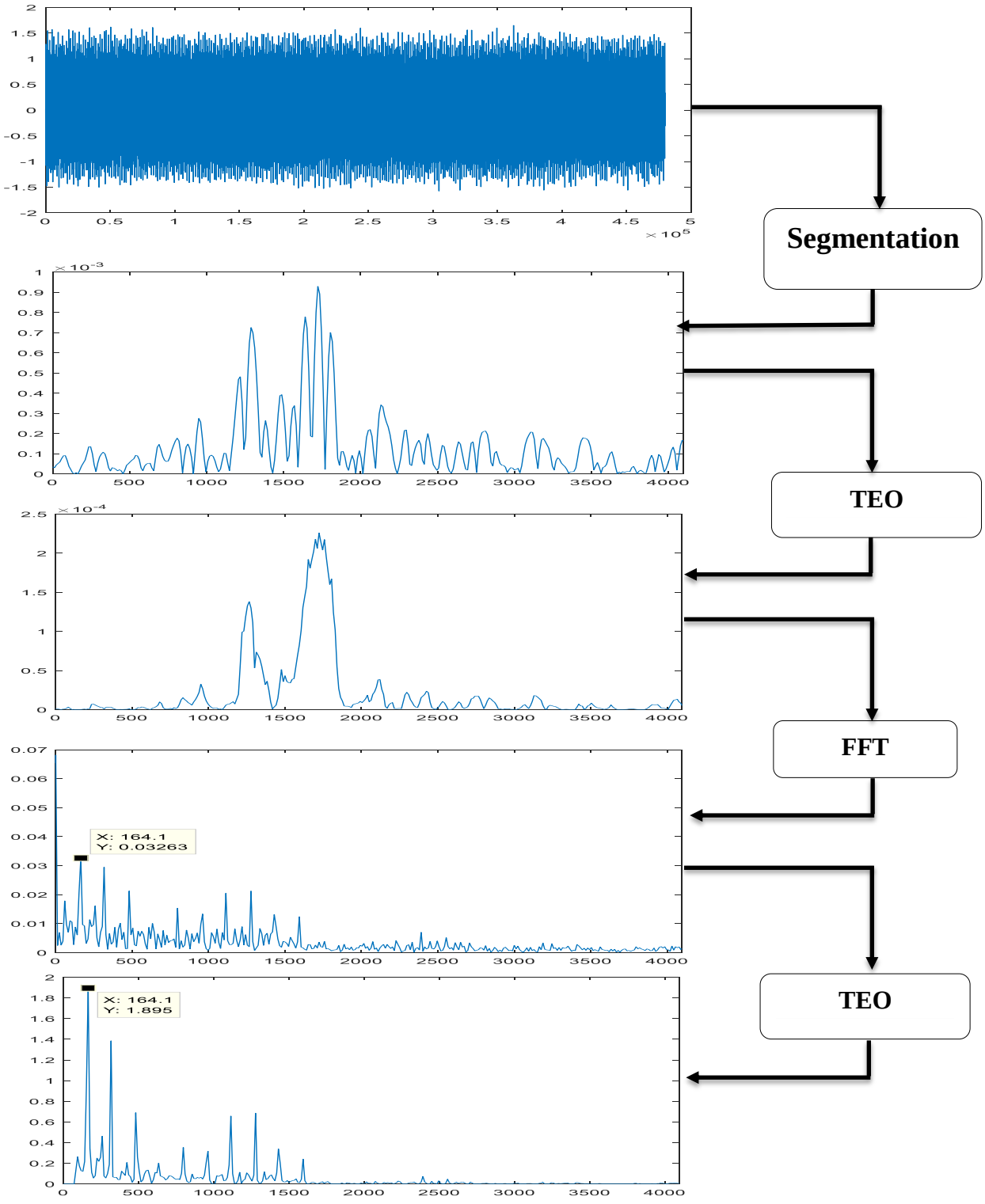


Figure III- 7: Application de notre approche.

III.4.3 Comparaison entre notre approche et Hilbert

Dans cette partie nous avons fait une étude comparative entre notre approche et l'outil classique de détection d'enveloppe (Hilbert).

D'après les résultats obtenus, nous constatons facilement que les résultats de détection d'enveloppe par notre approche sont les plus précis, les plus clairs et les plus lisses, comparant avec ceux obtenus par l'approche classique (Hilbert).

III.5 Préparation des données

Nous remarquons que les informations des défauts se trouvent dans les 200 points premier donc nous prenons ces 200 comme entrée dans le modèle CNN.

Dans cette partie nous avons pris des signaux de différents défauts de la base de données selon les classes qu'on a choisi 4 classes.

Nous avons traité ces fichiers selon les étapes de l'organigramme de la figure (III-8).

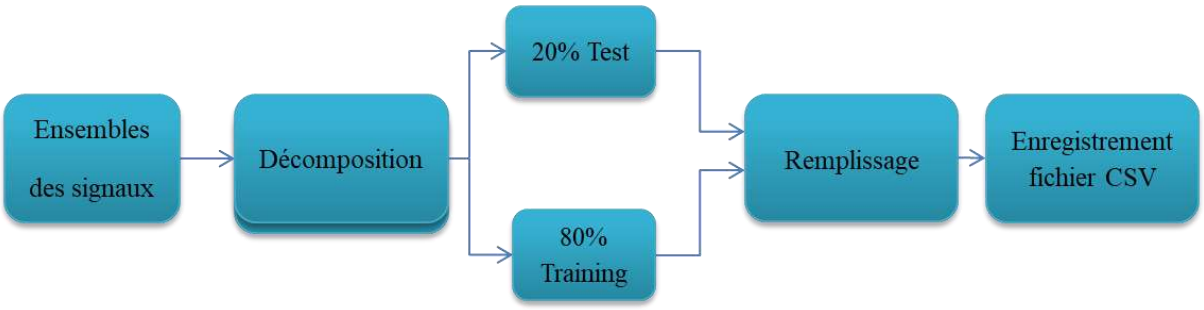


Figure III- 8: Organigramme de traitement des données.

III.5.1 Décomposition des signaux

Cette décomposition c'est la séparation des signaux résultants après le traitement de notre approche qu'on veut diviser en trois parties (figure III-9) :

- Partie 1 : 70% des signaux originale pour la partie d'entrainement des réseaux neurones.
- La deuxième partie de 30% des signaux pour le test on va prendre 10% pour Validation et 20% pour le test.

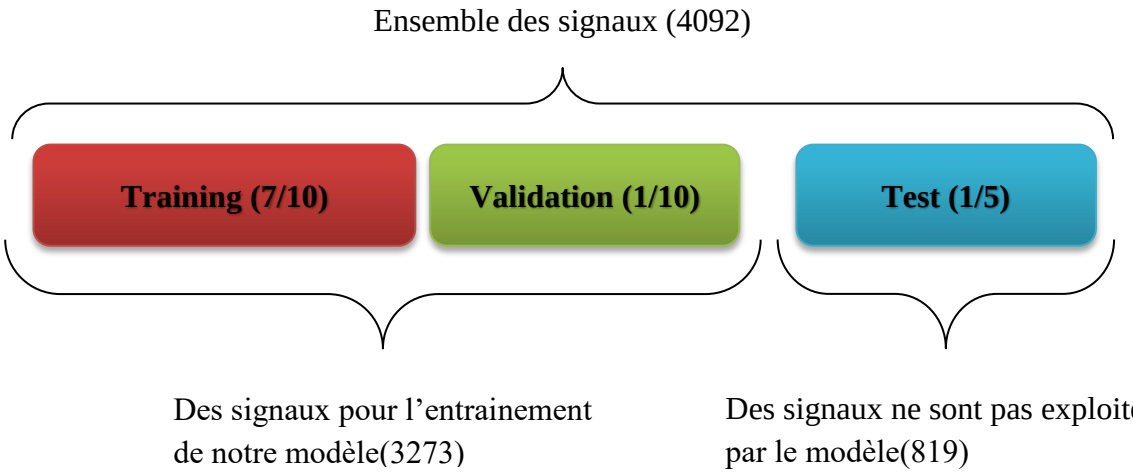


Figure III- 9: Décomposition des signaux.

