

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي

UNIVERSITE BADJI MOKHTAR - ANNABA
BADJI MOKHTAR – ANNABA UNIVERSITY



جامعة باجي مختار – عنابة

Faculté : TECHNOLOGIE

Département : Electronique

Domaine : SCIENCES ET TECHNOLOGIES

Filière : AUTOMATIQUE

Spécialité : AUTOMATIQUE ET INFORMATIQUE INDUSTRIELLE

Mémoire

Présenté en vue de l'obtention du Diplôme de Master

Thème:

**Diagnostic des défauts de roulement basé sur les réseaux
de neurones BILSTM**

Présenté par :

- REDJATI Meriem Takoua
- RACHID Kaouter

Encadrant : BOUGHAZI Mohamed

Professeur

UBMAnnaba

Jury de Soutenance :

BOULEBTATECHE Brahim	MCB	U.B.M.Annaba	Président
BOUGHAZI Mohamed	Professeur	U.B.M.Annaba	Encadrant
KHERFANE Hamid	Professeur	U.B.M.Annaba	Examineur

Année Universitaire : 2022/2023

Dédicaces

Je dédie ce modeste travail :

*A mes chers parents qui ont été toujours présents,
merci de fond du cœur pour leurs déroulement, leurs
disponibilité et leurs affection tout au long de mes
études.*

À mes chers frères.

*A tout mes amis(es) et mes collègues de la promotion de
l'automatique, spécialement.*

Meriem

Dédicace

Je dédie ce travail à mes très chers parents pour leur soutien indéfectible. Que dieu les protège.

*À mes chères sœurs Rahma, Salsabil, Asma,
Pichou.*

À mon frère Adem Abd el kader.

À toute ma famille et à ceux qui ont cru en moi.

À tous mes amis du plus proche au plus loin.

*À tous personne qui occupe une place dans mon
cœur.*

À tous mes enseignants, leur générosité et leur soutien. M'obligent à leur exprimer notre profond respect et notre loyale considération.

Kaouther

Résumé

La détection précoce des défauts de roulements permet d'augmenter la durée de vie des moteurs et ainsi minimiser les pertes financières. A cet effet, nous proposons dans Ce mémoire présente une étude approfondie sur la détection des défauts de roulement des moteurs à l'aide de la méthode basée sur les réseaux neuronaux BILSTM (Mémoire à long terme bidirectionnelle).

La méthode proposée exploite les capacités des réseaux neuronaux BILSTM pour analyser les données de vibration des moteurs et identifier les signaux caractéristiques associés aux défauts de roulement. L'utilisation de ces réseaux pour la détection des défauts de roulement des moteurs présente un potentiel prometteur. Cette approche permet d'améliorer la maintenance prédictive des moteurs en permettant une détection rapide et précise des défauts de roulement.

Mots clés : *Défauts de roulements, Diagnostic de défauts, Machines tournantes, Analyse vibratoire, Moteur, Réseau neuronaux BILSTM.*

Abstract

Early detection of bearing faults can extend motor life and minimize financial losses. To this end, we propose in this thesis presents an in-depth study of motor bearing fault detection using the BILSTM (Bidirectional Long Short-Term Memory) neural network-based method.

The proposed method exploits the capabilities of BILSTM neural networks to analyze motor vibration data and identify characteristic signals associated with bearing faults. The use of these networks for the detection of motor bearing faults shows promising potential. This approach improves predictive maintenance of motors by enabling fast and accurate detection of bearing faults.

Key words: *Bearing defects, Defect Diagnosis, Rotary Machines, Vibration Analysis, Motor, and Neural Network BILSTM.*

ملخص

يتيح اكتشاف الحوادث في وقت مبكر زيادة عمر المحركات وبالتالي تقليل خسائر مالية. لهذا الغرض، ونحن نقدم في هذه المقالة دراسة مفصلة حول اكتشاف أخطاء تشغيل المحركات باستخدام الطريقة القائمة على الشبكات العصبية ذاكرة طويلة الأجل الثنائية.

يستخدم المنهج الموصى به القدرات من الشبكات العصبية الذاكرة طويلة الأمد لتحديد بيانات التوترات المحملة وتحديد الإشارات المميزة المرتبطة بالخلافات في التشغيل. إن استخدام هذه الشبكات لتحديد أية نقص في تشغيل المحركات لديها إمكانيات رائعة. يساعد هذا النهج على تحسين صيانة المحركات المتوقعة من خلال إمكانية اكتشاف الفشل السريع والواضح.

الكلمات المفتاحية: العيوب في الحركة، والتشخيص الناقص، وآلات تحول، وتحليل الاهتزازات، محرك، و الشبكة العصبية للذاكرة القصيرة الأجل الطويلة الأجل.

Liste des figures

Figure 1.1 – Composants de la machine tournante.

Figure 1.2 – Moteur asynchrone.

Figure 1.3– Stator d'une machine asynchrone.

Figure 1.4 – Rotor à cage d'écureuil.

Figure 1.5 – Aperçu des différentes composantes de la machine asynchrone à cage.

Figure 1.6 – Constitution d'un roulement à bille.

Figure 1.7 – Principaux types de roulements.

Figure 1.8 – Exemples de défaillances affectants les roulements ; a: grippage, b:fractur, c:usure de contact, d: corrosion de contact, e: écaillage de bague interne.

Figure 1.9 – Différentes types d'engrenages utilisés dans l'industrie ; a:engrenage droit a denture droite, b: engrenage droit à denture hélicoïdale, c:engrenage conique, d:engrenage roue et vis sans fin.

Figure 1.10 – Défauts de piqures ou putting.

Figure 1.11 – Défauts de grippage.

Figure 1.12 – Défauts d'engrenages ; a : écaillages, b : fissuration de dent, c : rupture de dents.

Figure 1.13 – Défaut de balourd.

Figure 1.14 – Défauts d'alignement.

Figure 1.15 – Différentes méthodes d'analyse.

Figure 1.16 – Technique infrarouge appliqué au moteur électrique.

Figure 1.17 – Exemple diagnostic avec analyse des vibrations.

Figure 2.1 – Schéma des différents cas d'utilisation pour un type d'entraînement donnée.

Figure 2.2 – Machine Learning versus Deep Learning.

Figure 2.3- Structure d'un neurone biologique.

Figure 2.4- L'architecture d'un réseau de neurones artificiels.

Figure 2.5- La structure d'un neurone artificiel.

Figure 2.6- Le schéma de CNN.

Figure 3.1 – les portes LSTM.

Figure 3.2 – LSTM architecture.

Figure 3.3- Structure de base d'un modèle (BILSTM).

Figure 3.4- Architecture (BILSTM) pour la classification.

Figure 4.1- Matlab Logo.

Figure 4.2- Segmentation imbriquée des signaux de vibration.

Figure 4.3 : Interfaçage proposé.

Figure 4.4- Résultat d'apprentissage du modèle (BILSTM).

Figure 4.5 – Matrice de confusion pour un des folds pour les 15 classes(BILSTM).

Figure 4.6- Prédiction des différents échantillons des 15 classes pour un fold (BILSTM).

Figure 4.7 – Matrice de confusion globale(BILSTM) pour les 15 classes.

Figure 4.8- Prédiction des différentes 15 classes(BILSTM).

Figure 4.9- Résultat d'apprentissage du modèle (GRU).

Figure 4.10 – Matrice de confusion pour un des folds pour les 15 classes(GRU).

Figure 4.11- Prédiction des différents échantillons des 15 classes pour un fold(GRU).

Figure 4.12 – Matrice de confusion globale(GRU) pour les 15 classes.

Figure 4.13 – Prédiction des différentes 15 classes(GRU).

Figure 4.14 – Résultat d'apprentissage du modèle (BILSTM)14 classes.

Figure 4.15 – La matrice de confusion globale pour les 14 classes.

Figure 4.16 – Prédiction des différent 14 classes.

Liste des tableaux

Tableau 1.1 – Résumé des modes de défaillances.

Tableau 4.1 – Base de données.

Tableau 4.2 – Validation croisée en 5 folds.

Tableau 4.3– Résultat d'apprentissage du modèle BILSTM pour 15 classes.

Tableau 4.4– Résultat d'apprentissage du modèle BILSTM pour 14 classes.

Tableau 4.5 – Tableau de comparaison des métriques.

Tableau 4.6 – Résultat d'apprentissage pour les 14 classes.

Tableau 4.7- Comparaison de la précision de la classification sur l'ensemble de données portant sur l'université Case Western Reserve.

Tableau 4.8- Comparison of test results for different learning rates and batch sizes (Optimizer = Adam and Sgdm).

Table des matières

Dédicace	
<u>Meriem</u>	2
Kaouther	2
Liste des figures.....	6
Liste des tableaux	8
INTRODUCTION GENERALE.....	12
Chapitre I : La machine asynchrone et le contexte de recherche	
La machine asynchrone et le contexte de recherche.....	13
I. Introduction.....	14
I.1 Machines tournantes	14
I.2 Moteur asynchrone.....	15
I.2.1 Eléments d'un moteur asynchrone	15
I.3 Défauts de roulements.....	17
I.3.1 Défauts d'Engrenage	19
I.3.2 Défauts localisés.....	20
I.3.3 Le Balourd.....	21
I.3.4 Défauts d'alignement :	22
I.4.1 Les méthodes Classiques.....	24
I.4.2 La Thermographie Infrarouge	24
I.4.3 L'analyse vibratoire.....	25
I.5 Conclusion	26
Chapitre II: Les Réseaux de Neurone Artificiel	
II.1 Introduction	28
II.2 Définition de l'apprentissage automatique (Machine Learning)	28
II.3 Type d'apprentissage	29
II.4 Définition de l'apprentissage en profondeur (Deep Learning).....	29
II.5 L'Apprentissage automatique vers apprentissage profond.....	30
II.6 Réseaux de Neurones.....	31
II.6.1 Neurone biologique.....	31
II.6.2 Principe des réseaux de neurones (Neural Network).....	32
II.6.3 Fonctionnement des Réseaux de Neurones.....	32
II.7 Types de Réseaux de Neurones :	32
II.7.1 Réseaux de Neurones Artificiels (ANN)	33
II.7.1.1 Architecture des réseaux de neurones artificiels	33
II.7.1.2 Le Perceptron	34
II.7.2 Réseaux de Neurones Récurents (RNN)	35

II.7.2.1 Les Réseaux de Hopfield.....	36
II.7.2.2 LSTM	36
II.7.2.3 GRU	37
II.7.2.4 JANET.....	37
II.7.2.5 La machine de Boltzmann profonde (DBN ou Deep Belief Network).....	37
II.7.3 Réseaux de Neurones à Convolution (CNN).....	38
II.7.3.1 Les Couches CNN	38
II.7.3.2 L’algorithme de rétro propagation	39
II.8 Conclusion	39
Chapitre III: <i>Modèles de réseaux de neurones récurrents BILSTM et GRU</i>	
III.1 Introduction	42
III.2 Les Métriques	42
III.3 Long Short-Term Memory (LSTM).....	44
III.4 L'architecture d'un modèle LSTM.....	45
III.4.1 Poids de LSTM (Weights).....	45
III.4.2 Portes de LSTM (Gated).....	45
III.5 Modèle Bidirectionnel LSTM	47
III.6 Modèle GRU (Gated Recurrent Unit)	49
III.6.1 Structure d’un modèle GRU	49
III.7 Techniques d'optimisation.....	51
III.7.1 L’option ADAM	51
III.7.2 L’option SGD	52
III.8 Évaluation des Hyper-paramètres	53
III.8.1 Le choix du taux d’apprentissage	53
III.8.2 Le choix de la taille du mini-batch :	53
III.8.3 Le choix du nombre epochs	53
III.8.4 Le choix du taux de dropout	54
III.8.5 Le choix de nombre d’unités cachées LSTM	54
III.9 Conclusion.....	54
Chapitre IV : Résultats et Discussion	
IV.1 Introduction.....	55
IV.2 Outils du travail (MATLAB):	55
IV.3 Description de la base de données :	56
IV.4 Phase préparation des données :.....	57
IV.5 Interfaçage proposé	57
IV.6. Diagnostic des défauts de roulements par les deux méthodes proposées pour 15 classes	58
IV.6.1 Les mesures d'évaluation.....	58

IV.6.2 Modèle BILSTM	60
IV.6.3 Validation croisée en 5 Folds	60
IV.6.3 Modèle GRU :	64
IV.6.4 Comparaison entre la méthode BILSTM et la méthode de GRU.....	68
IV.6.5 Pour les 14 classes	69
IV.6.6 Comparaison avec d'autres méthodes.....	71
IV.7 Recherche du modèle optimal.....	72
IV.8 Conclusion	73
CONCLUSION GENERALE	74
<i>Bibliographie Générale</i>	75

INTRODUCTION GENERALE

Les machines asynchrones, également connues sous le nom de moteurs asynchrones, sont des dispositifs électromécaniques couramment utilisés dans de nombreuses applications industrielles et domestiques. Ces moteurs jouent un rôle essentiel dans la conversion de l'énergie électrique en énergie mécanique, permettant ainsi le fonctionnement de divers équipements et machines.

Cependant, malgré leur fiabilité et leur efficacité, les moteurs asynchrones sont sujets à des défauts de roulement, qui peuvent entraîner des problèmes de performance et de durabilité. Les roulements constituent des composants essentiels des machines tournantes, assurant la liaison mobile en rotation entre les différentes parties de la machine avec un minimum de frottement. L'intégration de l'intelligence artificielle et des réseaux de neurones dans l'analyse et la maintenance des moteurs asynchrones offre de nouvelles perspectives pour la détection précoce et la prévention des défauts de roulement. L'intelligence artificielle permet d'exploiter des modèles prédictifs et des algorithmes d'apprentissage automatique pour surveiller en temps réel les paramètres de fonctionnement des moteurs et identifier les signaux précurseurs de problèmes de roulement.

Les réseaux de neurones, en particulier, jouent un rôle crucial dans l'analyse des vibrations, des signaux électriques et d'autres données provenant des moteurs asynchrones. Ces réseaux sont capables d'apprendre à partir de grands ensembles de données, d'identifier des schémas complexes et de détecter les anomalies dans le comportement des roulements. Grâce à leur capacité d'adaptation et à leur précision, les réseaux de neurones contribuent à améliorer la maintenance prédictive et à réduire les coûts liés aux pannes et aux arrêts imprévus.

En résumé, l'intégration de l'intelligence artificielle et des réseaux de neurones ouvre de nouvelles perspectives dans la détection précoce et la prévention des défauts de roulement des moteurs asynchrones. Cette combinaison de technologies permet d'améliorer la fiabilité, la performance et la durabilité de ces machines, en offrant des solutions avancées de maintenance prédictive et en réduisant les interruptions coûteuses de production.

Chapitre I

La machine asynchrone et le contexte de recherche

I. Introduction

Les machines tournantes jouent généralement un rôle stratégique dans le processus de fabrication, ce qui nécessite la sélection d'un bon processus pour maintenir les machines industrielles en bon état de fonctionnement et prolonger leur durée de vie, économisant ainsi beaucoup de temps et d'argent. Pour cette raison, la compréhension des pièces mécaniques et de leurs défaillances peut aider à atteindre cet objectif. Les moteurs sont des machines qui convertissent l'énergie en mouvement. Ils sont utilisés dans une variété d'applications aux équipements industriels et aux outils électriques. Les roulements sont des composants clés des moteurs, car ils permettent aux arbres et aux autres pièces mobiles de tourner en douceur. Ce chapitre présente les défauts de roulements des machines tournantes, et les différentes méthodes utilisées au diagnostic de ces défauts.

I.1 Machines tournantes

En générale une machine tournante comprend essentiellement une partie tournante le rotor et une partie fixe le stator (Figure 1.1). Et pour qu'il y ait une rotation facilitée, il faudrait des guidages et liaisons en rotation (coussinet, roulements), Dans le domaine des machines électriques, il existe quelques principes généraux de fonctionnement. La classification classique fait apparaître les grandes familles suivantes [1].

- Machines à courant continu.
- Machines synchrones et alternateurs.
- La pompe.
- La turbine.

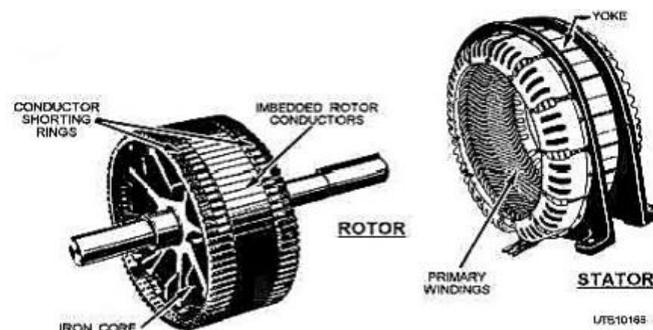


Figure 2.1 : Composants de la machine tournante.

I.2 Moteur asynchrone

Le moteur asynchrone, également connu sous le nom de moteur à induction, est largement utilisé dans de nombreuses applications industrielles et domestiques. Son fonctionnement repose sur le principe de l'induction électromagnétique entre le stator et le rotor. Le moteur asynchrone se compose d'un stator fixe et d'un rotor monté sur un axe de rotation. Le stator est constitué de tôles d'acier au silicium avec des encoches pour les enroulements, reliés à la source d'alimentation. Le rotor peut être de deux types : à enroulements bobinés ou à cage d'écureuil, selon qu'il est accessible de l'extérieur ou fermé en permanence sur lui-même.

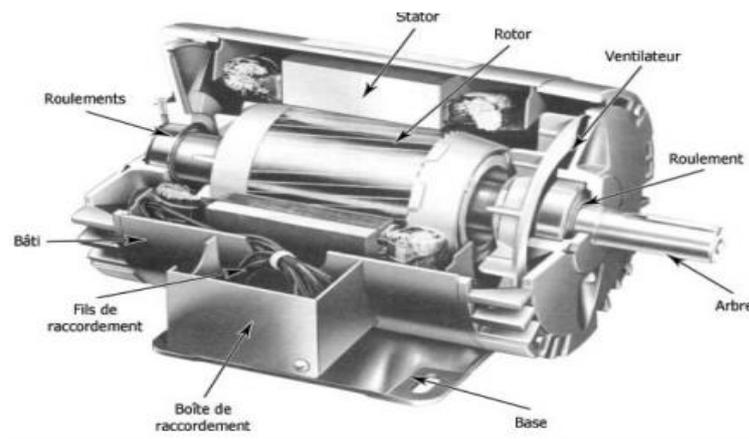


Figure 1.3: Moteur asynchrone.

La vitesse de rotation du rotor d'un moteur asynchrone est légèrement inférieure à celle du champ magnétique du stator, d'où son appellation "asynchrone". Cette différence de vitesse est appelée "glissement" et est généralement exprimée en pourcentage.

Les moteurs asynchrones sont couramment utilisés dans les pompes, les ventilateurs, les compresseurs, les machines-outils et d'autres applications industrielles. Ils sont appréciés pour leur construction simple, leur robustesse, leur coût abordable, leur faible besoin d'entretien et leur capacité à fonctionner dans des conditions variables.

I.2.1 Éléments d'un moteur asynchrone

I.2.1.1 Le Stator

Le stator comporte une carcasse en acier renfermant un empilage de tôles minces identiques en forme de couronne qui constituent un cylindre vide ; ces tôles sont percées de trous à leur périphérie intérieure. L'alignement de ces trous forme des encoches dans lesquelles on loge un bobinage triphasé. Cette couronne est serrée dans une carcasse en fonte [2].

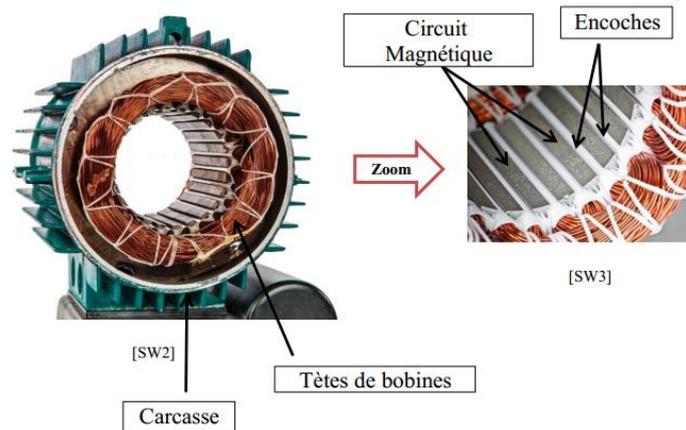


Figure 1.3: Stator d'une machine asynchrone.

I.2.1.2 Le Rotor

L'enroulement du rotor à cage d'écureuil est constituées de barres de cuivre nues introduites dans les encoches ; ces barres sont soudées ou rivées à chaque extrémité à deux anneaux qui les court-circuitent. L'ensemble ressemble à une cage d'écureuil d'où le nom de rotor à cage d'écureuil. Dans les moteurs de petite moyenne puissance, les barres et les anneaux sont formés d'un seul bloc d'aluminium coulé [2].