III.5.2 Remplissage des signaux

Après avoir effectué la segmentation et le prétraitement, l'étape suivante consiste à prendre chaque signal en tant que vecteur de ligne et à combiner différentes les signaux dans matrices avec un ordre bien définit (figure III-10).

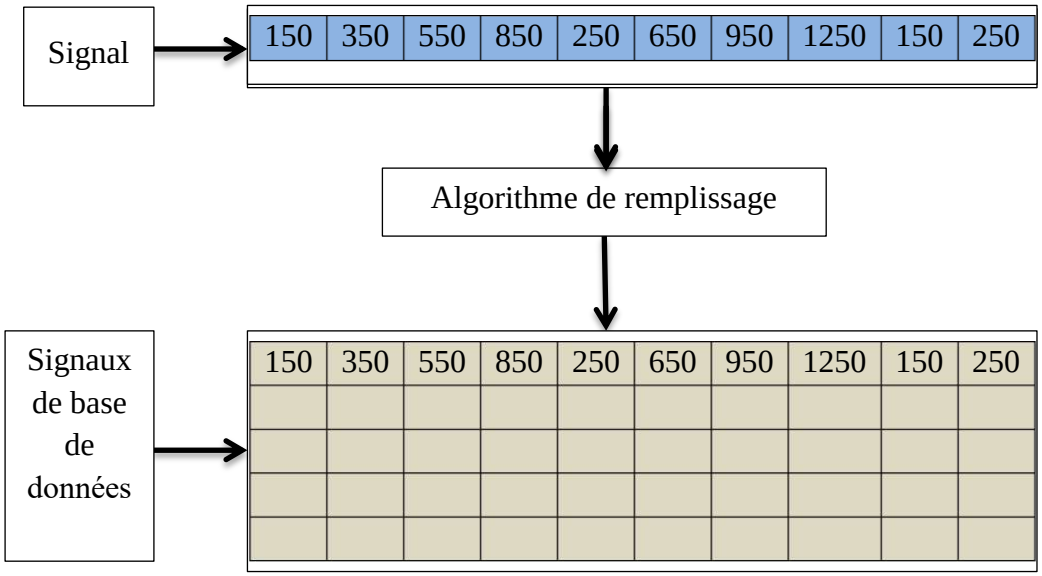


Figure III- 10: Remplissage des signaux.

III.5.3 Enregistrement

Après avoir effectué les différentes étapes précédentes, nous avons enregistré la base données sous forme un fichier CSV à l'aide d'une commande dans le Matlab (figure III-11).

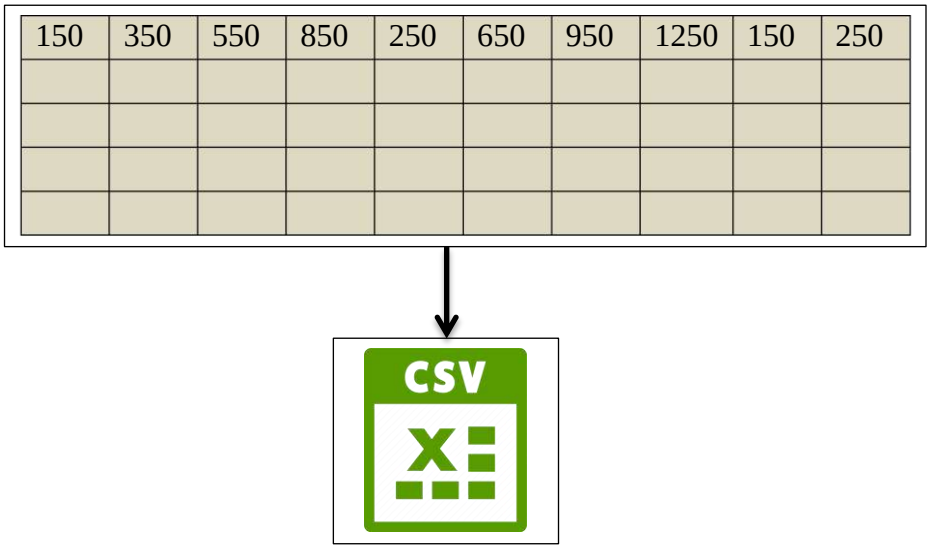


Figure III- 11: Enregistrement sous forme csv.

III.5.4 Distribution de base de données

Après avoir appliqué les différentes étapes de traitement, nous pouvons voir que notre base de données contient 3274 signaux, chacun sous la forme d'un vecteur ligne (figure III-12).

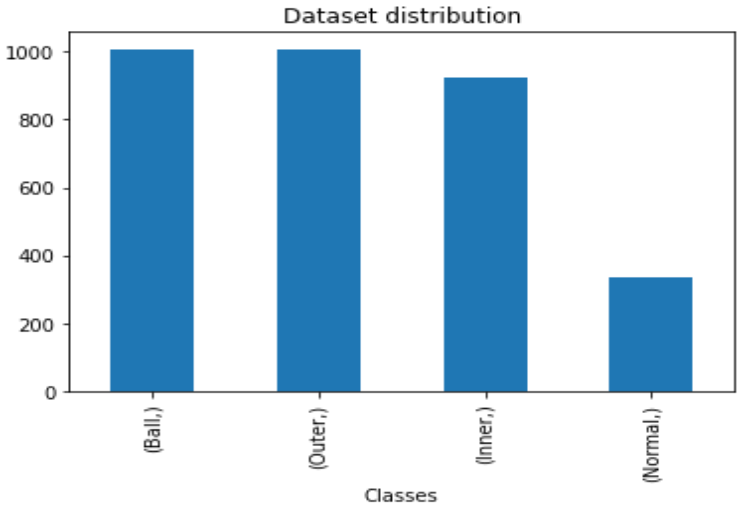


Figure III- 12: Distribution de la Base de données.

Normal	⟶	336
Inner	⟶	922
Outer	⟶	1008
Ball	⟶	1008

Le diagramme de dispersion (Scatter plot) (figure III-13) est largement utilisé pour tracer des points de données sur un axe horizontal et un axe vertical dans le but de montrer à quel point une variable est affectée par une autre.

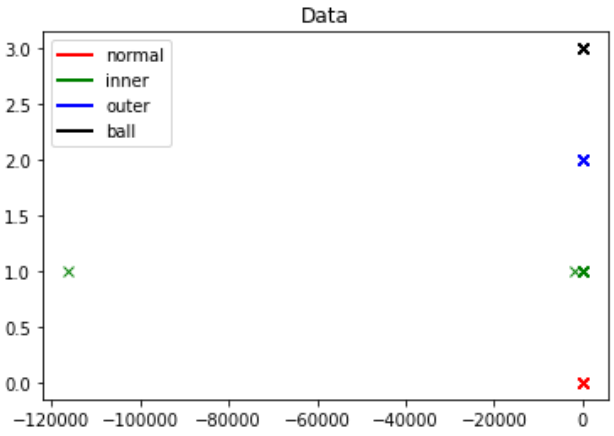


Figure III- 13: Scatter plot du Base de données.

III.6 Modèle CNN

De point cela on a créé notre propre architecture (modèle) CNN à l'aide de logiciel Python (Anaconda Jupyter).

(Figure III-14) représente les différentes étapes à suivre pour entrainer le réseau CNN

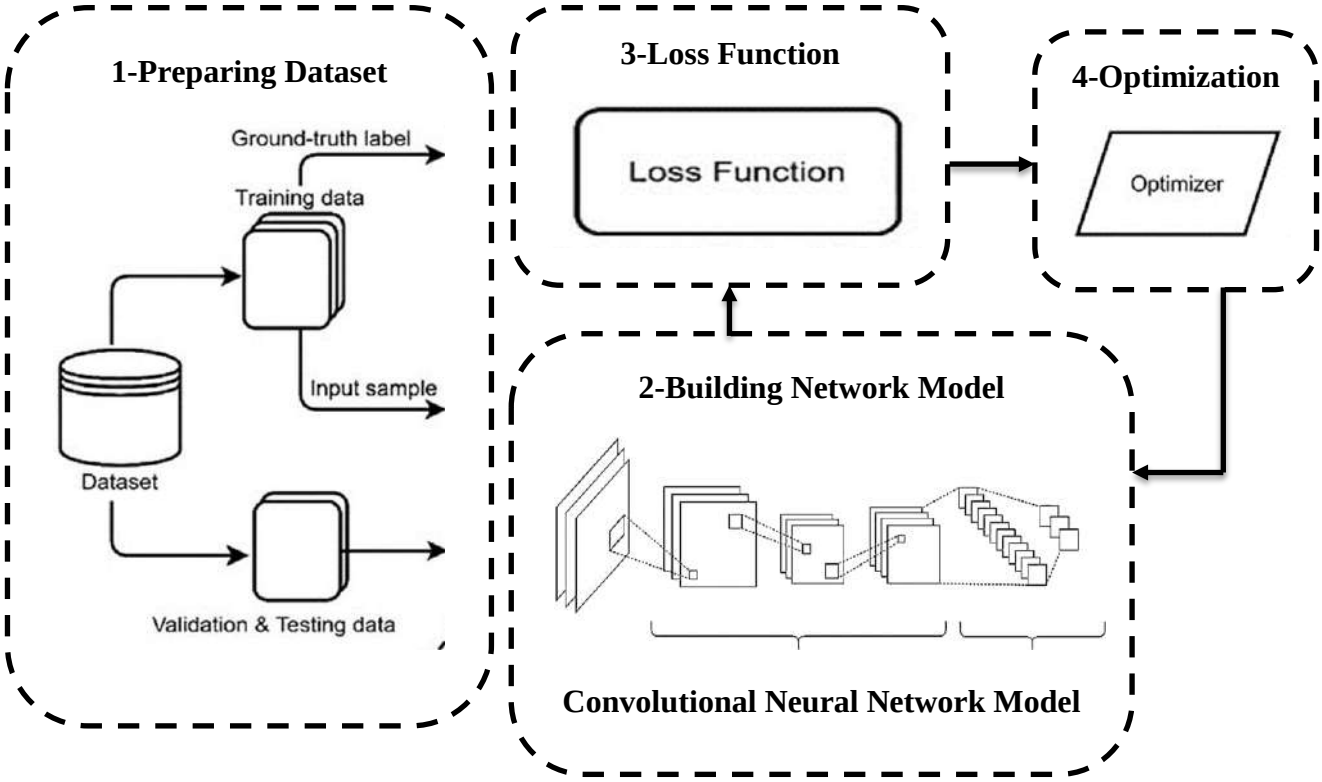


Figure III- 14: Diagramme schématique de procédure d'entrainement des réseaux CNN.

III.6.1 Architecture utiliser

Pour une très bonne classification des défauts de roulements, nous avons créé notre propre architecture du modèle CNN à 1D qui comporte les différentes couches dans le tableau suivant :

	Architecture	
	Conv1D (32,3) Padding	
	MaxPooling1D(2)	
	Conv1D (64,3) Padding	
	MaxPooling1D(2)	
	Conv1D (64,3)	
	AveragePooling1D(2)	
	Flatten	
	Dense(50)	
	Dense(70)	
	Dense(32)	
	Dense(4)	
Entrainement	**Précision**	98.8%
	Pertes	0.08
Validation	**Précision**	98%
	Pertes	0.09

Tableau III- 2: Les couches utilisent dans cette architecture.

Cette architecture est constitué de :

- 3 couches de Convolutions 1D.
- 2 couches de Maxpooling 1D.
- Une couche d'AveragePooling 1D.
- 4 couches Dense.

Nous avons des paramètres totaux de 101512.

Après de nombreux tests et modifications au niveau de l'architecture, nous constatons que cette architecture est la plus précise et qu'elle nous donne les meilleurs résultats que les autres architectures (figure III-15).

Notre signal de 200 points passe par la première couche de convolution. Cette couche est se compose de 32 filtres de longueur 3 avec le padding, chaque couche de convolution est suivie par une fonction d'activation ReLu, alors que la sortie de cette couche est de taille 200×32.

Cette sortie est injecté pour applique MaxPooling1D pour réduire la taille du signal. La sortie de cette couche, est de taille 100×32.

Cette dernière est à son tour été introduit comme entré dans la deuxième couche de convolution qui se compose de 64 filtres de longueur 3 avec padding, cette couche de convolution est suivie par une fonction d'activation ReLu, la sortie est de taille 100×64. Ensuite, on applique MaxPooling1D pour réduire la taille de cette sortie à 50×64. On procède de la même manière pour de convolution mais sans padding et d'AveragePooling1D. La sortie de ces couches est de taille 24×64.

Après les 3 couches de convolution et les 3 couches de Pooling1D, nous utilisons trois couches de connexion entière (FC). La première couche de 50 neurones avec une fonction d'activation ReLu, suivie par une deuxième couche de 70 neurones avec la même fonction d'activation ReLu, suivie par une deuxième couche de 32 neurones avec la même fonction d'activation ReLu et la dernière couche est un softmax qui permet de calculer la probabilité des 4 classes (nombre de classe dans notre base de données).

Cette architecture donne comme résultats une précision de 98.8% pour l'entrainement, et 98% pour la validation. Pour les pertes on a obtenu 0.08 pour l'entrainement, et 0.09 pour la validation.