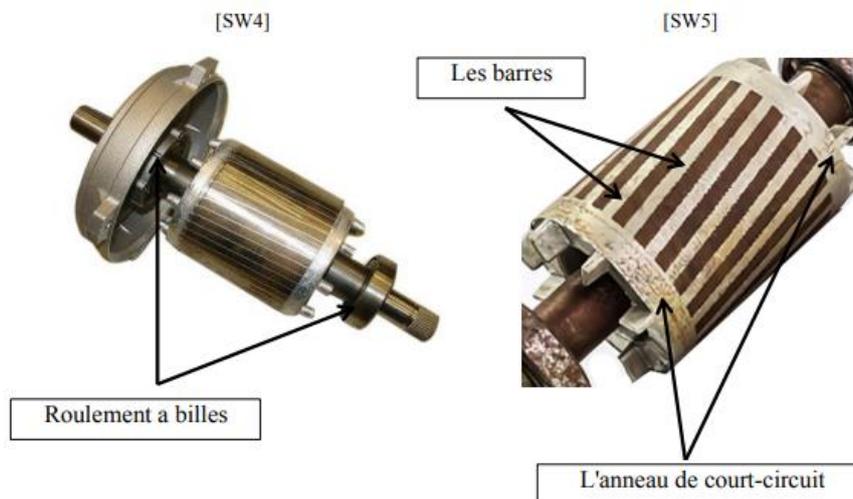


Figure 1.4: Rotor a cage d'écureuil.

I.2.1.3 Organes mécaniques

La carcasse sert de support. Elle joue le rôle d'enveloppe et assure la protection contre l'environnement extérieur. L'arbre est un organe de transmission. Il comprend une partie centrale qui sert de support au corps du rotor et un bout d'arbre sur lequel est fixé un demi-accouplement [3].

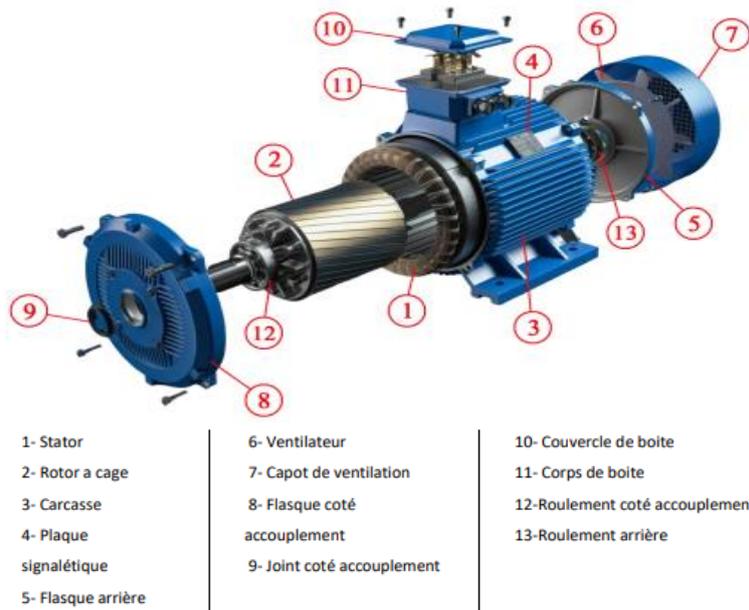


Figure 1.5: Aperçu sur les différentes Composantes de la Machine Asynchrone à Cage.

I.3 Défaits de roulements

Selon une étude statistique réalisée par Bonnett en 2008, il a été révélé que 69% des défaillances observées sur les moteurs asynchrones de grande puissance étaient attribuables à des problèmes de roulements [4]. Cela démontre que les roulements sont les éléments les plus importants et les plus fragiles des machines tournantes. Leur fonction est de remplir plusieurs rôles essentiels, tels que permettre la liaison mobile en rotation entre deux surfaces avec un minimum de frottement, supporter différentes charges radiales ou axiales, et corriger les défauts d'alignement en assurant un positionnement précis de l'arbre.

Un roulement est généralement composé de quatre éléments essentiels : une bague extérieure, une bague intérieure, un liquide lubrifiant et des éléments roulants tels que des billes, des rouleaux ou des aiguilles. Dans la pratique, le choix du roulement approprié dépend du type d'application, chaque classe de roulement ayant des caractéristiques spécifiques en termes de vitesse limite de rotation, de capacité à supporter des charges radiales et axiales, ainsi que de tolérance aux défauts d'alignement.

La figure 1.6 montre les différents composants d'un roulement à bille et les principaux types de roulements sont illustrés dans la figure 7.

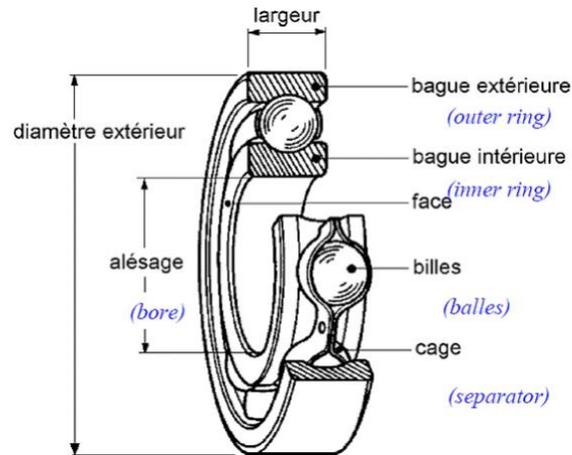


Figure 1.6 : Constitution d'un roulement à bille.

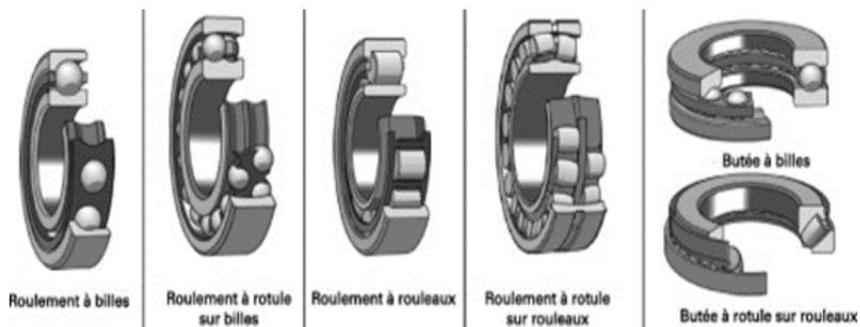


Figure 1.7 : Principaux types de roulements.

On peut distinguer plusieurs avaries affectant les roulements, et causées par quatre grandes familles de causes, qu'on peut les répertoriées comme suit [5-6] :

- Dégradations liées à une mauvaise lubrification, couvrant 36% des cas : quel que soit un manque ou un excès de lubrifiant qui peut produire une augmentation de la température du roulement, conduisant à une rupture des bagues, écaillage, grippage et détérioration des cages.
- Dégradations liées à des contraintes mécaniques excessives, couvrant 34% des cas : comme la vitesse, les charges radiales et axiales exercées sur les points de contact entre les éléments roulants et les bagues, pouvant causer un écaillage, détérioration des cages et grippage.
- Dégradations liées au montage incorrect, couvrant 16% des cas : comme les défauts d'alignement, traces de coups, empreintes des corps roulants, corrosion par contact et certaines détériorations de cages.
- Dégradations liées au milieu industriel difficile, couvrant 14% des cas : comme l'environnement humide et la présence des acides causant la corrosion du roulement, la

présence des polluants de nature abrasive comme la poussière et le sable, qui contaminent le lubrifiant, dégradent la surface de roulement et causant son usure.



Figure 1.8 : Exemples de défaillances affectants les roulements ; a: grippage, b:fractur, c:usure de contact, d: corrosion de contact, e: écaillage de bague interne.

I.3.1 Défauts d'Engrenage

Les engrenages jouent un rôle essentiel en tant qu'éléments fondamentaux dans la composition d'une machine en rotation. Ils sont formés de deux roues dentées, dont l'une entraîne l'autre grâce aux dents en contact, et tournent autour d'un axe à des vitesses différentes. Leur fonction principale est de transmettre le mouvement de rotation d'un arbre à un autre, de changer la direction du mouvement rotatif et même d'ajuster les vitesses de rotation [7].

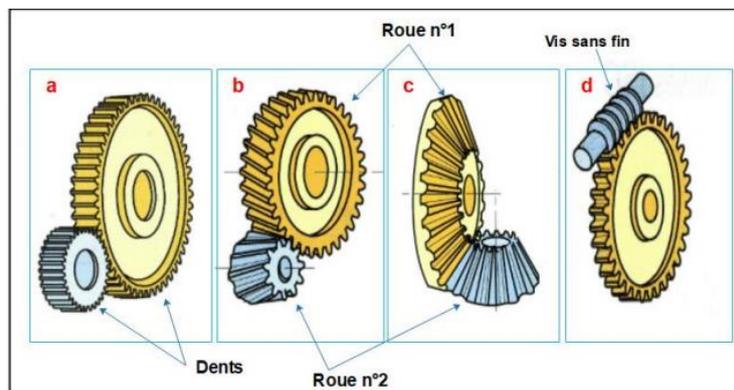


Figure 1.9: Différentes types d'engrenages utilisés dans l'industrie ; a:engrenage droit a denture droite, b: engrenage droit à denture hélicoïdale, c:engrenage conique, d:engrenage roue et vis sans fin.

En excluant les défauts liés à la fabrication et au montage (tels que l'excentricité des roues et les défauts de parallélisme), on peut principalement distinguer deux catégories de défauts pouvant altérer le bon fonctionnement des engrenages : les défauts généralisés et les défauts localisés.

Les défauts généralisés sont ceux qui affectent une grande partie ou la totalité des dentures d'engrenage. Parmi ces défauts, on trouve l'usure abrasive, qui survient lorsque les surfaces de contact des dents sont progressivement usées par le frottement, ainsi que les piquûres (ou pitting), qui sont des petites dégradations localisées sur les surfaces des dents causées par des contraintes répétées [8].

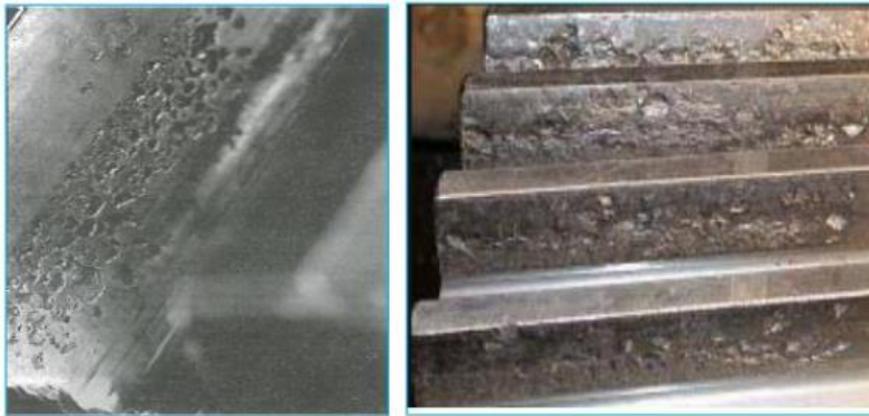


Figure 1.10: Défauts de piquures ou putting.

1.3.2 Défauts localisés

Sont généralement des défauts qui se localisent sur des dents particulières et ils peuvent conduire rapidement à la rupture de ces dents, comme l'écaillage, le grippage et la fissuration.



Figure 1.11: Défauts de grippage.

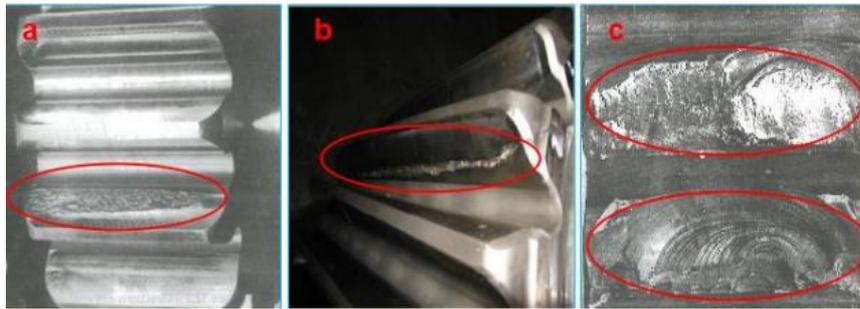


Figure 1.12: Défauts d'engrenages: **a.** Ecaillages, **b.** Fissuration de dent, **c.** Rupture de dents.

1.3.3 Le Balourd

Le balourd, également connu sous le nom de défaut d'équilibrage, est la principale cause de vibrations rencontrées fréquemment dans les machines tournantes. Il résulte d'une répartition inégale de la masse autour de l'axe d'un rotor. En pratique, il est impossible de faire coïncider l'axe de rotation avec le centre de gravité de chaque section élémentaire du rotor lors de la construction des machines. Les causes du déséquilibre, ou balourd, sont multiples et peuvent inclure :

- Défauts d'usinage, d'assemblage des rotors ou de montage.
- Balourd évolutif : ce phénomène se produit généralement en raison des frottements entre l'arbre et l'un des paliers de la machine tournante, ce qui entraîne un chauffage asymétrique.
- Origines mécaniques : telles que la dégradation des pales ou aubes, l'érosion ou le dépôt de matière sur le rotor, etc.
- Origines thermiques : où les variations de température modifient la géométrie des parties en rotation, entraînant une déformation des rotors, des fissures transversales de l'arbre, une asymétrie de l'arbre, un glissement et une dilatation contrariée des bobinages des machines électriques.

La maintenance régulière, la surveillance de l'état des machines et la correction du balourd contribuent à assurer un fonctionnement optimal, une durée de vie prolongée et une sécurité accrue des machines tournantes.

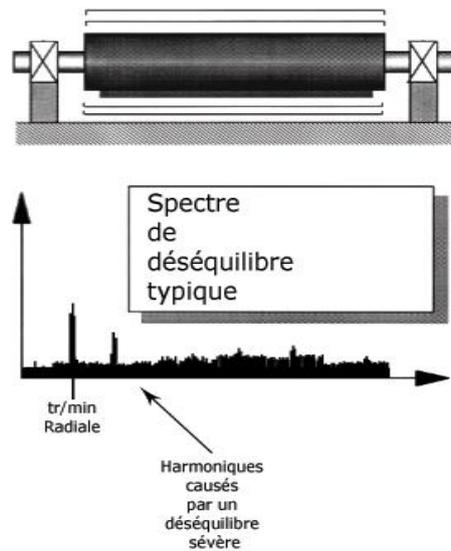


Figure 1.13 : Défauts de Balourd.

I.3.4 Défauts d'alignement :

L'arbre est un composant essentiel des machines tournantes, assurant la transmission de la puissance et du mouvement de rotation. Il joue également un rôle de support et de liaison pour les autres parties de la machine. Le défaut d'alignement, également appelé désalignement ou délignage, est un problème aussi fréquent que le balourd. Il se produit lorsque les axes de rotation de deux ou plusieurs arbres ne sont pas parfaitement alignés. Il entraîne une augmentation des forces radiales et axiales sur les accouplements, les roulements et les joints, ce qui provoque l'usure de ces composants et une flexion de l'arbre, réduisant ainsi la quantité de puissance transmise. Plusieurs facteurs peuvent être à l'origine d'un mauvais alignement de l'arbre :

- Avaries liées à des contraintes mécaniques : telles que le desserrage des pieds de fixation ou le déplacement des patins de la machine en raison des vibrations.
- Avaries liées à des contraintes thermiques : comme les variations de température et l'expansion thermique des pièces dues à la chaleur générée par le fonctionnement de la machine.
- Modifications mineures des paramètres de processus, telles que la pression.

Il est essentiel de détecter et de corriger le défaut d'alignement de l'arbre, car il peut entraîner une dégradation des performances, des vibrations excessives, une usure prématurée des composants et une diminution de l'efficacité de la transmission de puissance. Des techniques d'alignement précises, telles que l'utilisation de lasers ou de comparateurs d'alignement, peuvent être employées pour rétablir un alignement correct des arbres.

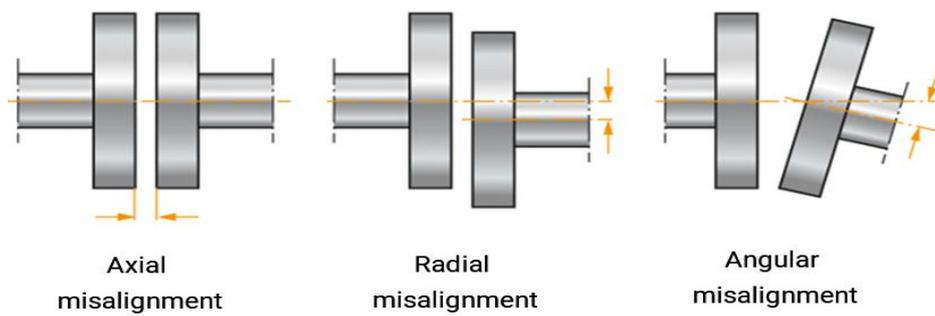


Figure 4: Défauts d'alignement.

Organe	Mode de défaillance
Roulement	<ul style="list-style-type: none"> - Blocage - Bruit - Ecaillage - Echauffement - Fuites - Grippage des bagues - Laminage - Rupture de la cage - Usure - Matage - Centrifugation

Tableau 1 : Résumé des modes de défaillances

I.4 Les Méthodes de Diagnostique

Il existe plusieurs méthodes de diagnostic des défauts mécaniques dans les machines tournantes, et qu'on peut les classer en deux grandes approches :

I.4.1 Les méthodes Classiques

La surveillance d'un équipement de machine implique de relever régulièrement un indicateur d'état de dégradation ou de performance. Différentes techniques d'analyse sont disponibles (FIGURE 1.7), telles que l'analyse vibratoire, l'émission acoustique, la thermographie, l'analyse des huiles et des lubrifiants, la variation de résistance électrique, etc.

Le choix de l'indicateur dépend du type de machine étudiée et du type de défaillance à détecter. Ces techniques permettent de détecter les signes précurseurs de problèmes, d'évaluer l'état de l'équipement et de prendre des mesures préventives ou correctives pour éviter les pannes coûteuses [9].

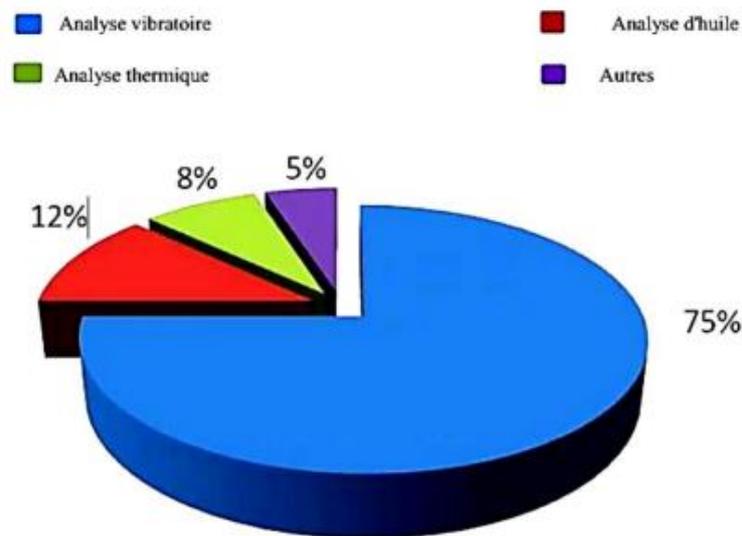


Figure 5: différentes méthodes d'analyse.

I.4.2 La Thermographie Infrarouge

La thermographie infrarouge repose sur le fait que tout corps dont la température est supérieure à 0°C émet un rayonnement électromagnétique. Ce rayonnement est directement lié à la température de surface du corps. La thermographie infrarouge consiste à mesurer ce rayonnement, à le quantifier et, si le système est correctement calibré, à créer une carte des températures à la surface de l'objet observé.

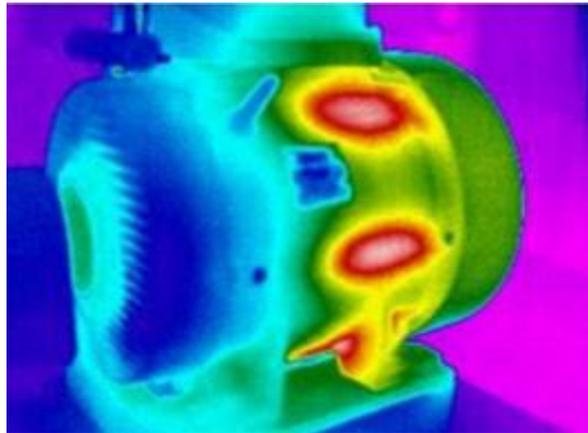


Figure 6: technique infrarouge appliqué au moteur électrique.