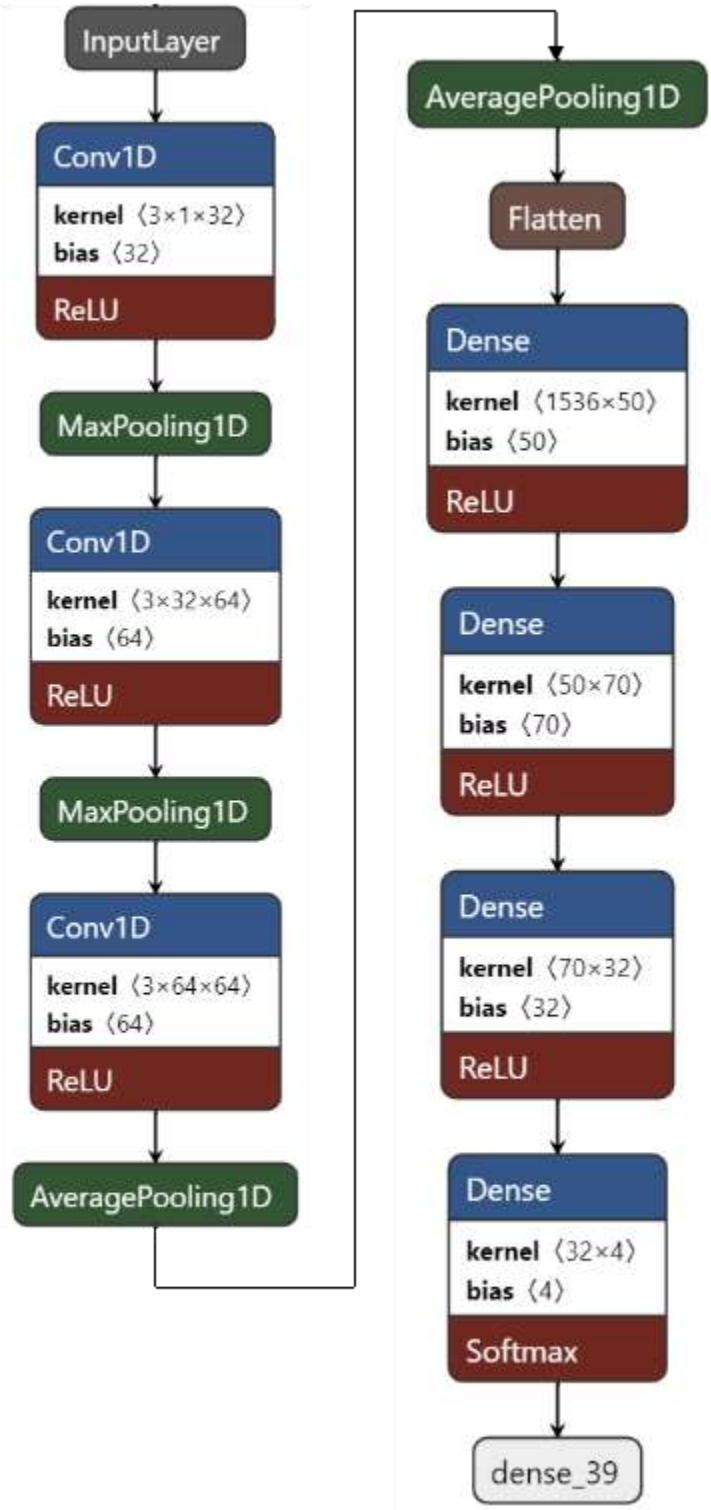


Figure III- 15: L'Architecture utilisée.

III.6.2 Méthode d'optimisation

Nous avons eu recours à de nombreuses méthodes pour obtenir un très bon résultat de performance de classification et pour préserver le meilleur modèle. Ces méthodes sont :

III.6.2.1 KFold Cross Validation

C'est une méthode d'estimation des performances d'un modèle sur des données invisibles. Cette technique utilisée pour le réglage des hyper paramètres de sort que le modèle avec la valeur la plus optimale des hyper paramètres puisse être formé. L'avantage de cette approche est que chaque exemple est utilisé pour l'entrainement et la validation (as part of a test fold) exactement une fois. Cela donne une estimation de la variance de performance du modèle plus faible que la méthode d'exclusion. (Nous avons utilisé) cette technique (est utilisée) pour minimiser et éviter l'Overfiting, qui peut se produire lorsqu'un modèle est formé à l'aide de toutes les données.

Nous avons utilisé la méthode de K-Fold Cross Validation pour mesure de « tester » le modèle sur K (10) ensembles de données différentes, ce qui permet de garantir que notre modèle est généralisable (figure III-16).

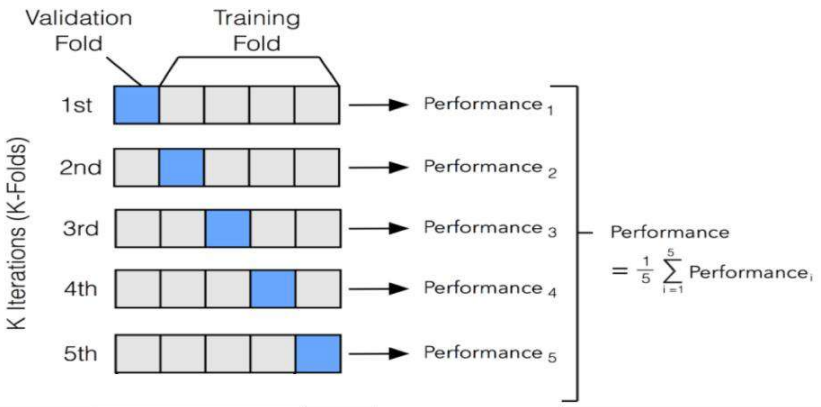


Figure III- 16: K-Fold Cross Validation.

III.6.2.2 Callbacks Model Check Point

Nous avons utilisé cette fonction de Callbacks Keras car ils peuvent jouer un rôle très important. L'entrainement de notre modèle peut prendre des heures ou des jours pour compléter, nous devrions donc avoir une fonction pour contrôler notre modèle durant l'exécution.

Nous avons utilisé cette fonction pour sauvegarder le modèle le plus précis et en comparant avec la précision de la prochaine époque, s'il augmente nous pouvons prendre ce modèle et écrasé l'ancien modèle pour garder le meilleur modèle.

III.6.2.3 Compiler

- Loss : Cette fonction de cout que le modèle va utiliser pour minimiser les erreurs. Nous utilisons le « categorical_crossentropy », car nous avons une classification multi classe.

- Optimizer : Nous avons utilisé l'optimiser stochastique « Adam », car il donne des très bons résultats.
- Metrics : Dans le cas d'un problème de classification, nous utilisons metrics= ['accuracy'].

III.7 Résultats de classification

Les résultats obtenus par notre modèle CNN 1D avec l'utilisation de la méthode KFold, nous avons illustre les différents résultats en termes de :

- Précision.
- Erreur.
- Rapport de classification.
- Matrice de confusion.

Nous avons prendre un nombre des époques égale à 100 époque.

III.7.1 Graphe de précision et d'erreur

La (figure III-17) représente les graphes de : Accuracy et Loss d'entrainement et de validation en fonction le nombre des époques avec l'utilisation de KFold, on remarque que l'accuracy pour la validation augmente avec le nombre des époques de la même façon pour l'entrainement.

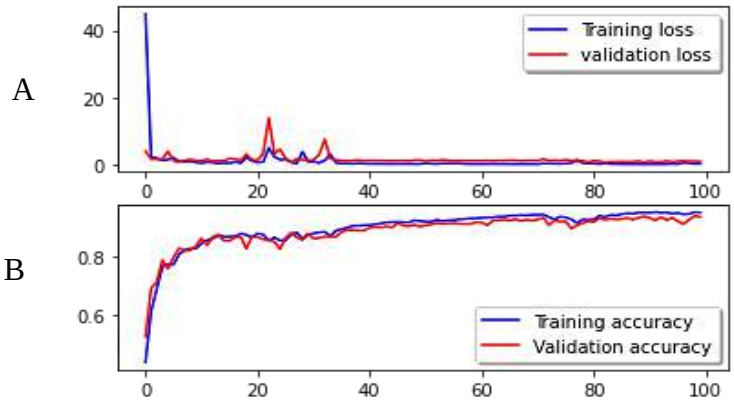


Figure III- 17: A- Graphe de Loss, B-Graphe de l'accuracy.

(La figure III-17-A) montre l'erreur d'apprentissage et de la validation qui diminue dans l'intervalle du nombre d'époques [0-5] et dans l'intervalle [5-100], la validation et l'apprentissage se stabilise dans le niveau (0).

(La figure III-17-B) montre la précision de l'apprentissage et de la validation augmente dans l'intervalle du nombre d'époques [0-20] et dans l'intervalle [20-100] la validation et l'apprentissage reste presque stationnaire dans le niveau (1).

III.7.2 Rapport de classification

Pour la classification les classes Normal (0) et Inner (1), le taux de précision est de 100%, le f1-score est de 100% ce qui est un très bon résultat,

Et pour les classes qui va nous permettre de Outer (2) et Ball (3) on voit que le taux de précision est comprise a 97%, le f1-score est comprise a 97% ce qui est un résultat très proche du bien (figure III-18).

	precision	recall	f1-score	support
0	1.00	1.00	1.00	24
1	1.00	1.00	1.00	83
2	0.97	0.97	0.97	80
3	0.97	0.97	0.97	74
accuracy			0.98	261
macro avg	0.99	0.99	0.99	261
weighted avg	0.98	0.98	0.98	261

Figure III- 18: Rapport de Classification.

- Précision : C'est le rapport entre les vrais positifs et la somme des vrais et faux positifs. Est le rapport entre les prévisions positives correctes et les prévisions positives totales.
- Recall : C'est le rapport entre les vrais positifs et la sommes des vrais positifs et des faux négatifs. C'est une mesure du nombre de points positifs que notre modèle est capable de rappeler à partir des données.
- F1-Score : C'est la moyenne harmonique pondérée de précision et recall. Plus la valeur du f1-score est proche de 1.0. Est une moyenne harmonique de précision et de recall.

$$\text{F1 Score} = \frac{\text{Précision} + \text{Recall}}{2} \qquad \text{Éq 30}$$

- Support : C'est le nombre des signaux de chaque classe dans la base de données.

D'après l'analyse du rapport de classification nous remarquons que le taux de precision est de 97% avec un recall de 97% et f1-score de 97% pour la troisième classe, ce n'est pas un bon résultat par rapport aux autres classe car les signaux de cette classe sont similaires que celle de la quatrième classe Ball .

III.7.3 Matrice de Confusion

La matrice de confusion permet d'évaluer la performance de notre modèle, puisqu'elle reflète les métriques du Vrai positif, Vrai négatif, Faux positif et Faux négatif.

(La Figure III-19) illustre de près la position de ces métriques pour chaque classe.

Dans notre matrice nous remarquons que les classes Normal et Inner sont bien classées, mais les classes Outer et Ball sont moins classées par rapport aux autres classes.

La classe Outer 97% est bien classée mais 3% sont classé comme classe Ball, De plus on trouve que 3% de la classe Ball est classé comme classe Outer car les deux pics sont presque adjacents.

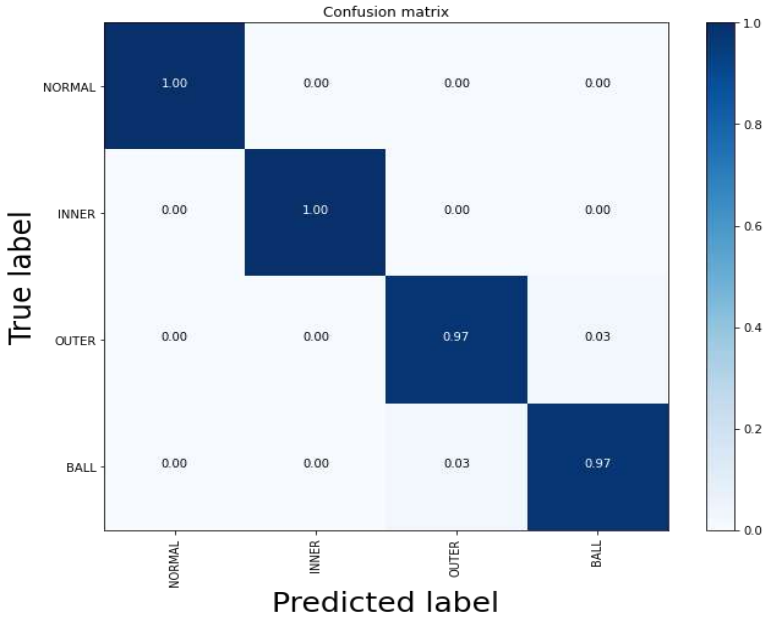


Figure III- 19: Matrice de confusion de validation.

Le modèle qui donne grand précision représenté au tableau ci-dessus :

	Accuracy	**Loss**
Train	98.8%	0.08
Validation	98%	0.09

Tableau III- 3: Meilleur Résultat obtenu avec KFold.

III.8 Comparaison

Nous essayons notre architecture précèdent sans l'utilisation de KFold et nous obtenons les résultats ci-dessus :

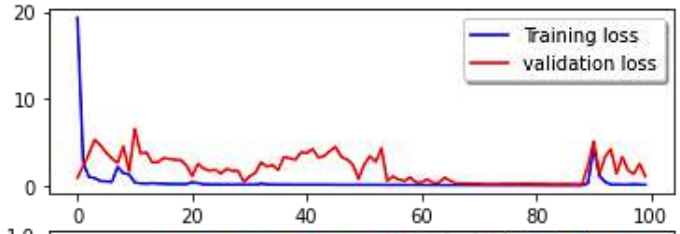
	Accuracy	**Loss**
Train	98%	0.03
Validation	96%	0.17

Tableau III- 4: Meilleur Résultat obtenu sans KFold.

La (figure III-20) représente les graphes de : Accuracy et Loss d'entrainement et de validation en fonction le nombre des époques sans l'utilisation de KFold, on remarque que

l'accuracy pour la validation augmente avec le nombre des époques de la même façon pour l'entrainement.

A



B

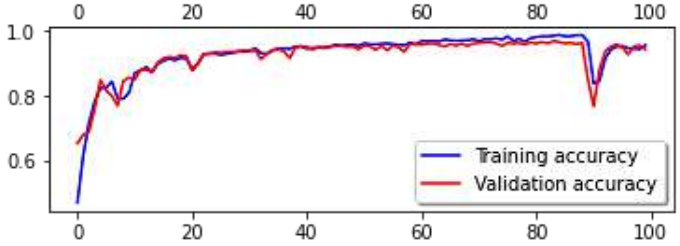


Figure III- 20: A- Graphe de Loss sans KFold, B-Graphe de l'accuracy sans KFold.