I.4.3 L'analyse vibratoire

Les vibrations des machines sont la réponse de leurs structures aux différentes forces auxquelles elles sont soumises. La surveillance des vibrations permet de détecter précocement l'apparition de forces perturbatrices, ce qui permet de prévenir les défaillances. En raison de son efficacité et de sa facilité d'installation, l'analyse des vibrations est l'une des techniques les plus utilisées pour la surveillance et le diagnostic des défauts des machines tournantes.

Lors de l'acquisition des informations vibratoires, il est essentiel de choisir les composants de la chaîne de mesure de manière à enregistrer des signaux contenant le plus d'informations vibratoires possible. Les techniques d'exploitation et de traitement des signaux vibratoires varient en fonction de la dynamique et de la cinématique de la machine surveillée, ainsi que de la nature des défauts suspectés. Dans les sections suivantes, nous décrirons successivement la chaîne de mesure des signaux vibratoires et les principales techniques d'exploitation. Un exemple de diagnostic basé sur l'analyse des vibrations est présenté dans la Figure 1.17 [10].

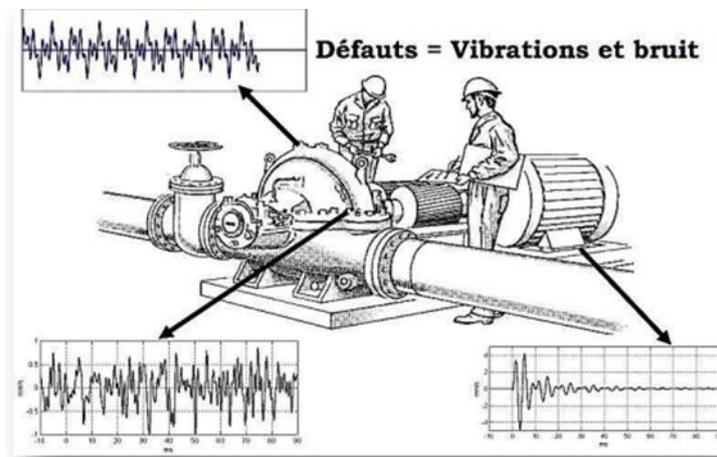


Figure 7: Exemple diagnostic avec analyse des vibrations.

I.5 Conclusion

Les roulements sont des composants essentiels des moteurs, permettant aux pièces mobiles de tourner en douceur et de manière fiable. Les différents types de roulements ont leurs avantages et leurs inconvénients, et il est important de choisir le bon type de roulement en fonction de l'application. En utilisant les bonnes combinaisons de roulements, les ingénieurs peuvent créer des moteurs performants et durables pour une variété d'applications.

Le diagnostic par différentes méthodes permet de détecter les défauts mécaniques pouvant affecter les différents composants des machines tournantes. C'est pourquoi les institutions œuvrant dans le domaine de la production industrielle ont l'intention de développer ces techniques de surveillance dans le but de réparer les pannes et de remettre la machine défectueuse dans son performance dans sa chaîne de travail.

Le travail de développement de ces technologies est passé par étapes de la phase classique à la phase d'intelligence artificielle lequel, à son tour de faciliter la tâche de diagnostic et augmente la précision du processus de surveillance, et avec l'avancement de la science dans le domaine de l'intelligence artificielle, les tâches de ces derniers augmentent et la dépendance à son égard augmente davantage dans le domaine du diagnostic

Dans ce chapitre, on a présenté un aperçu théorique sur les principaux défauts de roulement dans les machines tournantes et les différentes méthodes de diagnostic classiques.

Chapitre II

Les Réseaux de Neurone Artificiel

II.1 Introduction

L'objectif de la recherche en intelligence artificielle (IA) consiste à développer des systèmes informatiques capables de réaliser des tâches en utilisant des processus de réflexion similaires à ceux des êtres humains. Ainsi, le défi réside non seulement dans la compréhension de la pensée humaine, mais aussi dans sa modélisation et sa reproduction.

L'intelligence artificielle (IA) est devenue un sujet d'une grande importance tant dans les médias que dans les revues scientifiques, principalement en raison des nombreuses réalisations qui en ont découlé, dont beaucoup sont le fruit de progrès dans le domaine de l'apprentissage automatique. L'IA repose sur une approche d'apprentissage visant à reproduire, à travers une application, un système ou un processus, une partie de l'intelligence humaine. Parmi les exemples de systèmes d'intelligence artificielle, on peut citer la reconnaissance faciale et la perception visuelle. La machine Learning (ML), un sous-domaine de l'IA, utilise des réseaux neuronaux artificiels (ANN) pour imiter la manière dont les êtres humains prennent des décisions. La machine Learning permet aux ordinateurs de développer des modèles d'apprentissage de manière autonome, sans programmation préalable, en se basant sur de vastes ensembles de données. Juste en dessous se trouve le Deep Learning (DL), l'une des nombreuses approches de la machine Learning qui a connu un succès considérable ces dernières années. Le DL est caractérisé par des réseaux neuronaux profonds capables d'effectuer des tâches complexes de manière plus efficace et précise.

Dans ce chapitre, nous aborderons les différentes formes d'apprentissage automatique ainsi que les différents types de réseaux neuronaux. Nous commencerons par une présentation générale de ces concepts, puis nous nous concentrerons sur les réseaux de neurones artificiels (ANN), les réseaux de neurones récurrents (RNN) et les réseaux de neurones convolutifs (CNN).

II.2 Définition de l'apprentissage automatique (Machine Learning)

L'apprentissage automatique, en tant que sous-domaine de l'intelligence artificielle (IA), se focalise sur la création de systèmes capables d'apprendre et d'améliorer leurs performances en fonction des données qu'ils analysent. L'intelligence artificielle et l'apprentissage automatique sont souvent mentionnés conjointement, soulignant l'étroite relation entre ces deux domaines [1].

II.3 Type d'apprentissage

Les algorithmes de Deep Learning peuvent être divisés en deux catégories principales : l'apprentissage supervisé et l'apprentissage non supervisé.

- L'apprentissage supervisé utilise un ensemble de données d'apprentissage étiquetées, ce qui signifie que chaque exemple de données est associé à une étiquette qui indique sa classe ou sa valeur attendue. Il existe deux principaux types de modèles d'apprentissage supervisé : les modèles de régression, qui visent à prédire des valeurs continues, et les modèles de classification, qui cherchent à attribuer des exemples à des catégories discrètes prédéfinies.
- **L'apprentissage non supervisé**, en revanche, consiste à apprendre des modèles à partir de données non étiquetées. Dans ce cas, l'algorithme doit extraire automatiquement des structures ou des regroupements dans les données sans recevoir d'informations explicites sur les catégories attendues. L'apprentissage non supervisé vise à découvrir des relations intrinsèques et à organiser les données en fonction de leurs similarités ou de leurs motifs sous-jacents.

Ces deux types d'apprentissage, supervisé et non supervisé, jouent des rôles importants dans le domaine du Deep Learning, offrant des approches complémentaires pour extraire des informations utiles et effectuer des tâches d'analyse de données avancées.

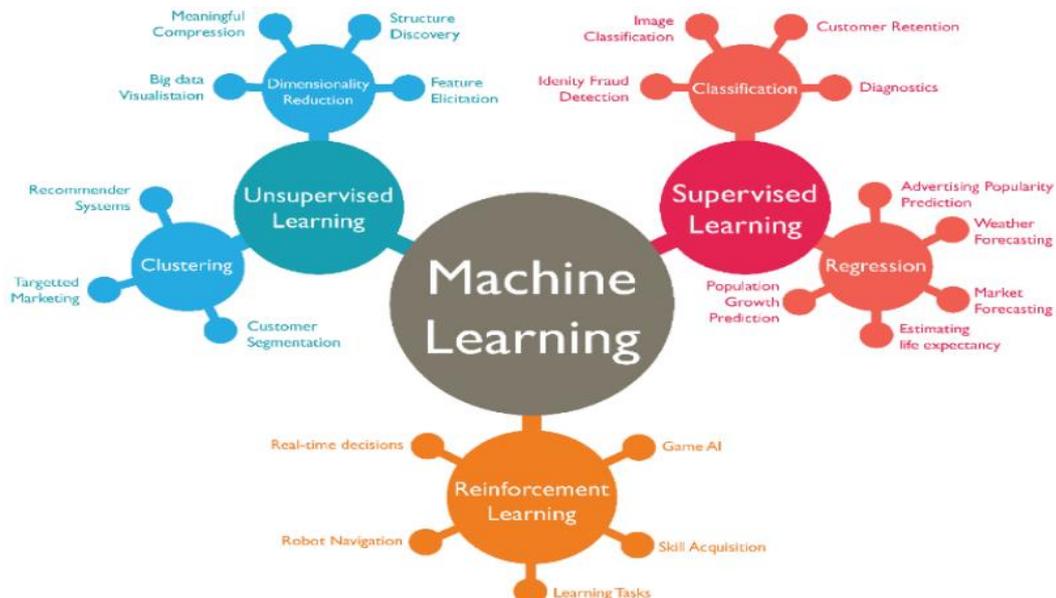


Figure 2.1: Schéma des différents cas d'utilisation pour un type d'entraînement donnée.

II.4 Définition de l'apprentissage en profondeur (Deep Learning)

L'apprentissage profond (*Deep Learning* ou *DL*) est une classe de techniques d'apprentissage automatique (*Machine Learning* ou *ML*) dans lesquelles les informations sont traitées en utilisant des couches hiérarchiques afin de comprendre les représentations et les caractéristiques des données à des niveaux de complexité croissants. L'apprentissage profond est également connu sous les noms d'apprentissage hiérarchique ou d'apprentissage profond structuré.

Tous les algorithmes d'apprentissage profond sont basés sur des réseaux de neurones (*Neural Networks*) qui partagent certaines propriétés fondamentales. Ces réseaux sont composés de neurones interconnectés et organisés en couches. Ce qui les distingue, c'est l'architecture spécifique du réseau, c'est-à-dire la manière dont les neurones sont organisés, et parfois la méthode utilisée pour leur apprentissage.

L'apprentissage profond a permis des avancées significatives dans des domaines tels que la vision par ordinateur, la reconnaissance vocale, la traduction automatique et de nombreux autres domaines de l'IA. Grâce à la puissance des réseaux de neurones profonds, il est possible d'extraire des représentations complexes et d'effectuer des tâches sophistiquées en exploitant les relations hiérarchiques et les caractéristiques latentes des données.

II.5 L'Apprentissage automatique vers apprentissage profond

La réussite de la plupart des méthodes d'apprentissage automatique (*Machine Learning*) repose sur les représentations et fonctions d'entrée créées par les humains. L'expertise humaine dans la conception des représentations et fonctions d'entrée est essentielle pour fournir des informations significatives au modèle d'apprentissage et guider l'optimisation des paramètres. Cependant, l'émergence de techniques telles que l'apprentissage profond a permis aux modèles d'apprentissage automatique d'apprendre directement des représentations et fonctions d'entrée à partir des données, réduisant la dépendance à l'égard d'une conception humaine préalable.

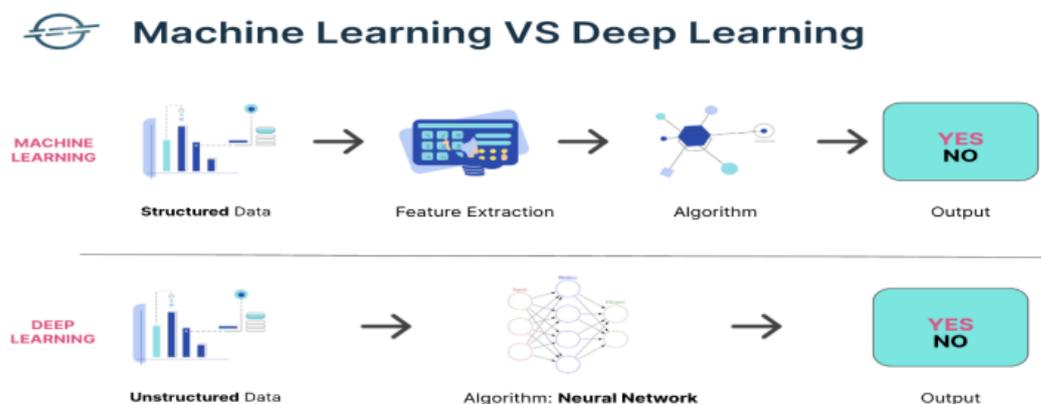


Figure.2.2: Machine Learning versus Deep Learning.

II.6 Réseaux de Neurones

II.6.1 Neurone biologique

De nombreuses inventions ont été inspirées par la nature, et les réseaux de neurones artificiels (ANN) ne font pas exception. Ils tirent leur inspiration de l'architecture du cerveau humain et des neurones biologiques qui le composent. Le cerveau humain est constitué d'environ 10 milliards de neurones, et chaque neurone est connecté à environ 10 000 autres neurones. Les neurones biologiques sont composés de corps cellulaires contenant un noyau, ainsi que de nombreuses extensions ramifiées appelées dendrites. Ils possèdent également une 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 d'autres neurones (voir figure 2.3) [2].

Cette organisation complexe du cerveau humain a été une source d'inspiration pour le développement des réseaux de neurones artificiels, qui tentent de reproduire ces connexions et ces interactions neuronales pour réaliser des tâches d'apprentissage et de traitement de l'information.

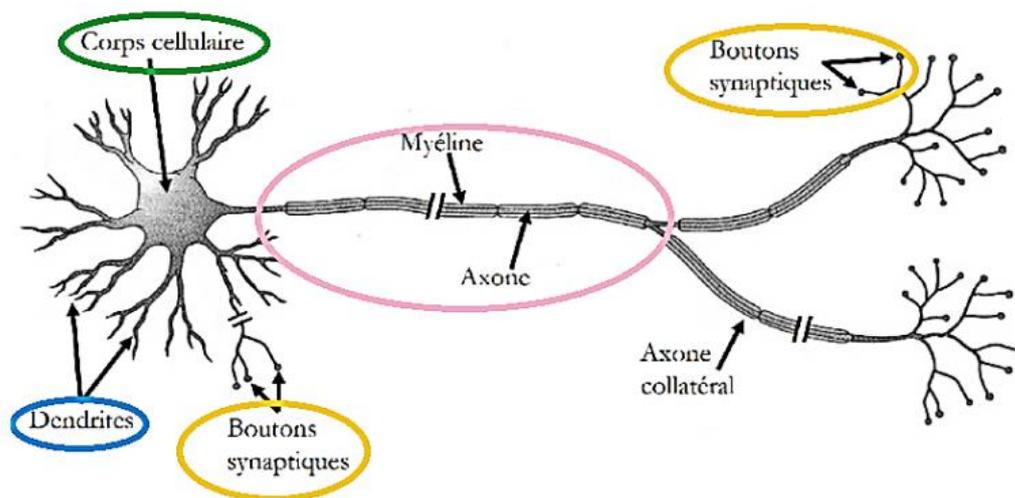


Figure.2.3: Structure d'un neurone biologique

Chaque neurone reçoit des signaux électrochimiques provenant d'autres neurones via ses dendrites. Si ces signaux entrants sont suffisamment forts pour activer le neurone, celui-ci est stimulé et transmet le signal le long de son axone, qui le relaie ensuite aux dendrites d'autres neurones. Ces neurones à leur tour peuvent être activés, prolongeant ainsi le processus de transmission du message.

Les réseaux de neurones artificiels cherchent à imiter ce processus en simulant les interactions entre les neurones et la propagation des signaux.

II.6.2 Principe des réseaux de neurones (Neural Network)

Un réseau de neurones est un outil puissant pour analyser les relations complexes entre les données d'entrée et de sortie. Il est basé sur le principe de l'apprentissage, où le réseau ajuste ses paramètres internes, appelés poids synaptiques, afin de capturer les modèles et connaissances présents dans les données. Grâce à l'apprentissage et à l'optimisation de ces poids synaptiques, le réseau peut généraliser et effectuer des prédictions précises sur de nouvelles données. En combinant de nombreux neurones interconnectés et en utilisant des techniques d'apprentissage, les réseaux de neurones sont capables de modéliser des relations complexes et de fournir des solutions efficaces à des problèmes variés.

II.6.3 Fonctionnement des Réseaux de Neurones

Le fonctionnement des réseaux de neurones repose sur la répartition des valeurs des variables dans des unités appelées neurones. Ces neurones ont pour responsabilité de combiner les informations qu'ils reçoivent les uns des autres afin de déterminer la valeur du coefficient de discrimination. Cette capacité de discernement dépend de la connectivité entre les neurones, c'est-à-dire des liens entre les unités chargées d'intégrer l'information. Chaque neurone reçoit des informations numériques de ses neurones voisins, qui sont pondérées par des poids représentant la force de la connexion. Des calculs sont ensuite effectués dans chaque neurone, et le résultat de ce processus est transmis aux neurones situés en aval.

Networks de neurones exploitent opérations mathématiques et fonctions d'activation pour traiter et analyser données.

II.7 Types de Réseaux de Neurones :

Il existe plusieurs architectures de réseaux de neurones couramment utilisées dans la pratique. Certaines des architectures les plus populaires sont les suivantes :

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

II.7.1 Réseaux de Neurones Artificiels (ANN)

Le réseau de neurones à propagation avant (feedforward neural network) est un des formes les plus simples d'ANN. Dans ce type de réseau, les données ou l'entrée se déplacent dans une seule direction, passant par les nœuds d'entrée pour atteindre les nœuds de sortie. Il peut être composé de couches cachées ou sans couche cachée. Le réseau propage simplement les données de manière linéaire à travers le réseau, sans rétroaction, en utilisant généralement une fonction d'activation pour introduire la non-linéarité dans le modèle [3]. Ce type de réseau est souvent utilisé pour des tâches de classification et de prédiction, où les données d'entrée sont transformées en une sortie correspondante à l'aide des poids et des fonctions d'activation du réseau.

II.7.1.1 Architecture des réseaux de neurones artificiels

Les réseaux neuronaux sont des structures sophistiquées composées de neurones artificiels qui reçoivent plusieurs entrées et génèrent une ou plusieurs sorties correspondantes (Figure 2.4).

Il se compose de :

- Couche Entrée (Input Layer) : c'est l'ensemble de neurones qui porte le signal d'entrée 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.

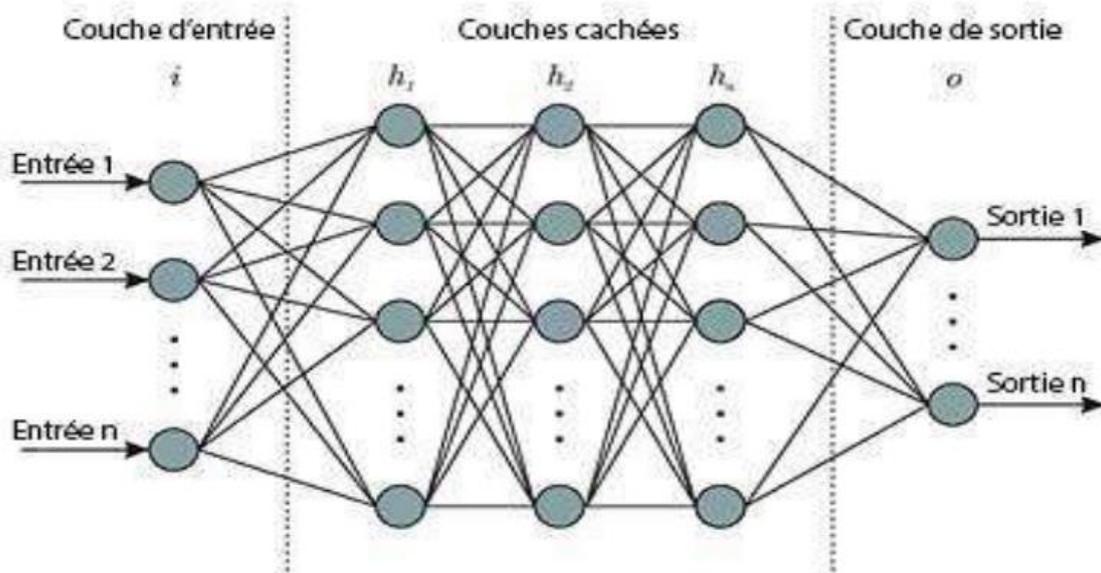


Figure 2.4: L'architecture d'un réseau de neurones artificiels.

II.7.1.2 Le Perceptron

Le perceptron est une composante essentielle des réseaux de neurones artificiels, représentant un modèle de neurone artificiel capable d'effectuer une fonction non linéaire algébrique. Il agit en tant que récepteur pour une ou plusieurs entrées, notées x_j et chaque entrée est individuellement pondérée par w_j . Les poids w_j déterminent l'importance relative des différentes entrées pour la sortie finale du perceptron. La somme pondérée des entrées est ensuite additionnée au biais du neurone b_j , puis utilisée comme argument pour une fonction d'activation f (voir Figure 2.5). L'utilisation d'une fonction d'activation introduit une non-linéarité dans le modèle, permettant ainsi au perceptron d'apprendre des relations complexes entre les entrées et la sortie.