On remarque d'après les deux méthodes utilisées on a des bons résultats qui a eu une meilleur classification des signaux vibratoires, ce qui prouve plus d'efficacité du modèle CNN pour résoudre ce type de problématique.

Aussi on remarque qu'avec l'utilisation de KFold nous obtenons un graphe de précision et des pertes plus claires que celles qu'on n'utilise pas la méthode de KFold.

III.9 Résultats du test de modèle CNN

Dans cette partie nous utilisons les signaux de test final qui ne sont pas exploités ultérieurement (partie de 20%) par le modèle CNN

Les étapes de chargé notre modèle CNN expliquer dans la figure (III-21) :

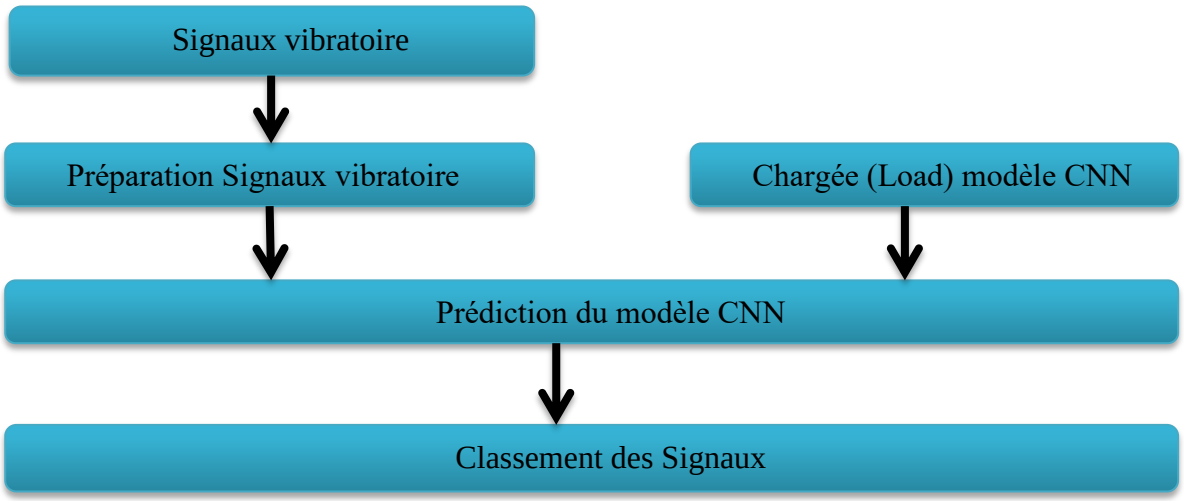


Figure III- 21: Test expérimental.

III.9.1 Chargée (Load) modèle CNN

Nous avons utilisé la commande « load_model » pour charger le modèle CNN.

III.9.2 Préparation Signaux vibratoire

- Introduit les signaux de test avec la commande « pandas.read_csv ».
- Prendre les valeurs depuis le fichier csv avec la commande « data.values ».
- Reformulation des signaux.

III.9.3 Classement des Signaux

Après la préparation des signaux et le chargement de notre propre modèle CNN 1D utilisé nous utilisons la commande « modèle.predict » pour classifier les signaux utilisées. Les résultats de cette prédiction sont dans (figure III-22).

```
[ 80,   0,   0,   0]
[  0, 217,   0,   1]
[  0,   0, 220,  20]
[  0,   4,  19, 217]
```

Figure III- 22: Classement des signaux.

(Figure III-23) représente les résultats de classification des signaux de quatre classes utilisées par des signaux de test final (partie de 20%) par le modèle CNN.

Dans notre matrice on remarque que les classes : (normal) et (inner) sont bien classées.

Mais les classes (outer) et (ball) sont moins classées par rapport aux autres classes, puisque la classe (outer) 92% est bien classée mais 8% sont classé comme classe (Ball) car les deux classes sont très similaires.

De plus on trouve que 8% de la classe Ball est classé comme classe Outer car les deux pics sont presque adjacents et 2% comme classe Inner, et 90% qui restent ça classe est bien.

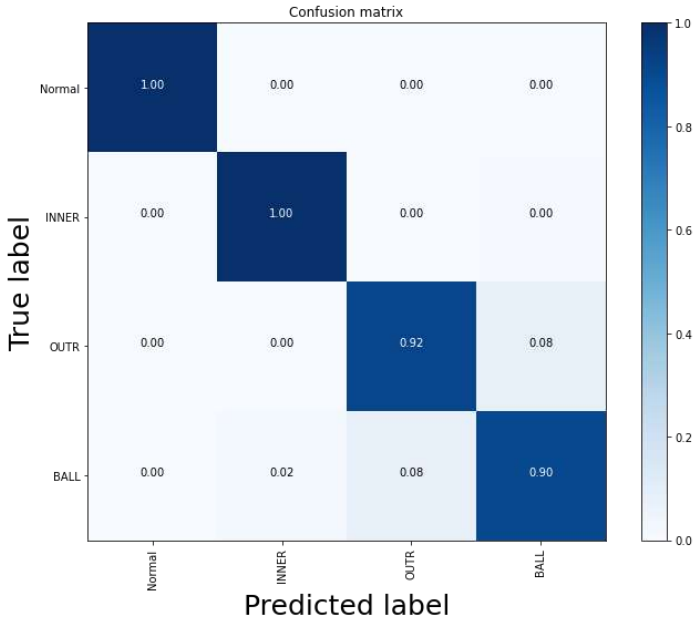


Figure III- 23: Matrice de confusion de test.

Tableau (III-5) nous montre la précision et l'erreur absolue de test que nous avons avec les commandes « accuracy_score », « mean_absolute_error » qui nous donne les résultats suivants :

	Précision	**Erreur absolue**
Résultats	**0.9434**	**0.062**

Tableau III- 5: les résultats de test expérimental.

III.10 Système d'implémentation

III.10.1 Raspberry Pi

Raspberry Pi est un petit ordinateur à carte unique. En connectant des périphériques comme Clavier, souris, affichage au Raspberry Pi, il agira comme un mini-ordinateur personnel. Il est plus lent que l'ordinateur portable ou de bureau, mais est toujours un ordinateur qui peut fournir toutes les fonctionnalités ou capacités attendues, à une faible consommation d'énergie.

Raspberry Pi Foundation fournit officiellement Raspbian OS basé sur Debian. De plus, ils fournissent NOOBS OS pour Raspberry Pi. Nous pouvons installer plusieurs versions tierces du système d'exploitation comme Ubuntu, Archlinux, RISC OS, Windows 10 IOT Core, etc.

Raspbian OS est un système d'exploitation officiel disponible gratuitement à utiliser. Ce système d'exploitation est efficacement optimisé pour une utilisation avec Raspberry Pi. Raspbian ont une interface graphique qui comprend des outils pour la navigation, la programmation Python, etc.

Nous devrions utiliser la carte SD pour stocker l'OS (système d'exploitation). Raspberry Pi a basé ARM Broadcom Processor SoC avec puce GPU (Unité de traitement graphique) [33].