Lorsqu'il s'agit de prédire avec un réseau de neurones, les données sont introduites en tant qu'entrées dans la première couche, et les calculs se propagent à travers toutes les couches successives jusqu'à atteindre les sorties finales. Ce processus est couramment appelé propagation vers l'avant ou propagation directe.

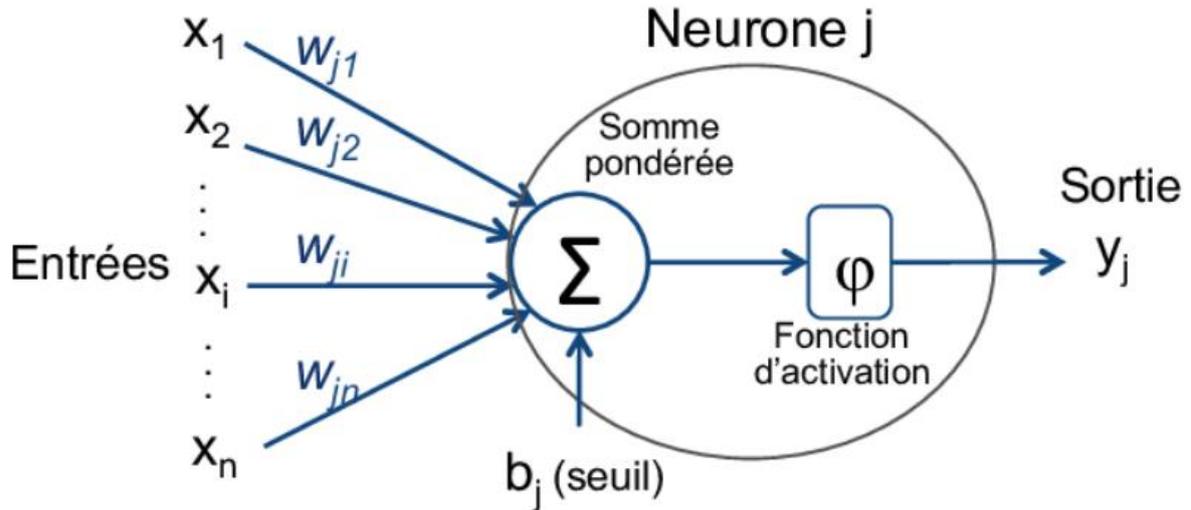


Figure 2.4 : La structure d'un neurone artificiel.

II.7.2 Réseaux de Neurones Récurrents (RNN)

Contrairement aux réseaux de neurones à propagation avant (Feed-Forward Neural Network - FFNN), les réseaux de neurones récurrents (Récurrent Neural Network - RNN) sont caractérisés par la présence d'au moins une boucle de rétroaction dans leur graphe de connexion. Au cours des 30 dernières années, plusieurs types de RNN ont été développés, tels que les réseaux d'Elman [4], les réseaux de Jordan [5] et les Echo State Networks [6]. Récemment, un type de RNN est devenu largement utilisé en raison de ses performances exceptionnelles dans diverses tâches : les réseaux de neurones à mémoire à court et long terme (Long Short-Term Memory - LSTM). Les LSTM jouent un rôle central dans cette thèse, et le chapitre 3 leur est consacré.

Dans cette partie, nous nous intéressons à la version la plus simple des RNN proposée par Jeff Elman en 1990. Dans cette approche, des liens sont ajoutés à un perceptron multicouche (Multi-Layer Perceptron - MLP), de sorte qu'une couche du réseau reçoive non seulement la sortie courante de la couche précédente, mais également sa propre sortie au pas de temps précédent. Cette modification a un impact considérable lorsque les entrées sont sous la forme de séquences temporelles ou spatiales, contrairement au cas où les vecteurs d'entrée x sont indépendants les uns des autres. En effet, la structure du RNN introduit un mécanisme de mémoire des entrées précédentes qui persiste dans les états internes du réseau et peut influencer toutes les sorties futures. En théorie, cette modification simple permet d'approximer avec une précision arbitraire n'importe quelle fonction qui transforme une séquence d'entrée en une séquence de sortie donnée, contrairement au MLP. Cependant, il est important de noter que les RNN de ce type

peuvent être particulièrement difficiles à entraîner (bien que des développements récents aient amélioré cette situation), et sans précautions particulières, le contexte utilisable reste généralement limité.

Parmi ces types de RNN nous citons quelques-uns :

II.7.2.1 Les Réseaux de Hopfield

Les réseaux de Hopfield sont un type de réseau de neurones récurrents utilisés dans des applications d'apprentissage non supervisé et dans la résolution de problèmes d'optimisation. Le réseau fonctionne comme une mémoire associative non linéaire, capable de récupérer et de reconnaître des motifs stockés dans un espace de données. L'idée principale derrière les réseaux de Hopfield est de créer des attracteurs dans l'espace de l'état du réseau. Ils sont largement utilisés pour des tâches telles que la récupération de la mémoire, la classification de motifs, la résolution de problèmes d'optimisation et la modélisation de systèmes dynamiques.

II.7.2.2 LSTM

Les cellules de mémoire à long terme court (LSTM) ont été proposées par Hochreiter et Schmidhuber en 1997. Elles ont été développées dans le but de résoudre le problème de l'apprentissage des dépendances à long terme dans les réseaux de neurones récurrents (RNN) traditionnels.

LSTM (Long Short Term Memory) est un des architectures récurrentes utilisées dans les réseaux de neurones artificiels. LSTM sont composés de couches de neurones qui sont connectées récurrentement, permettant à l'état précédent d'un neurone d'être utilisé comme contexte pour générer une sortie. Les cellules LSTM sont capables de mémoriser des informations sur des périodes de temps plus longues, ce qui permet de capturer des dépendances à long terme dans les séquences de données. Les LSTM ont connu un grand succès dans de nombreuses applications, notamment la traduction automatique, la génération de texte, la reconnaissance vocale.

La formation efficace des réseaux de neurones récurrents (RNN) est un des principaux défis auxquelles ils sont confrontés. Le problème principal est la propagation du gradient à travers les pas de temps, qui entraîne des problèmes de stabilité et de convergence lors de l'apprentissage. Pour résoudre ce défi, plusieurs techniques ont été développés, tels que la fonction d'activation adapté, les variantes de RNN et les techniques avancées d'optimisation. Grâce à ces avancées, il est possible de former efficacement les RNN et de les appliquant avec succès à une grande variété de tâches.

II.7.2.3 GRU

Le réseau GRU (Gated Recurrent Unit) est un réseau de neurones récurrents (RNN) composé de couches d'entrée, cachée et sortie. La couche d'entrée est constituée de segments de neurones déterminés par la dimension de l'espace des caractéristiques de l'entrée. La couche de sortie est déterminée par la taille de l'espace de sortie souhaitée. Les cellules GRU sont conçues pour réguler le flux d'information et contrôler l'oubli ou l'ajout de nouvelles informations à la mémoire interne du réseau. Les cellules GRU sont dotées de portes (gates) qui contrôlent l'accès à la mémoire et la mise à jour de son contenu. Le réseau GRU est largement utilisé dans des applications telles que la traduction automatique, la génération de texte, la reconnaissance de la parole.

II.7.2.4 JANET

Le Gated Recurrent Unit (GRU) est une variante de réseau récurrent qui propose une architecture simplifiée en supprimant certains ports du LSTM. Ceci a conduit à un nouveau type de réseau appelé JANET, qui ne conserve que le port d'oubli et la mémoire de cellule, résultant en un modèle plus général et moins gourmand en puissance de calcul. Cette simplification de l'architecture offre des avantages pratiques et facilite l'application de ces modèles dans différentes tâches. [7]

La Gated Recurrent Unit (GRU) est une simplification de la LSTM, car elle n'utilise pas d'états de cellule explicites. La GRU utilise un seul port de réinitialisation pour contrôler la quantité d'informations modifiées dans le caché d'état, ce qui la rend plus simple et plus efficace dans certaines situations. Cela simplifie l'architecture de la LSTM en évitant l'utilisation d'états de cellule explicites.

II.7.2.5 La machine de Boltzmann profonde (DBN ou Deep Belief Network)

Geoffrey Hinton a élaboré le DBN, une architecture de réseau de neurones composé de plusieurs machines Boltzmann restreintes (RBM). Les RBM sont utilisés pour l'apprentissage non supervisé et les DBN combinent les RBM en empilant les autres. Geoffrey Hinton a contribué significativement à l'avancement de l'apprentissage automatique et a ouvert des perspectives pour l'utilisation des réseaux de neurones dans diverses applications [7]. L'algorithme RBM (Restricted Boltzmann Machine) est un outil polyvalent utilisé dans diverses

tâches telles que la réduction de dimensions, la régression, le filtrage collaboratif, la classification, l'apprentissage de fonctionnalités et la modélisation de thèmes. Grâce à leur architecture et à leur fonctionnement, les RBM sont adaptés à une variété de tâches d'apprentissage automatique. Ils peuvent réduire la dimensionnalité des données, modéliser les relations entre les variables d'entrée et de sortie, prédire les préférences ou recommandations en fonction des comportements passés, et jouent un rôle crucial dans la classification.

II.7.3 Réseaux de Neurones à Convolution (CNN)

Les réseaux de neurones convolutifs sont conçus pour le traitement des signaux et des images, et ont été largement utilisés dans le domaine de la vision par ordinateur. Les couches de convolution permettent de capturer des motifs locaux dans les données d'entrée, ce qui permet d'extraire des caractéristiques importantes et de les mettre en évidence. Les ConvNets ont été largement utilisés dans des domaines tels que le traitement du signal et la classification d'images, et sont très performants dans la détection d'objets, la reconnaissance faciale, la segmentation d'images, la classification d'images, etc. Open CV fournit des outils avancés pour le traitement d'images et la vision par ordinateur. Les réseaux de neurones convolutifs (CNN) sont particulièrement efficaces pour le traitement d'images, car ils sont capables de détecter des caractéristiques à n'importe quel endroit. L'apprentissage des CNN implique l'optimisation des paramètres, y compris les noyaux de convolution, pour minimiser la différence entre les sorties du réseau et les étiquettes "ground truth" correspondantes. L'utilisation des CNN est répandue dans divers domaines, y compris la vision par ordinateur, la reconnaissance d'objets, la médecine, l'industrie automobile et les applications mobiles.

II.7.3.1 Les Couches CNN

CNN est une construction mathématique qui se compose généralement de trois types de couches distinctes (ou blocs de construction) : la convolution, le pooling et les couches fully connected.

- **Fully connected** La dernière couche d'un réseau profond, la couche classique de perception, opère la discrimination finale entre des images à reconnaître, par exemple. Les couches précédentes construisent et extraient leurs caractéristiques.
- **Convolution** utilise une réduction de dimension pour créer une convolution sur le signal d'entrée.

- **Pooling (POOL)** réduction de dimension en substituant une partie des entrées (sous-image) par une valeur, généralement le maximum.

II.7.3.2 L'algorithme de rétro propagation

L'algorithme d'apprentissage de rétro-propagation commence par circuler vers l'avant les données d'entrée jusqu'à obtenir une entrée calculée par le réseau. Ensuite, la sortie calculée est comparée à la sortie réelle connue. Les poids sont modifiés de sorte que l'erreur commise entre la sortie calculée soit minimisée à la prochaine itération. En prenant en compte la présence de couches cachées, l'erreur est rétro-propagée vers l'arrière jusqu'à la couche d'entrée tout en modifiant la pondération. Tous les exemples sont répétés jusqu'à ce qu'une erreur de sortie soit considérée comme négligeable. [8]

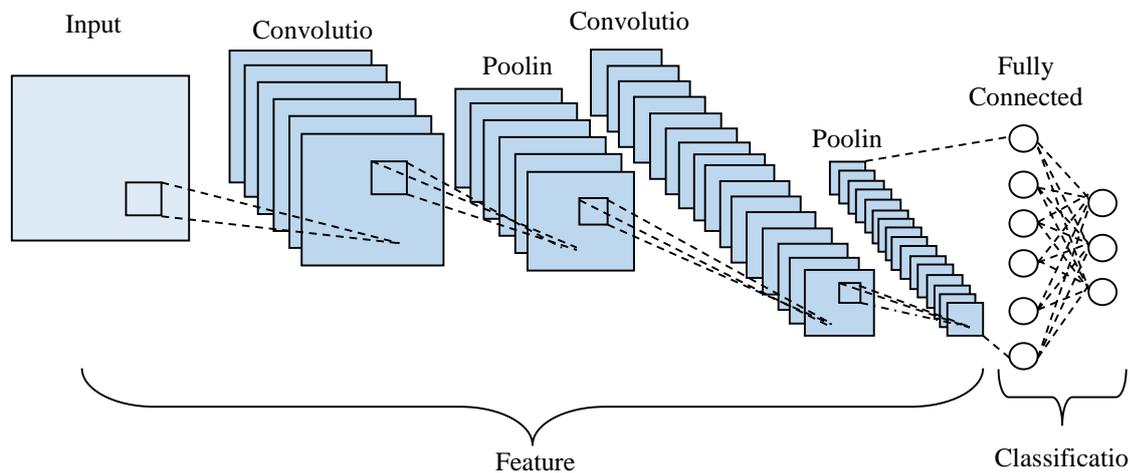


Figure.2.5 : Le schéma de CNN.

II.8 Conclusion

Un domaine de recherche très actif, la machine Learning continue d'avancer pour améliorer les résultats.

L'apprentissage profond est un outil puissant qui peut être utilisé pour réaliser de nombreuses actions et révolutionner des domaines technologiques. Il alimente la traduction automatique

moderne, les moteurs de recherche, les assistants informatiques et de nombreuses applications de notre vie quotidienne.

Ce chapitre examine différents types d'apprentissage automatique et d'apprentissage en profondeur, identifie la fonction d'activation de base et examine les architectures des réseaux neuronaux.

De plus, les types de réseaux de neurones les plus utilisés ont été présentés : ANN, RNN et CNN.

Chapitre III

Modèles de réseaux de neurones récurrents BILSTM et GRU

III.1 Introduction

Les types de réseaux de neurones récurrents (RNN) qui ont été développés pour modéliser et traiter des données séquentielles sont LSTM (Long Short-Term Memory), BILSTM (Bidirectional LSTM) et GRU. Ils étaient tous destinés à résoudre le problème de la dépendance et de l'oubli d'informations dans les RNN traditionnels.

Les portes spéciales des LSTM leur permettent d'accéder et de mémoriser des informations pendant de longues périodes. Lors de la modélisation séquentielle, les BILSTM peuvent prendre en compte le contexte passé et futur grâce à leurs connexions unidirectionnelles et bidirectionnelles. En revanche, les GRU utilisent une méthode plus simple avec moins de portes, ce qui les rend plus rapides dans l'entraînement et l'exécution.

Dans ce chapitre, nous présentons différentes métriques : la moyenne, la valeur maximale (max), le coût minimum... etc.

Ensuite nous nous intéresserons aux Les modèles BILSTM et GRU et les techniques d'optimisation ADAM et SGD sont utilisés pour optimiser les modèles de réseaux de neurones, y compris les architectures LSTM et GRU.

Enfin nous passerons à mettre en évidence l'évaluation des hyper-paramètres telles que le taux d'apprentissage, la taille du mini-batch, le nombre d'époques et le nombre de couches cachées, est essentielle pour obtenir des performances optimales.

III.2 Les Métriques

1. La formule moyenne générale est exprimée mathématiquement en :

$$\text{Moyenne} = \frac{\text{Sum de Observations}}{\text{Nombre total d'Observation}} \quad (3.1)$$

2. Déterminer la valeur maximale de l'équation quadratique

Par exemple, on peut affirmer que si l'équation a été représentée sous la forme de

$$ax^2 + bx + c \quad (3.2)$$

La formule pour trouver la valeur maximale sera

$$\text{MAX} = c - \left(\frac{b^2}{4a}\right) \quad (3.3)$$

3. Le coût minimum en utilisant l'équation

$$\text{MIN} = c - \left(\frac{b^2}{4a}\right) \quad (3.4)$$

4. RMS (Root Mean Square) :

$$x_{RMS} = \sqrt{\frac{x_1^2 + x_2^2 + \dots + x_n^2}{n}} \quad (3.5)$$

Où \mathbf{x} = valeurs de données.

Et \mathbf{n} = nombre total d'éléments.

5. Standard déviation :

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n}} \quad (3.6)$$

6. Lavariance est calculée comme :

$$S^2 = \sqrt{\frac{\sum (x_i - \bar{x})^2}{n-1}} \quad (3.7)$$

7. Kurtosis formula :

$$K = \frac{\sum (x - \bar{x})^4}{ns^4} \quad (3.8)$$

8. Valeur Peak-to-Peak :

$$P = \frac{MAX - MIN}{2} \quad (3.9)$$

9. Équation du facteur de pointe :

$$P1 = \left(\frac{MAX}{RMS}\right)^2 \quad (3.10)$$

10. Équation du facteur de pointe :

$$M = \text{Mean (abs (x))} \quad (3.11)$$

11. Facteur Impulse :

$$F_i = \frac{P}{M} \quad (3.12)$$

12. Clearance Factor:

$$F_C = \left(\frac{P}{M}\right)^2 \quad (3.13)$$

III.3 Long Short-Term Memory (LSTM)

Hochreiter et Schmidhuber (1997) ont introduit les cellules de mémoire à long terme courtes (LSTM) pour apprendre les dépendances à long terme [1].

Les architectures récurrentes les plus courantes dans les réseaux de neurones artificiels sont LSTM (Mémoire longue durée). Comme d'autres réseaux récurrents, il est constitué de couches de neurones connectés.

Les LSTM peuvent résoudre le problème du gradient et gérer les dépendances à long terme, ce qui les distingue des autres architectures récurrentes. Lors de l'apprentissage des réseaux de neurones profonds, les LSTM évitent le problème de la disparition ou de l'explosion du gradient.

Les informations sont stockées dans des cellules ou des blocs de mémoire dans les LSTM. L'état interne du réseau est maintenu et mis à jour par ces unités de calcul LSTM. Cela permet aux LSTM de conserver des informations sur de longues séquences temporelles et de faire des prédictions en se basant sur des informations antérieures.

L'utilisation de portes, telles que les portes d'oubli, d'entrée et de sortie, qui régulent le flux d'informations à travers la cellule, est une caractéristique importante des LSTM. Ces portes permettent aux LSTM de contrôler l'information qui est transmise, oubliée ou mémorisée à la sortie [2].

Les blocs mémoire sont un ensemble de blocs connectés de manière récurrente qui composent une couche LSTM. Ces blocs peuvent être considérés comme des puces de mémoire différentes dans un ordinateur numérique. Chaque année une ou plusieurs cellules mémoire. Connectées de manière récurrente et équipées de trois unités multiplicatives — les portes d'entrée, de sortie et d'oubli — qui offrent des analogues continus aux opérations d'écriture, de lecture et de réinitialisation pour les cellules [3].

III.4 L'architecture d'un modèle LSTM

III.4.1 Poids de LSTM (Weights)

Une cellule de mémoire dans une architecture LSTM possède des paramètres de poids pour l'entrée, la sortie et un état interne. Ces paramètres permettent de pondérer les différentes informations et de construire l'état interne de la cellule en fonction des pas du temps d'entrée.

- **Poids d'entrée** sont utilisés pour pondérer l'entrée au pas de temps actuel. Ils déterminent l'importance relative de chaque élément de l'entrée pour la cellule de mémoire.
- **Poids de sortie** sont utilisés pour pondérer la sortie du dernier pas de temps. Ils contrôlent l'influence de la sortie précédente sur la sortie actuelle de la cellule de mémoire.
- **L'état interne** est un état intermédiaire qui est utilisé dans le calcul de la sortie pour le pas de temps actuel. Il est construit en fonction de l'entrée courante, de l'état interne précédent et des poids d'entrée. Cet état interne permet à la cellule de mémoire de mémoriser des informations sur de longues séquences temporelles et de maintenir une mémoire à long terme [3].

III.4.2 Portes de LSTM (Gated)

Une cellule LSTM est composée de trois portes principales : la porte d'oubli (Forget Gate), la porte d'entrée (Input Gate) et la porte de sortie (Output Gate). Chaque porte est responsable de réguler le flux d'informations à travers la cellule.

- **La porte d'oubli** détermine quelles informations sont importantes à oublier de la cellule. Elle prend en compte l'état caché précédent et l'entrée actuelle pour décider quelles informations sont pertinentes et méritent d'être conservées.
- **La porte d'entrée** contrôle les informations nouvelles à ajouter à l'état caché. Elle examine l'entrée actuelle et l'état caché précédent pour déterminer les nouvelles informations à inclure.
- **La porte de sortie** décide quelle partie de l'état caché doit être exposée à la sortie du LSTM. Cette sortie peut être utilisée pour effectuer une prédiction ou comme entrée pour les étapes suivantes du modèle.

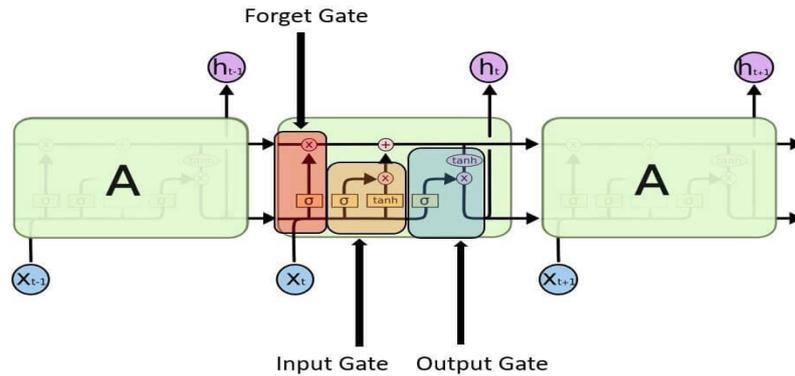
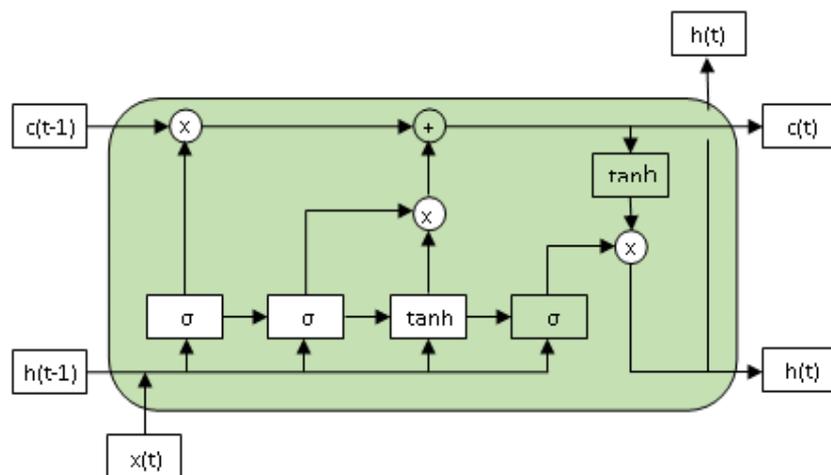


Figure 3.1 : les portes LSTM.

La structure principale des réseaux LSTM consiste en une couche d'entrée, une ou plusieurs couches cachées et une couche de sortie. Laissez $\{x_1, x_2, \dots, x_t\}$ la séquence d'entrée d'un LSTM, où $x_t \in R^d$ représente un vecteur d'entrée de la dimension de la fonctionnalité à l'étape de temps t -th. Le nombre de neurones dans la couche d'entrée est égal au nombre dans l'espace de fonction. Le composant le plus important de la structure LSTM est la cellule de mémoire (Figure.3.2). L'état de la cellule de mémoire C_{t-1} interagit avec la sortie intermédiaire h_{t-1} et l'entrée x_t afin de mettre à jour le vecteur d'états interne en fonction des sorties, de l'étape de temps précédente et des entrées de la phase de temps actuelle. En outre, la cellule de mémoire définit également un nœud d'entrée o_t , un port d'entrée i_t et un f_t de sortie pour maintenir et ajuster l'état de cette cellule. Le processus de calcul est décrit dans les équations (3.14) à (3.17).

Figure 3.2: LSTM architecture.



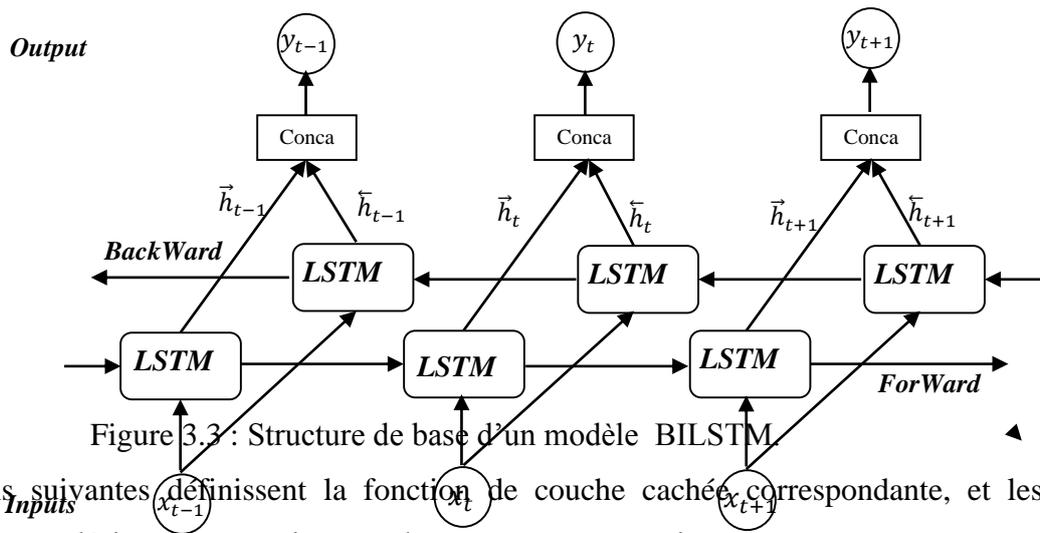
$$f_t = \sigma(W_f x_t + V_f h_{t-1} + b_f)$$

$$\begin{aligned}
i_t &= \sigma(W_i x_t + V_i h_{t-1} + b_i) \\
\hat{C}_t &= \tanh(W_C x_t + V_C h_{t-1} + b_C) \\
C_t &= f_t * C_{t-1} + i_t * \hat{C}_t \\
o_t &= \sigma(W_o x_t + V_o h_{t-1} + b_o) \\
h_t &= o_t * \tanh(C_t)
\end{aligned}
\tag{3.14}$$

$x_t \in R^d$ Et $h_t \in R^k$ sont les entrées et les sorties de l'unité LSTM. $i_t \in R^k, f_t \in R^k, o_t \in R^k$ les vecteurs d'activation de la porte d'entrée, oublier la porte et la porte de sortie, respectivement. Tous les paramètres $W \in R^{k \times d}$ $V \in R^{k \times d}$ $b \in R^k$, ont des paramètres apprenables et * est la multiplication par élément par, σ représente la fonction d'activation sigmoïde.

III.5 Modèle Bidirectionnel LSTM

Le réseau LSTM de base ne considère que le contexte antérieur des données sans contexte futur. Sachant que les données des capteurs de la série temporelle ont de fortes dépendances temporelles, le traitement de ces données est insuffisant. Afin de concevoir les informations temporelles des contextes futurs et passés, une structure LSTM bidirectionnelle est construite. Cette structure traite les données de séquence à la fois dans les chemins avant et arrière avec deux couches cachées, puis les informations de temps en avant et en arrière seront introduites dans la même couche de sortie. La structure du réseau BILSTM est illustrée dans la Figure 3.3.



Les équations suivantes définissent la fonction de couche cachée correspondante, et les symboles \rightarrow et \leftarrow désignent respectivement le processus avant et inverse.

$$\vec{f}_t = \sigma(\vec{W}_f \vec{x}_t + \vec{V}_f \vec{h}_{t-1} + \vec{b}_f)$$

$$\begin{aligned}
\vec{i}_t &= \sigma(\vec{W}_i \vec{x}_t + \vec{V}_i \vec{h}_{t-1} + \vec{b}_i) \\
\vec{c}_t &= \tanh(\vec{W}_c \vec{x}_t + \vec{V}_c \vec{h}_{t-1} + \vec{b}_c) \\
\vec{c}_t &= \vec{f}_t * \vec{c}_{t-1} + \vec{i}_t * \vec{c}_t \\
\vec{o}_t &= \sigma(\vec{W}_o \vec{x}_t + \vec{V}_o \vec{h}_{t-1} + \vec{b}_o) \\
\vec{h}_t &= \vec{o}_t * \tanh(\vec{c}_t)
\end{aligned} \tag{3.15}$$

$$\begin{aligned}
\vec{f}_t &= \sigma(\vec{W}_f \vec{x}_t + \vec{V}_f \vec{h}_{t-1} + \vec{b}_f) \\
\vec{i}_t &= \sigma(\vec{W}_i \vec{x}_t + \vec{V}_i \vec{h}_{t-1} + \vec{b}_i) \\
\vec{c}_t &= \tanh(\vec{W}_c \vec{x}_t + \vec{V}_c \vec{h}_{t-1} + \vec{b}_c) \\
\vec{c}_t &= \vec{f}_t * \vec{c}_{t-1} + \vec{i}_t * \vec{c}_t \\
\vec{o}_t &= \sigma(\vec{W}_o \vec{x}_t + \vec{V}_o \vec{h}_{t-1} + \vec{b}_o) \\
\vec{h}_t &= \vec{o}_t * \tanh(\vec{c}_t)
\end{aligned} \tag{3.16}$$

h_t Est le vecteur concaténé des sorties du processus avant et arrière. Il représente les éléments cachés de la structure BILSTM et est écrit comme suit:

$$h_t = \vec{h}_t \oplus \vec{h}_t \tag{3.17}$$

Le principe de fonctionnement d'un modèle BILSTM (Bidirectional Long Short-Term Memory) repose sur l'utilisation de deux ensembles de cellules LSTM : une pour traiter la séquence dans l'ordre direct (forward LSTM) et une autre pour traiter la séquence dans l'ordre inverse (backward LSTM). Cette approche permet au modèle de capturer à la fois les informations passées et futures d'une séquence, offrant ainsi une meilleure compréhension du contexte global [4].

L'architecture complète proposée pour la classification du modèle BILSTM comporte 6 couches au total, à savoir :

1. La couche d'entrée de séquence : Taille d'entrée égale 12.
2. Couche Bi-LSTM : Les couches BILSTM sont composées de deux ensembles de cellules LSTM : une pour le traitement de la séquence dans l'ordre direct (forward

LSTM) et une autre pour le traitement de la séquence dans l'ordre inverse (backward LSTM). Chaque ensemble de cellules LSTM conserve son propre état interne et produit sa propre sortie. BILSTM Layer (NumiddenUnits=150).

3. La couche de dropout : Des couches de dropout peuvent être ajoutées entre les couches BILSTM pour réduire le surajustement en désactivant aléatoirement certains neurones à chaque étape d'entraînement (Probability=0.75).
4. La couche entièrement connectée (fully connected): Une couche entièrement connectée qui suit souvent les couches LSTM et est utilisée pour générer une prédiction est appelée Dense (numClasses=15).
5. La couche Soft-Max.
6. La couche de classification.

Catégorie de classification: L'architecture de classification est décrite dans la figure 3.4.

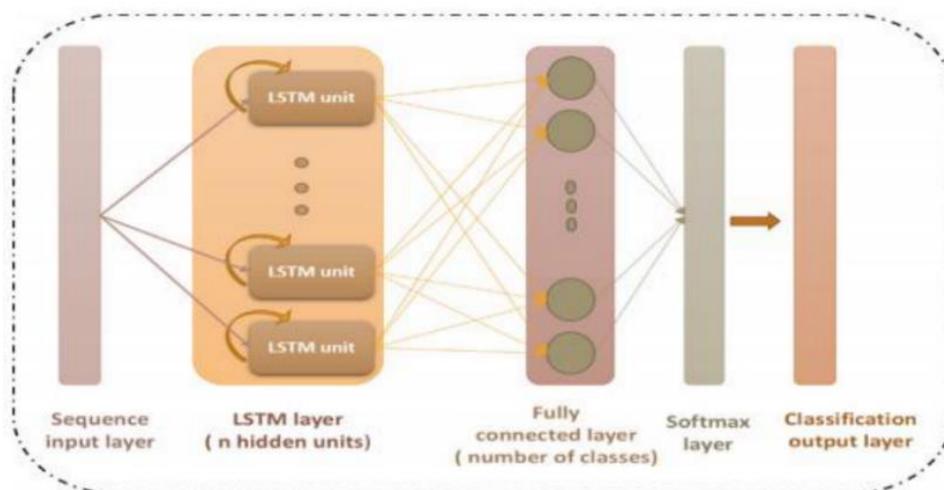


Figure 3.4: Architecture Bi-LSTM pour la classification.

III.6 Modèle GRU (Gated Recurrent Unit)

III.6.1 Structure d'un modèle GRU

Le réseau GRU appartient à l'algorithme RNN. La couche d'entrée est constituée de sveral neurones qui est déterminée par la longueur de l'espace des caractéristiques. La couche de sortie est également déterminée par la taille de l'espace de sortie. La couche cachée est la partie clé de l'algorithme GRURNN qui comprend les cellules de mémoire avec les fonctions principales de GRU. La structure d'une cellule de mémoire GRU est représentée sur la (Figure 3.4). La sortie d'une cellule de mémoire dépend de la porte de réinitialisation t_r et de la porte de mise à jour t_z .

La porte de réinitialisation r_t et la porte de mise à jour z_t sont liées à t_x et $t_1 h$. $t - x$ est la séquence d'entrée respective et $t_1 h$ est la sortie de la cellule de mémoire à l'instant précédent. Ces deux portes différentes jouent des rôles différents dans le réseau. La porte de réinitialisation contrôle le niveau d'ignorance de l'information en $t_1 h$. L'information est davantage ignorée lorsque la valeur de la porte de réinitialisation est plus petite. La porte de mise à jour est conçue pour contrôler l'impact des informations précédentes sur l'état actuel. Plus la valeur de mise à jour est grande, plus les informations précédentes sont utilisées pour déterminer l'état actuel [4].

La propagation vers l'avant peut être représentée sur la base des équations suivantes :

$$\text{Reset gate: } r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \quad (3.5)$$

$$\text{Update gate: } z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \quad (3.6)$$

$$\text{Output: } \tilde{h}_t = \tanh(W_{\tilde{h}} \cdot [r_t * h_{t-1}, x_t]) \quad (3.7)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

$$y_t = \sigma(W_o \cdot h_t)$$

La structure du modèle GRU dispose de deux portes:

- **Porte de réinitialisation (Reset Gate) :** La porte de réinitialisation contrôle l'information provenant de l'état précédent et l'information actuelle de l'entrée. Elle décide quelle partie de l'information précédente doit être oubliée et quelle partie doit être conservée. En d'autres termes, elle permet au modèle de décider si les informations historiques doivent être prises en compte pour la prédiction actuelle.
- **Porte de mise à jour (Update Gate) :** La porte de mise à jour contrôle la quantité d'informations qui doit être transmise à l'état actuel du modèle. Elle décide également si l'information de l'état précédent doit être transmise directement à l'état actuel ou si elle doit être mise à jour avec de nouvelles informations provenant de l'entrée actuelle.

En utilisant ces portes, le modèle GRU peut réguler et contrôler le flux d'informations à travers le réseau, ce qui permet de mieux gérer les problèmes de gradients explosifs ou de disparition des gradients rencontrés dans les RNN classiques.

Il convient de noter que le modèle GRU n'utilise pas de porte de sortie distincte comme c'est le cas dans le modèle LSTM (Long Short-Term Memory). Au lieu de cela, il combine les

informations filtrées par les portes de réinitialisation et de mise à jour pour générer une sortie directement à partir de l'état actuel.

L'architecture complète proposée pour la classification du modèle GRU comporte 6 couches au total, à savoir :

1. La couche d'entrée de séquence : Taille d'entrée égale 12.
2. Couche GRU: GRULayer (NumHiddenUnits=150).
3. La couche de dropout : (Probability=0.75).
4. La couche entièrement connectée : (numClasses=15).
5. La couche Soft-Max.
6. La couche de classification.

III.7 Techniques d'optimisation

L'optimisation joue un rôle crucial dans de nombreux domaines, y compris l'apprentissage automatique et la recherche opérationnelle. Parmi les techniques d'optimisation couramment utilisées, on trouve l'algorithme ADAM (Adaptive Moment Estimation) et le SGD avec moment (Stochastic Gradient Descent with Momentum). Ces deux méthodes sont particulièrement efficaces pour résoudre des problèmes d'optimisation non linéaires.

III.7.1 L'option ADAM

Adam (Adaptive Moment Estimation) est un algorithme d'optimisation couramment utilisé dans l'apprentissage automatique pour mettre à jour le poids d'un modèle de réseau neuronal. Il combine les avantages de deux autres méthodes d'optimisation, à savoir la descente de gradient stochastique (SGD) et le moment (momentum), pour accélérer la convergence et améliorer les performances du modèle. L'algorithme Adam utilise une estimation adaptative des moments du premier ordre (la moyenne mobile des gradients) et du second ordre (la moyenne mobile des carrés des gradients) pour ajuster le poids du modèle de manière efficace.

Fonctionnement de l'algorithme Adam :

1. **Estimation du moment du premier ordre (m)** : Adam calcule une estimation du moment du premier ordre en utilisant une moyenne mobile exponentielle des gradients précédents. Cela permet de garder une trace de l'historique des gradients et de prendre en compte les informations du passé pour ajuster le poids du modèle.
2. **Estimation du moment du second ordre (v)** : Adam calcule également une estimation du moment du second ordre en utilisant une moyenne mobile exponentielle des carrés

des gradients précédents. Cela fournit une mesure de la variabilité des gradients et permet d'adapter l'importance de la mise à jour des poids en fonction de cette variabilité.

3. **Correction du biais** : Au début de l'entraînement, les estimations du moment du premier et du second ordre sont biaisées vers zéro car elles sont initialisées à zéro. Pour corriger ce biais, Adam effectue une étape de correction en divisant les moments par un terme de correction basé sur le nombre d'itérations.
4. **Mise à jour des poids** : Une fois que les moments du premier et du second ordre sont efficaces et corrigés, Adam utilise ces estimations pour mettre à jour les poids du modèle. La mise à jour est effectuée en combinant l'information du moment du premier ordre (gradients) et du moment du second ordre (variabilité des gradients). Cela permet d'adapter la taille de la mise à jour en fonction de la direction et de la variabilité des gradients.

Il est robuste aux problèmes de choix de taux d'apprentissage initial et est adapté à l'apprentissage de modèles profonds et complexes. Cependant, le réglage approprié des hyperparamètres est crucial pour obtenir de bonnes performances avec Adam.

III.7.2 L'option SGD

La SGD (Stochastic Gradient Descent), ou descente de gradient stochastique en français, est un algorithme d'optimisation largement utilisé dans l'apprentissage automatique pour entraîner des modèles de réseau neuronal. C'est une méthode itérative qui vise à ajuster le poids du modèle en minimisant progressivement la fonction de perte.

L'algorithme SGD fonctionne en calculant le gradient de la fonction de perte par rapport au poids du modèle à chaque itération. Il met ensuite à jour les poids en prenant un petit pas dans la direction opposée du gradient. Contrairement à la descente de gradient classique, qui utilise l'ensemble complet des données d'entraînement pour calculer le gradient, la SGD calcule le gradient en utilisant un seul exemple (ou un petit sous-ensemble d'exemples) à chaque itération. Cela rend la SGD plus rapide en termes de temps de calcul, car elle évite de recalculer le gradient pour l'ensemble des données à chaque itération.

Cependant, il peut introduire du bruit dans la mise à jour des poids, ce qui peut rendre la convergence plus lente et moins stable. Le choix du taux d'apprentissage est également crucial pour les performances du modèle.

III.8 Évaluation des Hyper-paramètres

Les Hyper-paramètres tels que le taux d'apprentissage, la taille du lot (batch size), le nombre d'époques d'entraînement, etc., jouent un rôle crucial dans les performances et le temps d'apprentissage d'un modèle. Un taux d'apprentissage trop élevé peut entraîner une divergence du modèle, tandis qu'un taux d'apprentissage trop faible peut prolonger le temps d'apprentissage. De plus, la taille du lot peut affecter la stabilité de l'apprentissage et le temps d'exécution des itérations.

III.8.1 Le choix du taux d'apprentissage

Le taux d'apprentissage (Learning Rate) est un paramètre crucial dans l'entraînement des modèles de réseau neuronal. Il contrôle la vitesse à laquelle le modèle apprend à partir des données.

Le choix du taux d'apprentissage peut être réalisé de différentes manières, y compris expérimenter avec différentes valeurs en observant la performance du modèle au fur et à mesure de l'entraînement. Des méthodes adaptatives peuvent également être utilisées, et des techniques de recherche systématique comme la recherche par grille peuvent être utilisés. Il n'existe pas de valeur universelle qui fonctionne pour tous les cas, et il est recommandé d'expérimenter avec différentes valeurs pour trouver celle qui offre les meilleures performances pour votre modèle et votre problème spécifique. Pour cela, nous avons choisi trois valeurs différentes : large, moyenne et petite {0.01, 0.001, 0.0001}.