III.10.2 Raspberry Pi 3 B+

Le Raspberry Pi 3 modèle B+ est le dernier produit de la gamme Raspberry Pi 3, doté d'un processeur 64-bit quad core fonctionnant à 1.4GHz, dual-band 2.4GHz et 5GHz sans fil LAN, Bluetooth 4.2/BLE, Ethernet plus rapide, et PoE capacité via un PoE HAT séparé (figure III-24) [34]. Nous avons utilisé ce modèle dans notre projet.



Figure III- 24: Raspberry Pi 3 B+.

III.10.2.1 Caractéristiques du Raspberry Pi 3 B+

Tableau (III-6) donne les caractéristiques de Raspberry Pi 3B+ :

Processeur	Broadcom BCM2837B0, Cortex-A53 64-bit SoC @ 1.4GHz
Mémoire	1GB LPDDR2 SDRAM
Connectivité	2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth4.2, BLE Gigabit Ethernet over USB 2.0 4 × USB 2.0 ports
Vidéo et Son	1 × full size HDMI Port MIPI DSI pour connecter l'écran tactile Port de caméra MIPI CSI Sortie stéréo à 4 pôles et port vidéo composite
Carte SD	Port micro SD pour le chargement du système d'exploitation et stockage de données
Accès	En-tête GPIO 40 broches étendu
Multimédia	H.264, décodage MPEG-4 (1080p30); codage H.264 (1080p30); graphiques OpenGL ES 1.1, 2.0
Energie d'entrée	5V/2.5A DC via connecteur micro USB 5 V DC via l'en-tête GPIO Alimentation par Ethernet (PoE) activée (nécessite un HAT PoE séparé)

Tableau III- 6: Caractéristiques du Raspberry Pi 3 B+.

III.10.2.2 Préparation des données (Raspberry)

- ***TensorFlow Lite***

TensorFlow Lite propose un ensemble d'outils pour convertir et exécuter des modèles sur des dispositifs. Il est utilisé pour convertir des modèles Keras déjà formés à partir du format '.h5' au format '.tflite' plus léger, adapté au Raspberry Pi et autres appareils embarqués. Il est également livré avec une boîte à outils pour la quantification post-formation et formation sensible à la quantification des modèles, avec des possibilités d'accélérer l'inférence intégrée ou pour assurer une grande précision lors de l'utilisation de modèles quantifiés [35].

Comme première étape, nous avons converti notre propre modèle CNN sauvegardé du format « .h5 » au format « .tflite », la figure (III-25) représente cette conversion.

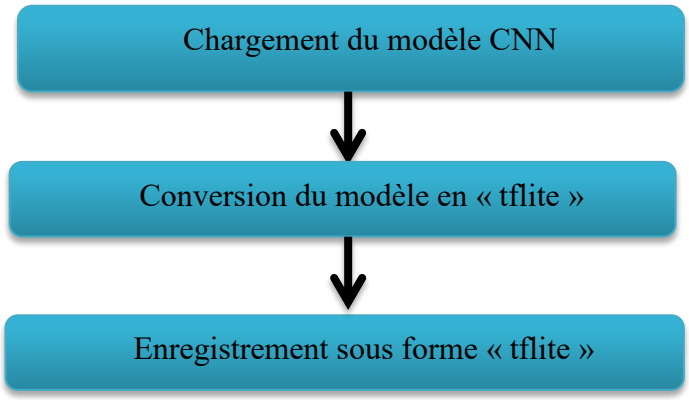


Figure III- 25: Préparation des données (Raspberry).

- Chargement du modèle : on a chargé le modèle réalisé avec la commande « load_model ».
- Conversion du modèle : on a utilisé la bibliothèque «tf.lite.TFLiteConverter.from_keras_model » pour convertir le modèle de la forme « .h5 » à la forme « .tflite ».
- Enregistrement du modèle : on a ouvert un nouveau fichier sous forme « tflite » et sauvegardé le modèle converti dans ce fichier.

III.10.2.3 Résultat obtenu

Après avoir effectué la partie d'implémentation sur la carte Raspberry Pi 3B+, nous essayons de tester le modèle CNN sur des signaux de test final avec fonction de prédiction. Après on a calculé la durée de prédiction la figure (III-26) représente les résultats de prédiction pour un seul signal.

Le fichier de test final dans cette partie segmenté le signal original pour prendre 4096 points comme entrée, ensuite on a le traité par l'application (TEO FFT TEO) après on prend seulement les 200 points premiers.

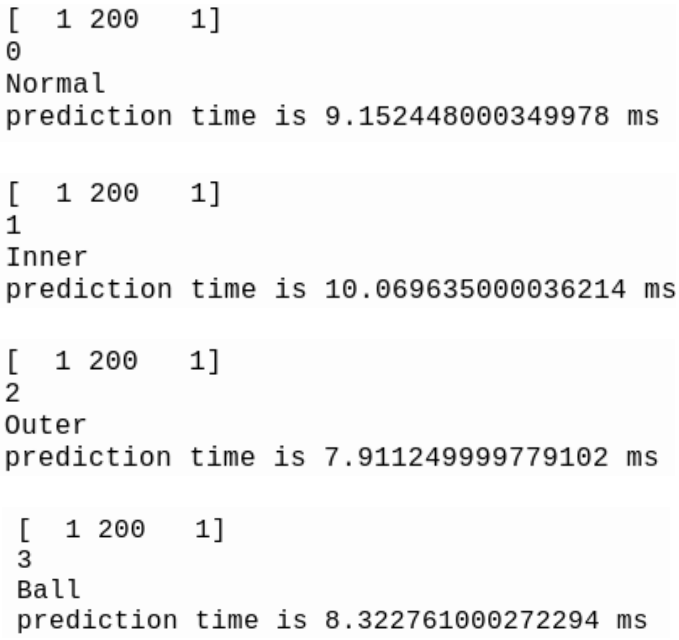


Figure III- 26: résultat d'implémentation avec Raspberry.

Le fichier de test final dans cette partie segmenté le signal original pour prendre 4096 points comme entrée, ensuite on a le traité par l'application (TEO FFT TEO) après on prend seulement les 200 points premiers.

On remarque d'après la figure (III-26) que le temps de prédiction le plus élevé est de 10.069 ms donc on déduit que la prédiction est très rapide et fiable.

Cependant, On a une durée de prédiction de 0.0083 s qui est très rapide et fiable.

	PC		**Raspberry Pi**
	Modèle h5	**Modèle tflite**	**tflite**
Un signal	0.057 s	0.006 s	0.0083s

Tableau III- 7: Résultats des implémentations

D'après les résultats du tableau (III-7) on a déduire quand le modèle sauvegarder sous forme « tflite » est plus rapide pour la prédiction (exécution) que celle de la forme « h5 » au PC.

On a aussi pour le Raspberry Pi 3B+ à une durée de prédiction très rapide.

III.11 Conclusion

Dans ce chapitre nous avons appliqué un prétraitement sur les différents signaux vibratoires avec des étapes bien définit. Après avoir effectué ce prétraitement nous avons crié une architecture CNN pour la détection des défauts des roulements à billes et on les classer.

Ensuit après l'entrainement du modèle CNN nous avons prendre les différents résultats : accuracy, loss, graphe loss accuracy, rapport de classification, matrice de confusion.

Après nous avons testé notre modèle CNN sur des signaux qui ne sont pas exploités ultérieurement.

Finalement on a été consacré à l'implémentation de notre méthode de prétraitement appliqué sur signaux vibratoires au niveau du Raspberry Pi 3 B+ à travers l'utilisation d'un langage python.

Conclusion générale

Le but de ce travail est d'implémenter sur Raspberry, un système de détection automatique des défauts de roulement pour les machines tournantes en utilisant les signaux vibratoires. Pour mieux exploiter les informations utiles dans le signal vibratoire, nous avons procédé un prétraitement qui consiste à appliquer la FFT et TEO afin d'améliorer le signale. Le signal issu du prétraitement sera utilisé pour reconstruire la base de données des signaux vibratoires et élaborés un modèle CNN pour la détection et la classification des défauts. Les tests ont donné des résultats concluants en matière de temps d'exécution et l'exactitude de la classification. Le traitement est effectué d'abord sur un PC avant de les transférer sur la plate-forme Raspberry en transformant le modèle élaboré en un modèle plus léger adapté aux plates-formes embarquées par l'utilisation de la bibliothèque TinyML de Tensorflow.

Enfin, nous souhaitons que le système de détection des défauts que nous avons élaboré pourra être testé et utilisé dans un environnement industriel réel. Ce système permettra d'une part de réduire la charge de travail des équipes de maintenance et d'autre part de diminuer les coûts de maintenance des machines.

Bibliographie

[1] AFNOR, « FD X 60-000: Maintenance industrielle - Fonction maintenance », Association Française de Normalisation, vol. 33, no 0. p. 1-29, 2002, [En ligne]. Disponible sur: http://www.ehpadneuilly.com/cariboost_files/FDX_60-000.pdf.

[2] P. C. Boulenger, Alain, Surveillance des machines tournantes par analyse vibratoire. Paris, France: Dunod, L'usine nouvelle, 2009.

[3] B. Samira, « Contribution à l'optimisation de la maintenance prédictive par l'utilisation du logiciel ONMITREND (System On Ligne) cas : Cimenterie Hamma Bouziane», UNIVERSITE DE MENTOURI DE CONSTANTINE, 2008.

[4] P. C. Boulenger, Alain, Surveillance des machines tournantes par analyse vibratoire. Paris, France: Dunod, L'usine nouvelle, 2009.

[5] BELOUADAH Abdenaceur, « Amélioration de la fiabilité d'un système Électromécanique par l'utilisation des opérations de la maintenance préventive », UNIVERSITE MOHAMED BOUDIAF DE M'SILA, 2016.

[6] S. Kojtych, « Introduction au traitement du signal et à l'analyse fréquentielle par transformées de Fourier », HAL, p. 47, 2019, [En ligne]. Disponible sur: https://hal.archives-ouvertes.fr/hal-02193929.

[7] S. Kass, « Diagnostic vibratoire autonome des roulements », INSA de Lyon, 2020.

[8] F. Landolsi, « SURVEILLANCE DES MACHINES PAR ANALYSE VIBRATOIR», in COURS DE TECHNIQUES DE SURVEILLANCE PARTIE, 2015, p. 11-20.

[9] M. Bellanger, « Traitement numérique du signal - Théorie et pratique », p. 465,2006.

[10] Boudraa, A.-O., et al. IF estimation using empirical mode decomposition and nonlinear Teager energy operator. in First International Symposium on Control, Communications and Signal Processing, 2004. 2004. IEEE.

[11] Teager, H. and S. Teager, Evidence for nonlinear sound production mechanisms in the vocal tract, in Speech production and speech modelling1990, Springer. p. 241-261.

[12] Kaiser, J.F. On a simple algorithm to calculate the'energy'of a signal. in International conference on acoustics, speech, and signal processing. 1990. IEEE.

[13] Maragos, P., T.F. Quatieri, and J.F. Kaiser. *Speech nonlinearities, modulations, and energy operators*. in *icassp*. 1991.

[14] Mitra, S.K., et al. A new class of nonlinear filters for image enhancement. in Acoustics, Speech, and Signal Processing, IEEE International Conference on. 1991. IEEE Computer Society.

[15] Kvedalen, E., Signal processing using the Teager Energy Operator and other nonlinear operators. Master, University of Oslo Department of Informatics, 2003. 21.

[16] Feldman, M., & Braun, S. (1997). Description of free responses of SDOF systems via the phase plane and Hilbert transform: The concepts of envelope and instantaneous frequency. In Proceedings of SPI, Orlando, Florida (Vol. 3089, pp. 973–979).

[17] Dr. Adrian Rosebrock, Deep Learning for Computer Vision with Python: Starter Bundle. [United States] : PyImageSearch, ©2017, 2017.

[18] Senani SAMY, « RESEAU DE NEURONES CONVOLUTIONNEL POUR LA DETCTION PRECORE DE LA RETINOPATHIE DIABETIQUE », Université Mouloud Mammeri de Tizi-Ouzou, 2019.

[19] L.Bergman et Y. Hoshen, « CLASSIFICATION-BASED ANOMALY DETECTION FOR GENERAL DATA », Comput. Sci. Mach. Learn., 2020, [En ligne]. Disponible sur: https://arxiv.org/abs/2005.02359.

[20] A. Géron, Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow, vol. 53, no 9. 2019.

[21] J. Nordby, « Environmental Sound Classification on Microcontrollers using Convolutional Neural Networks », 2019.

[22] Y. LeCun, Y. Bengio, et G. Hinton, « Deep learning », Nat. 2015 5217553, vol. 521, no 7553, p. 436-444, mai 2015, doi: 10.1038/nature14539.

[23] Yann LeCun et al. Generalization and network design strategies. Connectionism in perspective, 19 :143–155, 1989.

[24] Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, Gang Wang, Jianfei Cai, et al. Recent advances in convolutionalneural networks. Pattern Recognition, 77 :354–377, 2018.

[25] Yamashita, R., et al., Convolutional neural networks: an overview and application in radiology. Insights into imaging, 2018. 9(4): p. 611-629.

[26] Yamashita, R., et al., Convolutional neural networks: an overview and application in radiology. Insights into imaging, 2018. 9(4): p. 611-629.

[27] Lin M, Chen Q, Yan S (2013) Network in network. arXiv. Available online at: https://arxiv.org/pdf/1312.4400.pdf. Accessed 30/05/2022

[28] K. E. Koech, « Softmax Activation Function — How It Actually Works », 2020.

[29] https://www.python.org Consultez le 30/05/2022.

[30] https://www.tensorflow.org/, consulté le 28/05/2022.

[31] http://KERAS.io consulté le 28/05/2022.

[32] https://keras.io/about/, consulté le 30/05/2022.

[33] https://www.electronicwings.com/raspberry-pi/raspberry-pi-introduction.

[34] https://raspberrypi.dk/en/product/raspberry-pi-3-model-b-plus-2, consulté le 30/05/2022

[35] Wouters, N., EMBEDDING DEEP NEURAL NETWORKS FOR URBAN SOUND RECOGNITION ON AN FPGA.

[36] https://www.mathworks.com/discovery/what-is-matlab.html, consulté le 30/05/2022.

[37] O. 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. », UNIVERSITE DE REIMS CHAMPAGNE ARDENNE ECOLE, 2013.

[38] A. CEBAN, « Méthode globale de diagnostic des machines électriques », Université d'Artois, 2012.

[39] Rosebrock, A., Deep Learning for Computer Vision with Python: Starter Bundle2017: PyImageSearch.