III.8.2 Le choix de la taille du mini-batch :

Le choix de ce hyper-paramètre est important dans les modèles d'apprentissage profond, car il impacte directement la vitesse de convergence de l'optimisation de la fonction objectif.

Un mini-batch est un sous-ensemble d'échantillons de données utilisés pour effectuer une mise à jour des poids dans le modèle d'apprentissage automatique. La taille du mini-batch est déterminée par l'utilisateur et peut varier en fonction des ressources disponibles et de la taille de l'ensemble de données. L'utilisation de mini-batch permet de traiter les données de manière plus efficace et de bénéficier de la parallélisations lors de l'entraînement.

III.8.3 Le choix du nombre epochs

Une epoch représente une passe complète de l'ensemble des données d'entraînement à travers le modèle. Lors d'une epoch, tous les mini-batch sont utilisés pour mettre à jour les poids du modèle. Le nombre d'epochs détermine combien de fois l'ensemble des données sera utilisé

pour l'entraînement. Un nombre d'epochs plus élevé permet généralement au modèle de mieux s'ajuster aux données, mais il faut éviter le surajustement.

III.8.4 Le choix du taux de dropout

La méthode de régularisation «dropout» est une étape importante dans l'élaboration d'un modèle d'apprentissage profond afin d'éviter le sur-apprentissage. Cette méthode consiste à désactiver aléatoirement certains neurones. Cependant, le choix du taux de dropout est à déterminer empiriquement. Pour cela, nous avons expérimenté différents taux de dropout {0.25, 0.5, 0.75} sur nos modèles.

III.8.5 Le choix de nombre d'unités cachées LSTM

Afin de diminuer le taux d'erreur sur l'ensemble d'entraînement, augmenter la taille du réseau pourrait être une solution qui permet d'améliorer la capacité de représentation des modèles. De ce fait, nous avons augmenté le nombre des unités cachées des LSTM pour permettre à nos modèles de bien s'adapter à l'ensemble d'entraînement.

III.9 Conclusion

Dans ce chapitre, on a expliqué les architectures BILSTM et GRU sont des modèles récurrents efficaces pour la modélisation de données séquentielles. Les LSTM, avec leurs portes de contrôle du flux d'information, permettent de capturer les dépendances à long terme, tandis que les GRU offrent une architecture plus simple mais tout aussi performante, les techniques d'optimisation ADAM et SGD sont largement utilisées. ADAM, avec son adaptation automatique du taux d'apprentissage, offre une convergence plus rapide et nécessite moins de réglages manuels. SGD est une méthode classique qui peut être utilisée avec succès, mais nécessite souvent plus de temps et d'ajustements pour obtenir de bons résultats.

Chapitre IV : Résultats et Discussion

IV.1 Introduction

Ce chapitre présente un certain nombre d'expériences qui ont été menées pour valider la performance et l'efficacité des deux méthodes proposées (méthode BILSTM et méthode GRU) pour le diagnostic des défauts de roulements. L'évaluation de chacune des deux méthodes est effectuée selon deux options «ADAM et SGDM ». En outre, notre système de diagnostic a été comparé avec d'autres méthodes afin d'approuver l'efficacité et les performances apportées par nos méthodes proposées pour identifier et localiser avec succès les défauts de roulements avec une très grande précision. L'efficacité de l'algorithme proposé est testée à l'aide des données des roulements de la CWRU, qui sont présentées dans la section suivante.

IV.2 Outils du travail (MATLAB):

Matlab, abréviation de "Matrix Laboratory", est un langage de programmation de haut niveau et un environnement de calcul numérique largement utilisé dans divers domaines tels que l'ingénierie, les mathématiques, la physique et la finance.

Matlab fournit un ensemble complet d'outils pour l'analyse de données, le développement d'algorithmes et la visualisation. Il permet aux utilisateurs d'effectuer des calculs mathématiques complexes, de manipuler des matrices et des tableaux, de créer et d'exécuter des algorithmes, ainsi que de générer des visualisations de haute qualité des données. Avec ses capacités étendues et son utilisation répandue dans le monde universitaire et industriel, Matlab est devenu un outil puissant pour la recherche scientifique et technique, l'analyse de données et le développement d'algorithmes. Sa polyvalence, sa précision numérique et sa vaste bibliothèque de fonctions en font un choix populaire pour les professionnels et les étudiants.

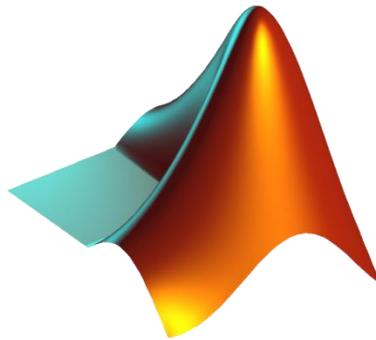


Figure 4.1:matlab logo.

IV.3 Description de la base de données :

Fault Diameter	Motor Load (HP)	Approx. Motor Speed (rpm)	Inner Race	Ball	Outer Race Position Relative to Load Zone (Load Zone Centered at 6:00)		
					Centered @6:00	Orthogonal @3:00	Opposite @12:00
0.007"	0	1797	<u>IR007_0</u>	<u>B007_0</u>	<u>OR007@6_0</u>	<u>OR007@3_0</u>	<u>OR007@12_0</u>
	1	1772	<u>IR007_1</u>	<u>B007_1</u>	<u>OR007@6_1</u>	<u>OR007@3_1</u>	<u>OR007@12_1</u>
	2	1750	<u>IR007_2</u>	<u>B007_2</u>	<u>OR007@6_2</u>	<u>OR007@3_2</u>	<u>OR007@12_2</u>
	3	1730	<u>IR007_3</u>	<u>B007_3</u>	<u>OR007@6_3</u>	<u>OR007@3_3</u>	<u>OR007@12_3</u>
0.014"	0	1797	<u>IR014_0</u>	<u>B014_0</u>	<u>OR014@6_0</u>	*	*
	1	1772	<u>IR014_1</u>	<u>B014_1</u>	<u>OR014@6_1</u>	*	*
	2	1750	<u>IR014_2</u>	<u>B014_2</u>	<u>OR014@6_2</u>	*	*
	3	1730	<u>IR014_3</u>	<u>B014_3</u>	<u>OR014@6_3</u>	*	*
0.021"	0	1797	<u>IR021_0</u>	<u>B021_0</u>	<u>OR021@6_0</u>	<u>OR021@3_0</u>	<u>OR021@12_0</u>

	1	1772	<u>IR021_1</u>	<u>B021_1</u>	<u>OR021@6_1</u>	<u>OR021@3_1</u>	<u>OR021@12_1</u>
	2	1750	<u>IR021_2</u>	<u>B021_2</u>	<u>OR021@6_2</u>	<u>OR021@3_2</u>	<u>OR021@12_2</u>
	3	1730	<u>IR021_3</u>	<u>B021_3</u>	<u>OR021@6_3</u>	<u>OR021@3_3</u>	<u>OR021@12_3</u>
0.028"	0	1797	<u>IR028_0</u>	<u>B028_0</u>	*	*	*
	1	1772	<u>IR028_1</u>	<u>B028_1</u>	*	*	*
	2	1750	<u>IR028_2</u>	<u>B028_2</u>	*	*	*
	3	1730	<u>IR028_3</u>	<u>B028_3</u>	*	*	*

Tableau 4.2 : Base de données.

Les données du signal de vibration sont collectées sous les charges 0, 1, 2 et 3 hp, correspondant aux vitesses 1797, 1772, 1750 et 1730 tr/min, respectivement, et enregistrées à une fréquence d'échantillonnage de 12 KHz. quinze classes sont sélectionnées pour chaque charge, chaque classe définissant un état de fonctionnement du roulement utilisé dans les expériences, comme décrit dans le tableau IV.1 - données de l'accéléromètre côté entraînement

IV.4 Phase préparation des données :

Afin d'avoir un nombre suffisant d'échantillons pour l'apprentissage et le test des classificateurs, les signaux de vibration sont divisés en segments de longueur égale. La longueur de chaque segment doit être suffisante pour extraire les 12 caractéristiques qui vont être utilisées comme signaux d'entrées. Le processus de division des signaux en segments est illustré ci-dessous :

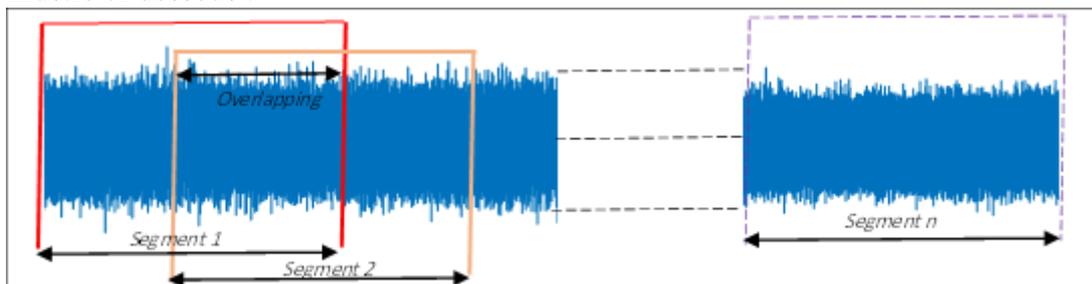


Figure 4.2: Segmentation imbriquée des signaux de vibrations.

Premièrement, les données de vibration sont divisées en n segments se chevauchant de longueur égale avec une taille de pas p . Le choix vise à éviter de perdre l'information entre deux segments successifs. La taille de pas p est égale à la moitié de la taille de segment. Dans notre travail, le nombre de segments pour chaque signal est déduit en fonction de la taille du segment. Dans nos approches, trois types de chevauchements (25%, 50% et 75%) avec des tailles de 1024, 1296 et 1600 points respectivement.

IV.5 Interfaçage proposé

L'interfaçage que nous avons proposé comporte trois parties Figure IV.2 :

- Partie I : préparation de données. Dans cette partie, on choisit la taille de segment, le pourcentage de chevauchement, nombre de classes et type de charge ;
- Partie II : le modèle et l'algorithme d'optimisation soit « Adam » ou « Sgdm » qui peut être utilisé pour mettre à jour les poids du réseau de manière itérative sur la base des données d'apprentissage en fonction des paramètres suivants (Nombre de Couches Cachées, NimiBatchSize, Nombre d'epochs et Learning Rate). Partie III : Apprentissage et validation : Le choix approprié des paramètres d'apprentissage améliore les performances du modèle telles que la précision de la classification et la vitesse de convergence. Une interface, basée sur le langage de programmation Matlab, a été développée pour entraîner le modèle en ajustant les différents paramètres d'apprentissage afin d'obtenir le modèle optimal.

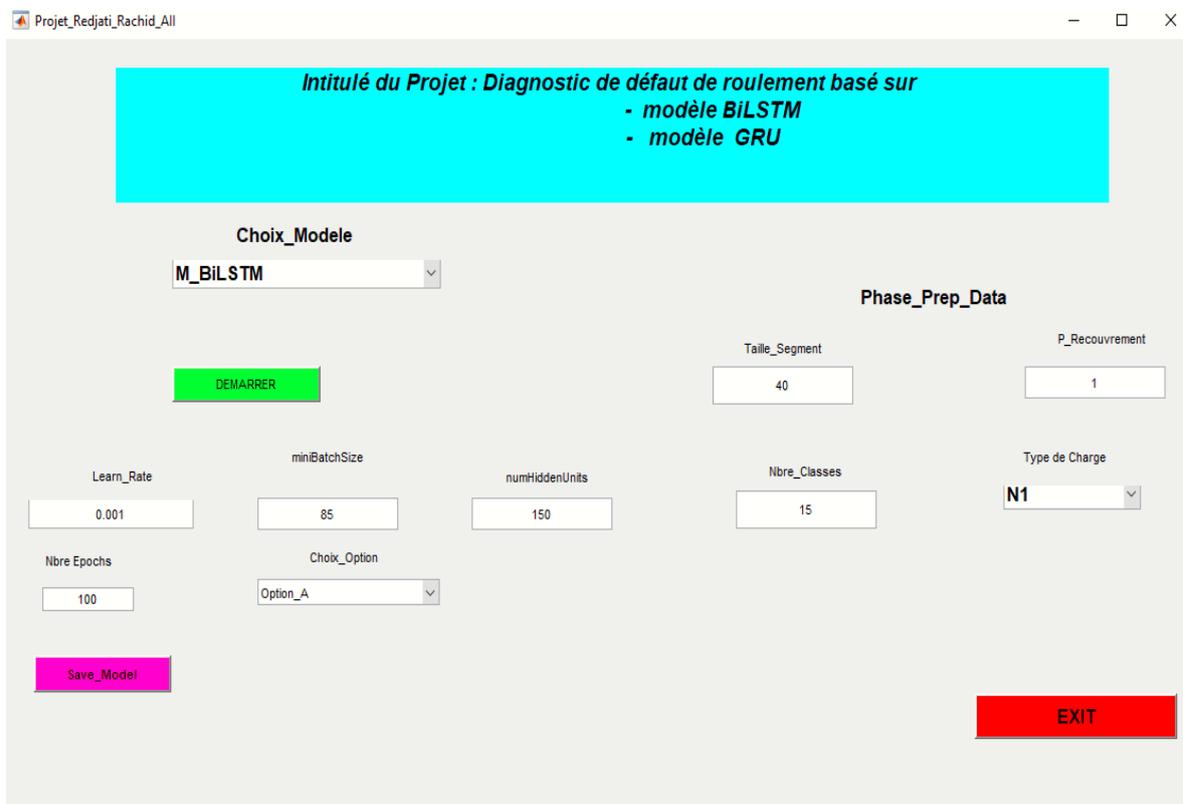


Figure 4.3: Interfaçage proposé.

IV.6. Diagnostic des défauts de roulements par les deux méthodes proposées pour 15 classes

IV.6.1 Les mesures d'évaluation

L'évaluation des performances de n'importe quel système de classification est une étape très importante parce que ces performances reflètent la fiabilité du système proposé. Il s'agit essentiellement d'une version étendue de la matrice de confusion. Il existe d'autres mesures que la matrice de confusion qui peuvent aider à mieux comprendre et analyser notre

modèle et ses performances.

1. **Taux d'apprentissage (Accuracy)** : Calcule le rapport entre le nombre de classes prédites correctes et le nombre total d'échantillons évalués.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

2. **Sensibilité (ou recall)** : Utilisé pour calculer la fraction des modèles positifs qui sont correctement classés.

$$\text{sensitivity} = \frac{TP}{TP + FN} \quad (2)$$

3. **Spécificité** : Utilisée pour calculer la fraction des motifs négatifs qui est correctement Classés.

$$\text{specificity} = \frac{TN}{FP + TN} \quad (3)$$

4. **Précision** : Utilisée pour calculer les motifs positifs correctement prédits par tous les motifs prédits dans une classe positive.

$$\text{precision} = \frac{TP}{TP + FP} \quad (4)$$

5. **F1 Score** : On calcule la moyenne harmonique entre les taux de rappel et de précision.

$$F1_{SCORE} = 2 \times \frac{PRECISION \times RECALL}{PRECISION + RECALL} \quad (5)$$

6. **False Positive Rate (FPR)**: Cette métrique fait référence à la possibilité d'un taux de fausses alarmes.

$$FPR = 1 - \text{Specificity} \quad (6)$$

7. **Le coefficient kappa** : Mesure l'accord inter-annotateurs ajusté en tenant compte de l'accord qui pourrait être obtenu par hasard.

$$kappa = (Po - Pe) / (1 - Pe) \quad (7)$$

8. **Erreur** : Dépend du contexte spécifique de la tâche. En général, l'erreur peut être calculée de différentes manières, mais une approche courante consiste à soustraire la précision de 1 (pour obtenir l'erreur de classification) :

$$\text{Erreur} = 100 - \text{Accuracy} \quad (8)$$

9. **Le coefficient de corrélation de Matthews (MCC)** : Une mesure de la qualité de la classification binaire. Il prend en compte les vrais positifs, les vrais négatifs, les faux

positifs et les faux négatifs pour fournir une mesure de la corrélation entre les prédictions du modèle et les vérités terrain réelles.

$$MCC = \frac{(TP * TN - FP * FN)}{\sqrt{((TP + FP) * (TP + FN) * (TN + FP) * (TN + FN))}} \quad (9)$$

IV.6.2 Modèle BILSTM

Comme mentionné auparavant, les signaux vibratoires pour chaque classe (défaut) sont divisés en segments imbriqués de la même longueur (1024 ,1296 et 1600 points respectivement). Où 80% pour l'apprentissage et 20% pour la validation. Un total de douze caractéristiques statistiques, indicateurs temporels, sont extraits de chaque signal reconstruit à base du traitement Les figure 4.3 et 4.4 montrent les courbes de précision et les courbes de perte d'entraînement et de validation pour l'ensemble des données de test, respectivement. On constate que le "modèle de BILSTM " proposé converge avec un nombre limité d'itérations et que le taux de précision de validation est de 99,5 %, tandis que la perte de validation a été réduite à presque zéro après 800 itérations. Ces performances démontrent l'excellent potentiel de la méthode de diagnostic proposée pour identifier les défauts de roulements des quinze classes dans la quatrième charge (vitesse).

IV.6.3 Validation croisée en 5 Folds

Dans cet ensemble de données, nous avons divisé l'ensemble de données en 5 fois pour l'analyse, c'est-à-dire que 5 algorithmes différents seraient formés à l'aide d'un ensemble de signaux différents de l'ensemble de données. Ce type d'étude de validation nous fournirait une meilleure estimation de nos performances par rapport à la méthode de validation typique.

Fold 1	Test	Training	Training	Training	Training
Fold 2	Training	Test	Training	Training	Training
Fold 3	Training	Training	Test	Training	Training
Fold 4	Training	Training	Training	Test	Training
Fold 5	Training	Training	Training	Training	Test

Tableau 4.2: Validation croisée en 5 folds.

Par exemple pour un des cinq folds, la figure IV.II représente l'évolution des deux paramètres (accuracy et Loss) lors de la phase d'apprentissage. A la fin on obtient un

taux de 99.54% du paramètre d'accuracy pour la validation.

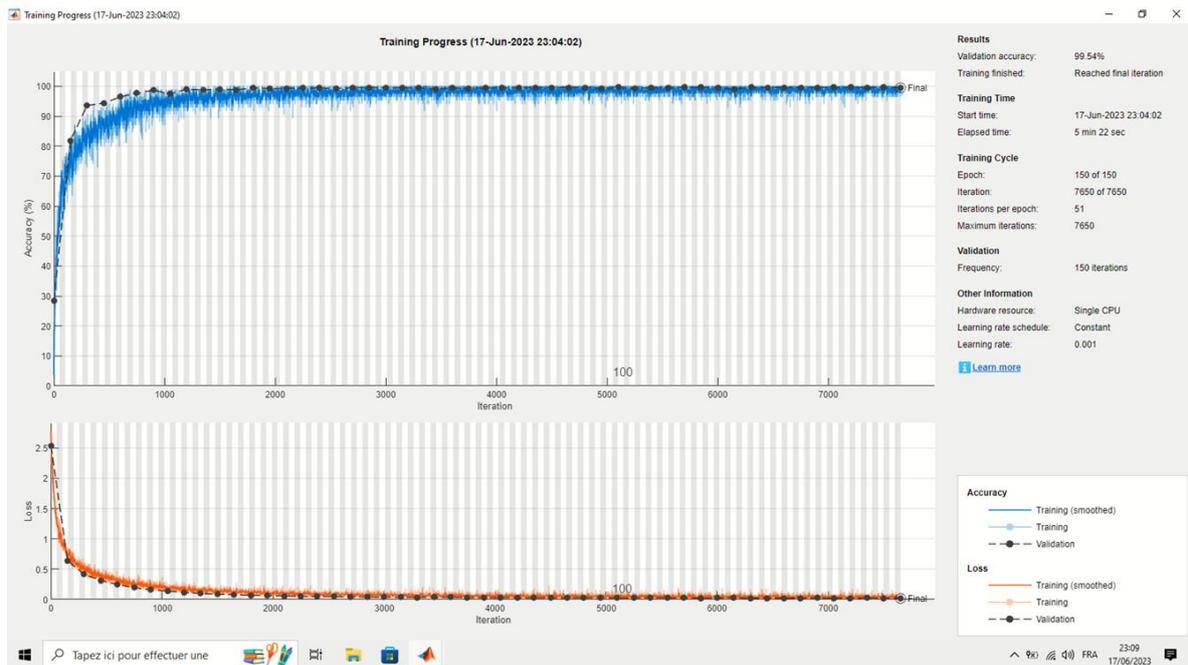


Figure 4.4: Résultat d'apprentissage du modèle BILSTM.

Nous constatons, d'autre part, dans la matrice de confusion de la figure 4.5 de la méthode proposée permet d'identifier l'état de fonctionnement des roulements avec une grande efficacité, et d'autre part, nous constatons que seul (05) échantillons sur 1105 échantillons de test ont été mal classés. On note également dans cette figure qu'à l'exception des classes 3, 11, 12,13 et 15 qui ont donné un taux de précision de 98,40%, toutes les autres classes sont classées avec une précision de 100%.

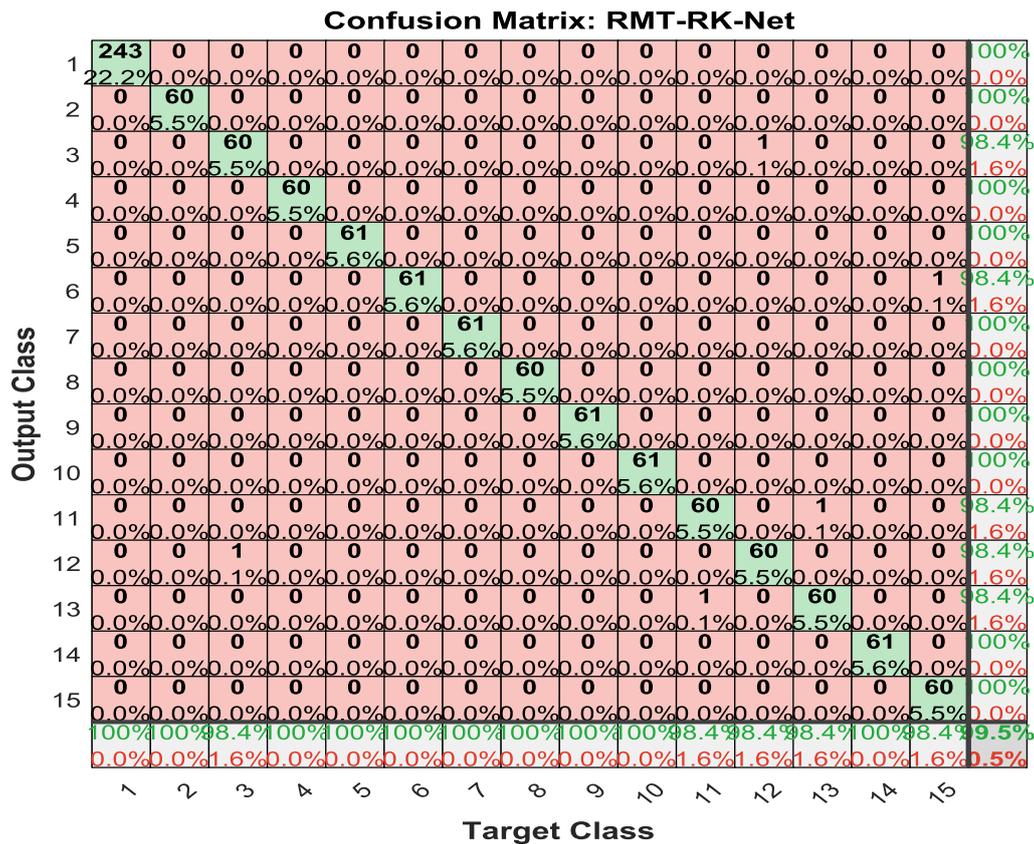


Figure 4.5: Matrice de confusion pour un des folds pour les 15 classes(BILSTM).

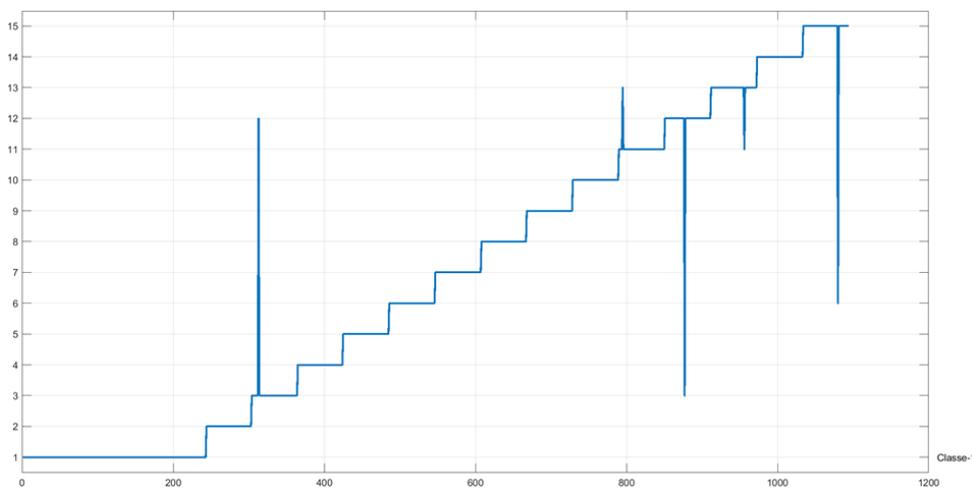


Figure 4.6: Prédiction des différents échantillons des 15 classes pour un fold (BILSTM).

Nous constatons, dans la matrice de confusion globale de la figure 4.7 de la méthode proposée permet d'identifier l'état de fonctionnement des roulements avec une grande efficacité. On note également dans cette figure nous donne un taux de précision de 99,50%.

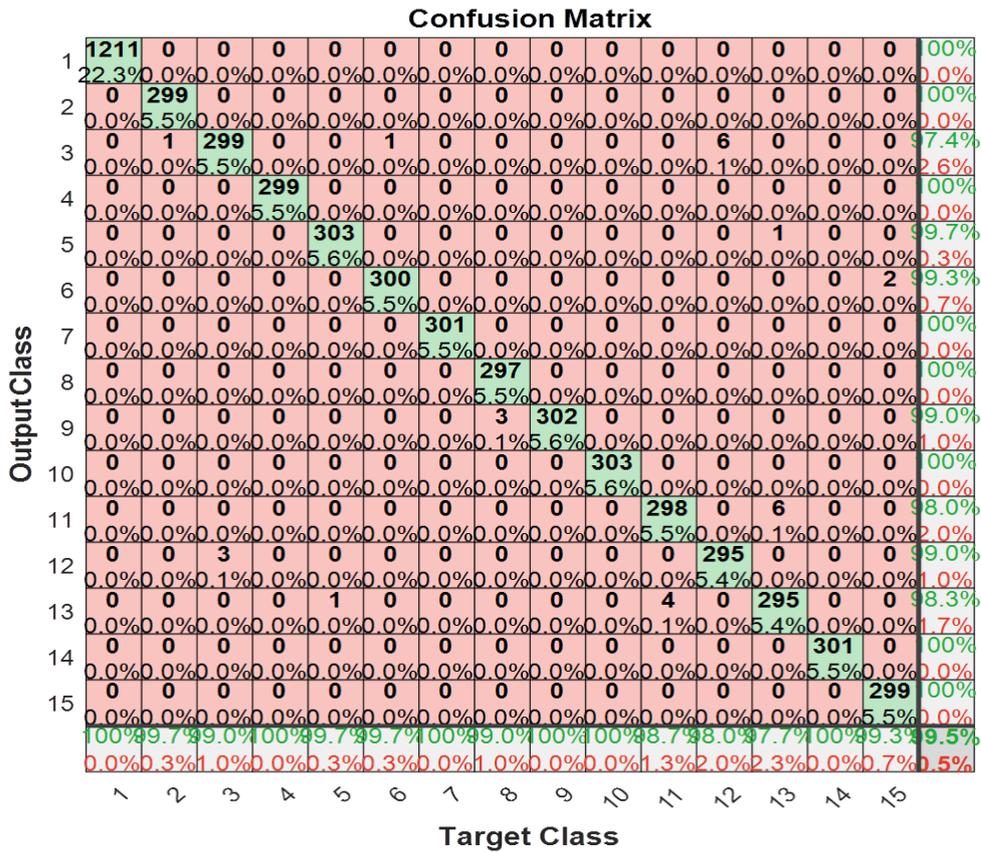


Figure 4.7: Matrice de confusion globale(BILSTM) pour les 15 classes.

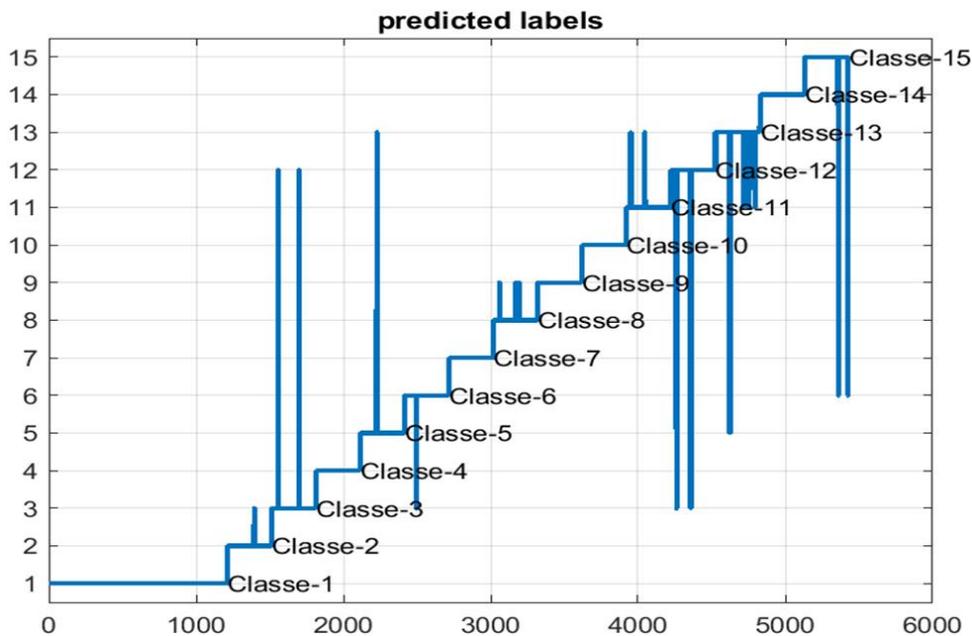


Figure 4.8: Prédiction des différentes 15 classes(BILSTM).

	ACCURACY	ERROR	RECALL	SPECIFICITY	PRECISION	FALSEPOSITIVE RATE	F1_SCORE	MATTHEWSCORRELATION COEFFICIENT	KAPPA
FOLD1	0,9954	0,0046	0,9945	0,9997	0,9946	0,0003	0,9945	0,9942	0,9633
FOLD2	0,9945	0,0055	0,9934	0,9996	0,9935	0,0004	0,9934	0,9931	0,9557
FOLD3	0,9945	0,0055	0,9933	0,9996	0,9934	0,0004	0,9933	0,9930	0,9555
FOLD4	0,9945	0,0055	0,9933	0,9996	0,9934	0,0004	0,9933	0,9930	0,9555
FOLD5	0,9954	0,0046	0,9944	0,9997	0,9945	0,0003	0,9944	0,9941	0,9628

Tableau 4.3: Résultat d'apprentissage du modèle BILSTM pour 15 classes.

IV.6.3 Modèle GRU :

Les figures 4.9 montrent les courbes de précision et les courbes de perte d'entraînement et de validation pour l'ensemble des données de test, respectivement. On constate que le "modèle de GRU " proposé converge avec un nombre limité d'itérations et que le taux de précision de validation est de 99,4 %

Par exemple pour un des cinq folds, la figure 4.9 représente l'évolution des deux paramètres (accuracy et Loss) lors de la phase d'apprentissage. A la fin on obtient un taux de 99.45% du paramètre d'accuracy pour la validation

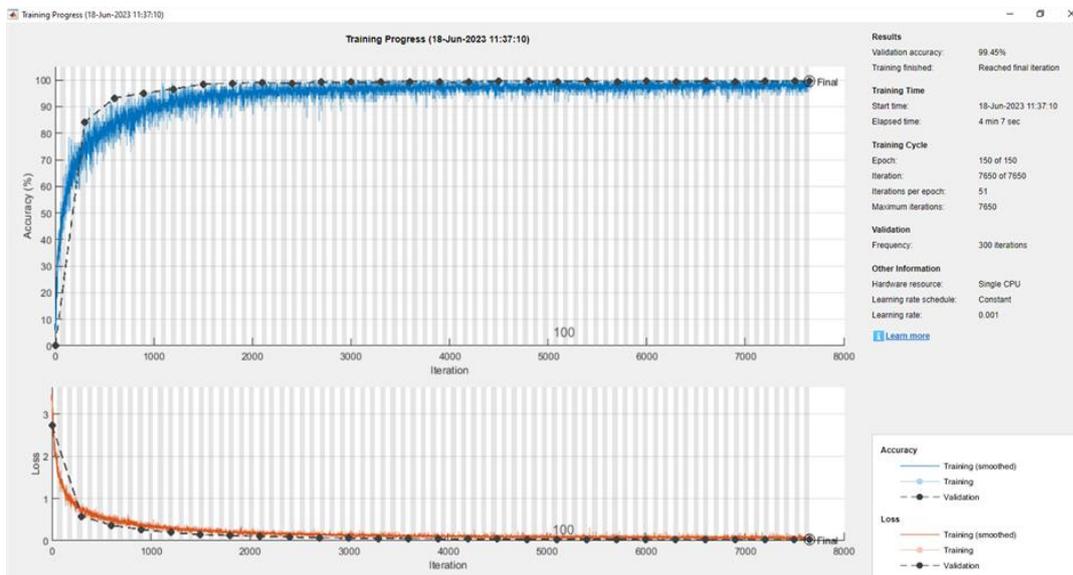


Figure4.9: Résultat d'apprentissage du modèle GRU.

Puis on a la matrice de confusion pour l'un des folds

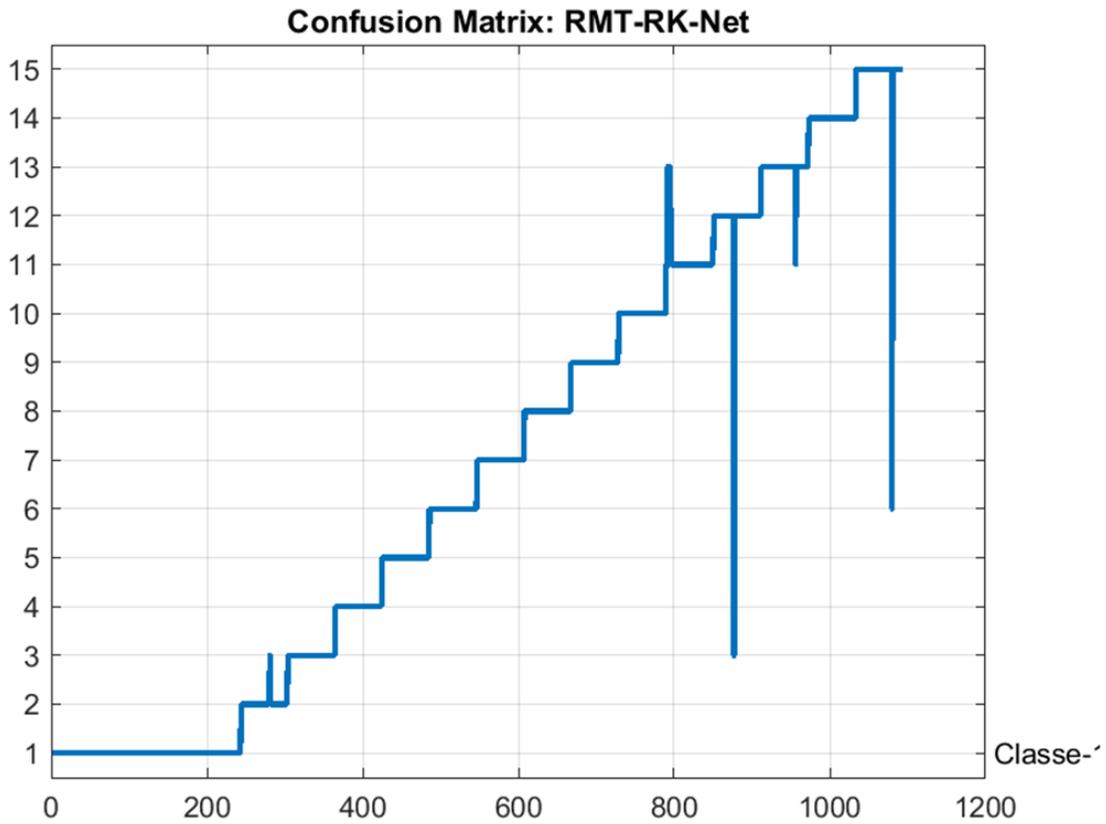


Figure 4.11: Prédiction des différents échantillons des 15 classes pour un fold(GRU).

Nous constatons, dans la matrice de confusion globale de la figure ci-dessous de la méthode proposée permet d'identifier l'état de fonctionnement des roulements avec une grande efficacité. On note également dans cette figure nous donne un taux de précision de 99,4%.

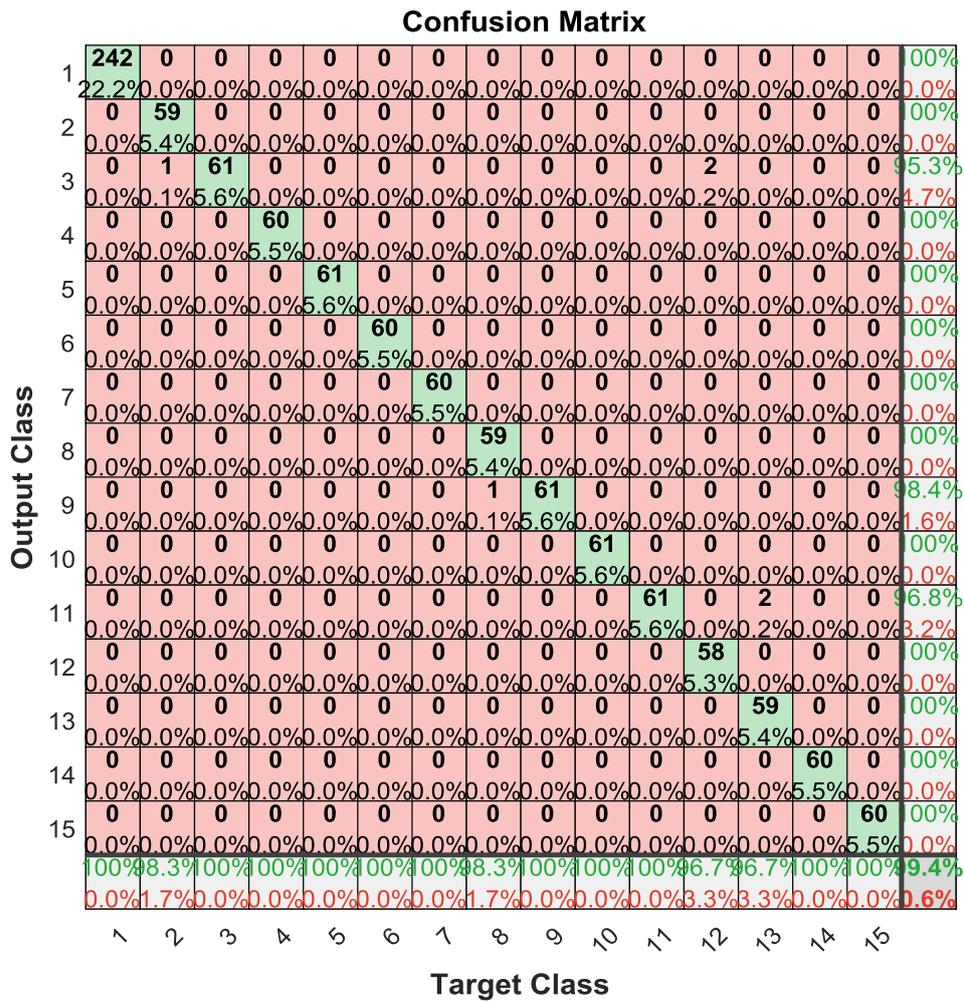


Figure 4.12 : Matrice de confusion globale(GRU) pour les 15 classes.

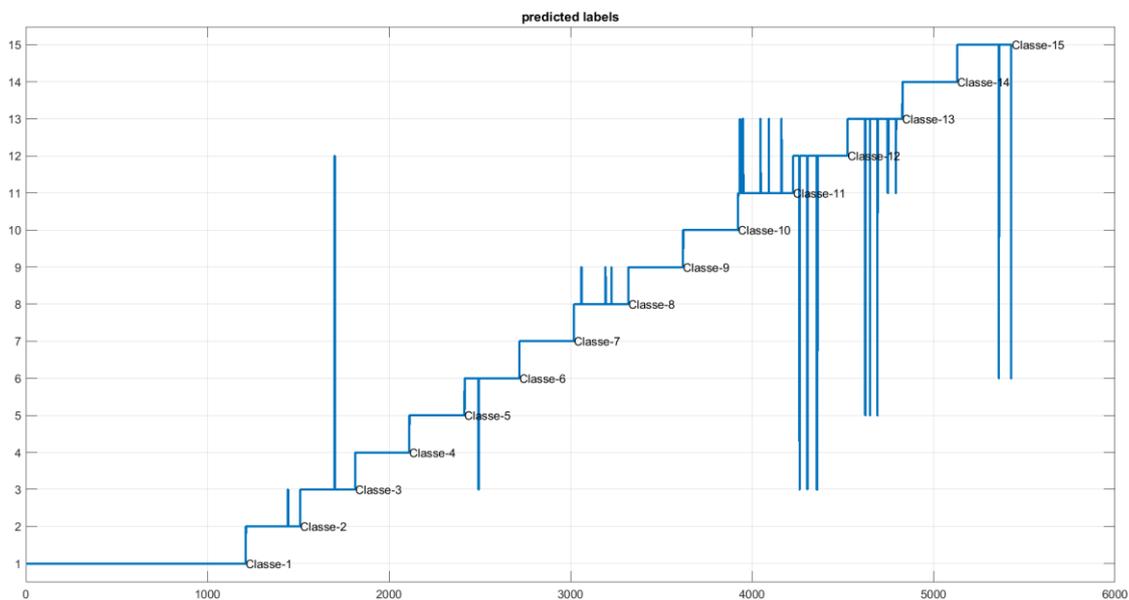


Figure4.13: Prédiction des différentes 15 classes(GRU).

	ACCURACY	ERROR	RECALL	SPECIFICITY	PRECISION	FALSE POSITIVE RATE	F1_SCORE	MATTHEWSCORRELATION COEFFICIENT	KAPPA
FOLD1	0,9945	0,0055	0,9934	0,9996	0,9935	0,0004	0,9934	0,9931	0,9559
FOLD2	0,9945	0,0055	0,9934	0,9996	0,9937	0,0004	0,9934	0,9931	0,9557
FOLD3	0,9917	0,0083	0,9900	0,9994	0,9902	0,0006	0,9900	0,9895	0,9333
FOLD4	0,9917	0,0083	0,9900	0,9994	0,9901	0,0006	0,9900	0,9895	0,9332
FOLD5	0,9926	0,0074	0,9911	0,9995	0,9913	0,0005	0,9911	0,9906	0,9405

Tableau 4.4: Résultat d'apprentissage du modèle BILSTM pour 14 classes.

IV.6.4 Comparaison entre la méthode BILSTM et la méthode de GRU

Les architectures Bidirectional Long Short-Term Memory (BILSTM) et Gated Recurrent Unit (GRU) sont toutes deux utilisées pour le traitement de données séquentielles au sein des réseaux de neurones récurrents (RNN). Bien qu'elles partagent certaines similitudes, elles présentent également des caractéristiques distinctes. Voici une comparaison entre les méthodes BILSTM et GRU :

	ACCURACY	ERROR	RECALL	SPECIFICITY	PRECISION	FALSE POSITIVE RATE	F1_SCORE	MATTHEWSCORRELATION COEFFICIENT	KAPPA
BILSTM	0,99486	0,00514	0,99398	0,99984	0,99388	0,0003	0,99398	0,99348	0,95876
GRU	0,9928	0,007	0,99178	0,9995	0,99176	0,0005	0,99158	0,99136	0,94372

Tableau 4.5: Tableau de comparaison des métriques.

Les résultats expérimentaux montrent que la méthode BILSTM est plus performante que la méthode GRU, où elle a réussi de maintenir un taux de classification élevé pour le diagnostic des défauts de roulements sous la même condition de vitesse. Cela nous a confirmé la supériorité de l'approche DL par rapport aux méthodes ML classiques lesquelles dépendent explicitement de la qualité des descripteurs extraits manuellement. Par conséquent, nous conservons la meilleure méthode pour le reste des expériences.

IV.6.5 Pour les 14 classes

Si on élimine une classe on remarque qu'on peut améliorer les performances du modèle avec un taux de 100% pour certain fold.

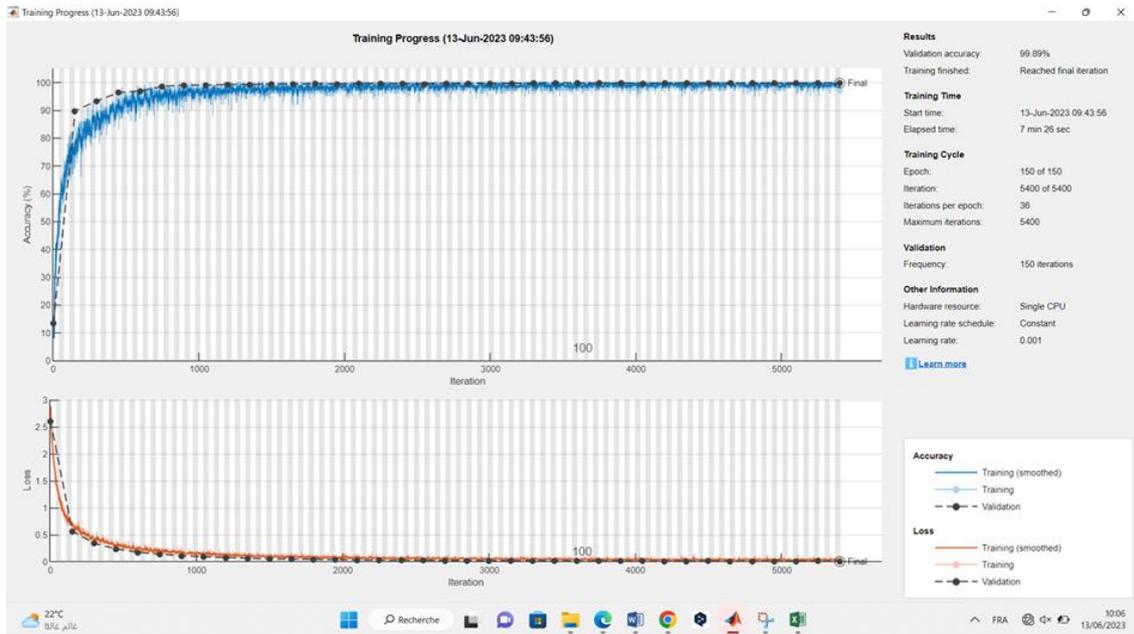


Figure 4.14: Résultat d'apprentissage du modèle BILSTM.(14 classes).

Confusion Matrix

Output Class	1	606 13.4%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%	
	2	0 0.0%	303 6.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%	
	3	0 0.0%	0 0.0%	299 6.6%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%	
	4	0 0.0%	0 0.0%	0 0.0%	298 6.6%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	7 0.2%	0 0.0%	0 0.0%	97.7% 2.3%	
	5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	301 6.7%	0 0.0%	100% 0.0%								
	6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	302 6.7%	0 0.0%	100% 0.0%							
	7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	299 6.6%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%	
	8	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	302 6.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%	
	9	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	301 6.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%	
	10	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	302 6.7%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%	
	11	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	301 6.7%	0 0.0%	0 0.0%	100% 0.0%	
	12	0 0.0%	0 0.0%	0 0.0%	2 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	294 6.5%	0 0.0%	99.3% 0.7%	
	13	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	303 6.7%	1 0.0%	99.7% 0.3%
	14	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	300 6.6%	100% 0.0%
			100% 0.0%	100% 0.0%	100% 0.0%	99.3% 0.7%	100% 0.0%	97.7% 2.3%	100% 0.0%	99.7% 0.3%						
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	

Figure 4.15: La matrice de confusion globale pour les 14 classes.

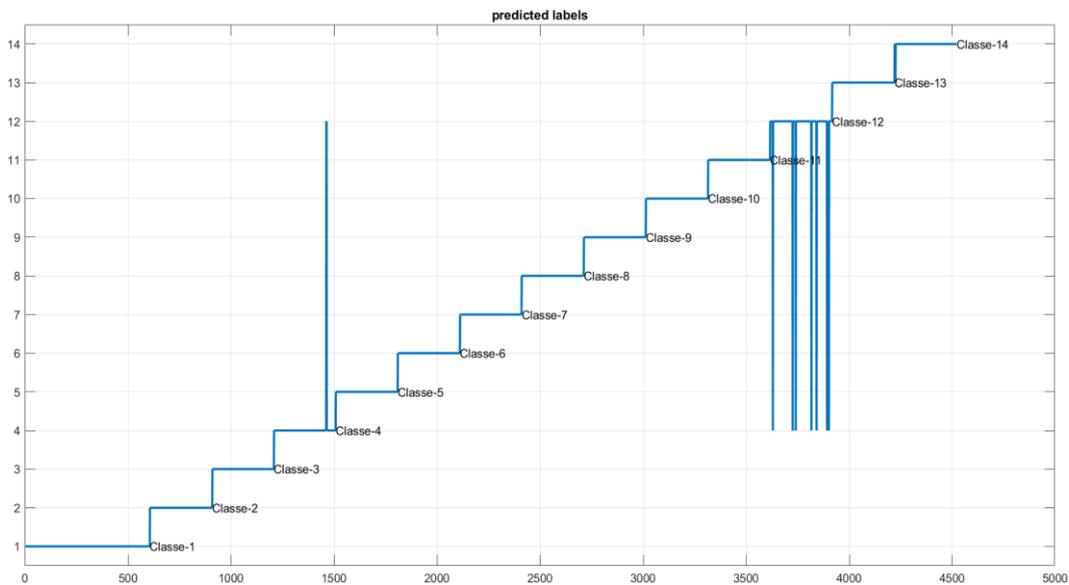


Figure 4.16: Prédiction des différents 14 classes.

On constate que :

- Les différents états de fonctionnement du système étudié sont bien classés et avec une précision (Accuracy) de 99.8%
- des défauts ont été enregistrés lors du processus de classification de la matrice de confusion, en raison de la confusion entre les valeurs des indicateurs du cas défaut normal et autre défaut.

	ACCURACY	ERROR	RECALL	SPECEFCITY	PRECISION	FALSEPOSITIVE RATE	F1_SCORE	MATTHEWSCORRELATION COEFFICIENT	KAPPA
FOLD 1	0,9956	0,0044	0,9953	0,9997	0,9953	0,0003	0,9953	0,9950	0,9669
FOLD 2	0,9989	0,0011	0,9988	0,9999	0,9988	0,0001	0,9988	0,9987	0,9917
FOLD 3	1,0000	0,0000	1,0000	1,0000	1,0000	0,0000	1,0000	1,0000	1,0000
FOLD 4	0,9989	0,0011	0,9988	0,9999	0,9988	0,0001	0,9988	0,9987	0,9916
FOLD 5	0,9956	0,0044	0,9952	0,9997	0,9953	0,0003	0,9952	0,9949	0,9665

Tableau 4.6: Résultat d'apprentissage pour les 14 classes.

IV.6.6 Comparaison avec d'autres méthodes

Les résultats de de méthode proposée (modèle BiLSTM & modèle GRU) sont comparés à des travaux de recherche portant sur le diagnostic des défauts de roulement à l'aide de la base de données CWRU. La comparaison de la classification des précisions avec les autres méthodes est présentée dans le tableau 9. Comme le montrent les résultats, l'approche proposée, qui traite un nombre de classes plus élevé que les autres méthodes, présente les meilleures performances. Les résultats obtenus démontrent l'efficacité de la méthode basée sur le modèle BiLSTM dans le diagnostic des défauts de roulements [1].

Méthode	Nombre de classes	Test accuracy (%)
Improved CNN+Softmax	10	99.12
Deep CNN+RF	10	97.73
Multi-scale CNN LSTM	10	98.46
TF-WConvLSTM	10	98.48
CNN-SVM	12	99.44
EEMD- CNN	12	99,87
Méthode propose “BiLSTM”	15	99.50
Méthode propose “GRU”	15	99.50
Méthode propose “BiLSTM”	14	99.80
Méthode propose “BiLSTM”	14	99.70

Tableau 4. 7: Comparaison de la précision de la classification sur l'ensemble de données portant sur l'université Case Western Reserve.

IV.7 Recherche du modèle optimal

La tâche d'apprentissage en profondeur est divisée en deux étapes : training et test. Training est le processus d'optimisation du modèle. Une tâche importante pour les modèles d'apprentissage en profondeur est le réglage des paramètres. Le choix approprié des paramètres d'apprentissage améliore les performances du modèle telles que la précision de la classification et la vitesse de convergence. Cette performance est très sensible au taux d'apprentissage initial, à l'optimiseur d'apprentissage et aux paramètres de régularisation. Dans cette section, le modèle conçu est testé pour des tailles de lot de 65 ,85 respectivement et pour un taux d'apprentissage de 0,001. Les résultats sont présentés dans le tableau 5 pour

l'algorithme d'optimisation adaptative du moment (ADAM) et pour l'algorithme de descente de gradient stochastique avec impulsion (SGDM). Comme on peut le voir sur les résultats présentés dans les tableaux, le meilleur taux de précision égal à 99,7 est obtenu pour une taille de lot de 85, un taux d'apprentissage égal à 0,001, et pour l'optimiseur Adam. Nous notons également que pour ce taux d'apprentissage, le taux de précision de la prédiction est supérieur à 97,40% quelle que soit la valeur de la taille du lot, alors qu'un taux d'apprentissage élevé peut entraîner une diminution de la précision de la prédiction. Le tableau montre que les résultats de précision de prédiction pour le modèle utilisant l'optimiseur SGDM sont largement inférieurs à ceux obtenus pour le modèle appliquant l'optimiseur ADAM.

Optimizer	ADAM		SGDM	
MiniBatchSize	85	65	85	65
Accuracy %	99.7	97.4	95.9	97.4

Tableau 4.8: Comparison of test results for different learning rates and batch sizes (Optimizer = Adam and Sgdm).

IV.8 Conclusion

Ce chapitre présente les résultats des expériences menées pour évaluer la performance et l'efficacité de la nouvelle méthode basée sur les réseaux neuronaux BILSTM pour le diagnostic des défauts de roulements. L'évaluation a été réalisée en comparant l'application de la méthode proposée avec deux options, ADAM et SGDM. De plus, notre système de diagnostic a été comparé à la méthode GRU en utilisant différentes caractéristiques et en évaluant son fonctionnement dans des conditions variées de charge et de vitesse.

Les résultats ont démontré que la méthode BILSTM, est plus efficace pour discriminer automatiquement entre les classes de défauts de roulements. Elle est capable d'identifier et de localiser avec succès les défauts de roulements avec une très grande précision, même avec une quantité limitée de données d'entraînement.

Ces résultats montrent le potentiel de la méthode BILSTM pour le diagnostic des défauts de roulements, surpassant même la méthode GRU dans cette application spécifique. Cela ouvre la voie à des améliorations futures et à l'application de cette méthode à d'autres systèmes rotatifs tels que les turbines et les boîtes de vitesses.

La méthode BILSTM offre une approche prometteuse et plus fiable pour le diagnostic des

défauts de roulements, ce qui peut conduire à des améliorations significatives dans la maintenance préventive et la fiabilité des systèmes industriels.

CONCLUSION GENERALE

Nous avons proposé deux méthodes de diagnostic des défauts de roulement, basées sur les réseaux de neurones BiLSTM et GRU. Les méthodes proposées ont été appliquées à la base de données de roulements CWRU. Les performances ont été testées pour quinze classes et pour différentes charges. Ces résultats sont supérieurs à ceux proposés dans la littérature traitant le même problème. Les résultats expérimentaux obtenus montrent que la méthode « BLSTM » a de meilleures performances par rapport la méthode « GRU ». De plus, ces résultats ont été comparés à ceux obtenus avec les techniques de diagnostic existantes dans la littérature, ce qui nous a permis de montrer que la méthode développée dans nos travaux de recherche pouvait atteindre des performances diagnostiques plus élevées et plus fiables. Les recherches futures peuvent être appliquées à d'autres machines tournantes telles que les turbines et les boîtes de vitesses. De plus, une exploration plus approfondie de la méthode développée peut être envisagée pour des données provenant de nouvelles sources d'informations, telles que des données acoustiques.

Bibliographie Générale

Chapitre I:

- [1] SALIOU, D. *Contribution au diagnostic industriel de défaut de roulement et de balourd par technique neuronales application à la machine asynchrone à cage*. 2007. Thèse de doctorat. Thèse de doctorat, Université Paris XII Val de Marne Creteil.
- [2] CHAHMI, Abdelghani. *Identification paramétrique de la machine asynchrone dédiée au diagnostic*. 2017. Thèse de doctorat. Université des sciences et technologie dOran-MB.
- [3]: BAYALA, Bapio. *La machine asynchrone. Edition revue*, 2010.
- [4] BOUGUERNE, Abla. "Diagnostic automatique des défauts des moteurs asynchrones "mémoire de Magister en électrotechnique. *Université MENTOURI Constantine, Année*, 2009.
- [5] BONNETT, Austin H. et YUNG, Chuck. Increased efficiency versus increased reliability. *IEEE Industry Applications Magazine*, 2008, vol. 14, no 1, p. 29-36.
- [6] ELFEKY, Ahmed A., MASOUD, Mahmoud I., et EL-ARABAWY, Ibrahim F. Production de signatures de défauts pour roulements à billes dans moteur à induction. In : *2007 Compatibilité en électronique de puissance*. IEEE, 2007. p. 1-5.
- [7] ELFEZAZI, S., MOKHLIS, A., BENMOUSSA, R., *et al.* Vers un outil, basé sur l'analyse fonctionnelle, pour la mise en œuvre des indicateurs de mesure de performance de la fonction maintenance. *Revue Française de gestion industrielle*, 2003, vol. 22, no 3, p. 77-92.
- [8] ILYES, KHELFI. *Diagnostic des machines tournantes par les techniques de l'intelligence artificielle*. 2014. Thèse de doctorat. Université Badji Mokhtar d'Annaba, Département de Génie Mé.
- [9]- DJEBILI, Omar. *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*. 2013. Thèse de doctorat. Reims.
- [10]- YIN, MENG, GREENLEAF, JAMES F., ET CHEN, SHIGAO. Biais observé dans les mesures de vitesse d'onde de cisaillement à temps de vol à l'aide de la force de rayonnement d'un faisceau à ultrasons focalisé heng zhao,* PENGFEI SONG,* MATTHEW W. URBAN,* RANDALL R. KINNICK. 2011.

Chapitre II :

- [1] LLORENS, Cédric, LEVIER, Laurent, VALOIS, Denis, *et al.* *Tableaux de bord de la sécurité réseau*. Editions Eyrolles, 2011.
- [2] GÉRON, Aurélien. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. «O'Reilly Media, Inc.», 2022.
- [3] ISLAM, Mohaiminul, CHEN, Guorong, ET JIN, Shangzhu. An overview of neural

network. *American Journal of Neural Networks and Applications*, 2019, vol. 5, no 1, p. 7-11.

[4] ELMAN, Jeffrey L. Trouver une structure dans le temps. *Sciences cognitives*, 1990, vol. 14, n° 2, p. 179-211.

[5] JORDAN, Michael I. Serial order: A parallel distributed processing approach. In: *Advances in psychology*. North-Holland, 1997. p. 471-495.

[6] JAEGER, Herbert ET HAAS, Harald. Exploiter la non-linéarité : prévoir des systèmes chaotiques et économiser de l'énergie dans les communications sans fil. *Sciences*, 2004, vol. 304, n° 5667, p. 78-80.

[7] JAEGER, Herbert ET HAAS, Harald. Exploiter la non-linéarité : prévoir des systèmes chaotiques et économiser de l'énergie dans les communications sans fil. *Sciences*, 2004, vol. 304, n° 5667, p. 78-80.

[8] BRAKNI, El Mahdi. *Réseaux de neurones artificiels appliqués à la méthode électromagnétique transitoire InfiniTEM*. Université du Québec en Abitibi-Témiscamingue (Canada), 2011.

Chapitre III :

[1] ARNOLD, Ludovic, REBECCHI, Sébastien, CHEVALLIER, Sylvain, *et al.* An introduction to deep learning. In: *European Symposium on Artificial Neural Networks (ESANN)*. 2011.

[2] ORDÓÑEZ, Francisco Javier ET ROGGEN, Daniel. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors*, 2016, vol. 16, no 1, p. 115.

[3] BROWNLEE, Jason. *Long short-term memory networks with python: develop sequence prediction models with deep learning*. Machine Learning Mastery, 2017.

[4] NACER, Saadi Mohamed, NADIA, Bouteraa, ABDELGHANI, Redjati, ET MOHAMED, Boughazi. A novel method for bearing fault diagnosis based on BiLSTM neural networks. *The International Journal of Advanced Manufacturing Technology*, 2023, p. 1-16.

Chapitre IV :

[1] NACER, Saadi Mohamed, NADIA, Bouteraa, ABDELGHANI, Redjati, ET MOHAMED, Boughazi. A novel method for bearing fault diagnosis based on BiLSTM neural networks. *The International Journal of Advanced Manufacturing Technology*, 2023, p. 1-16.