

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire
وزارة التعليم العالي و البحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique
جامعة سعد دحلب البلدية
Université SAAD DAHLAB de BLIDA
كلية التكنولوجيا
Faculté de Technologie
قسم الإلكترونيك
Département d'Électronique



Mémoire de Master

Mention : Génie Biomédical

Spécialité : Instrumentation Biomédical

présenté par : ISMAIL Fatima Zohra

Classification Du Signal ECG avec le Deep Learning.

Proposé par : Dr A . CHENTIR

Année Universitaire

2022 / 2023



Remerciements

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

Au terme de ce travail, je tiens tout d'abord à remercier mon encadreur Dr. A. CHENTIR qui m'a toujours soutenue par son aide et ses précieux conseils.

Je remercie tous les membres de ma famille pour leur aide, leur générosité et soutien moral qui ont été pour moi une source de courage et de confiance.

Je remercie aussi les honorables membres du jury qui ont accepté d'évaluer mon travail.

J'adresse mes sincères remerciements à tous les professeurs qui ont intervenu tout au long de ma formation universitaire et toutes les personnes qui par leurs paroles, leurs écrits, leurs conseils et leurs critiques ont guidé mes réflexions et ont accepté de me rencontrer et de répondre à mes questions durant mes études.

Enfin, un grand remerciement à tous ceux qui, par un mot, m'ont donné la force de continuer à travailler afin d'atteindre mes objectifs.

Dédicace

Aux plus chers êtres au monde : ma mère et mon père

*Pour leur amour, leur tendresse et pour leur soutien moral et matériel
durant toute les étapes de ma vie.*

A mon frère : Riad

A mes sœurs : Nadia, Soumia, Hafida, Asma et Maroua.

A mes neveux: Dhiaa E-dinne, Imade-Edinne, Wassim

Isam-Edinne, Alaa, Adam.

A tous mes chers amis

Résumé

L'électrocardiogramme (ECG) est un outil essentiel en médecine pour évaluer la santé cardiaque des patients. Il enregistre les impulsions électriques générées par le cœur et permet de détecter d'éventuelles anomalies. Dans ce projet, nous nous concentrons sur la classification des signaux ECG à l'aide des techniques de Deep Learning. Notre approche comprend deux méthodes : une classification binaire pour déterminer si le patient est atteint d'une maladie cardiaque ou non, suivie d'une classification multi-classe pour identifier le type spécifique de la maladie. Nous avons choisi d'utiliser deux algorithmes de Deep Learning, à savoir le perceptron multicouche (MLP) et le CNN, pour comparer leurs performances et choisir le plus optimal dans notre contexte. L'objectif est donc d'améliorer le diagnostic des maladies cardiovasculaires en automatisant la classification des signaux ECG.

Mots Clés: ECG, Classification, Deep Learning, MLP, CNN, Diagnostic.

Abstract

The electrocardiogram (ECG) is an essential tool in medicine to assess the heart health of patients. It records the electrical impulses generated by the heart and makes it possible to detect possible abnormalities. In this project, we focus on the classification of ECG signals using Deep Learning techniques. Our approach includes two methods : a binary classification to determine whether the patient has heart disease or not, followed by a multi-class classification to identify the specific type of the disease. We chose to use two Deep Learning algorithms, namely the multilayer perceptron (MLP) and the CNN, to compare their performances and choose the most optimal one in our context. The objective is then to improve the diagnosis of cardiovascular diseases by automating the classification of ECG signals.

Key Words: ECG, Classification, Deep Learning, MLP, CNN, Diagnostic.

ملخص

يعد مخطط كهربية القلب أداة أساسية في الطب لتقييم صحة القلب للمرضى. يسجل النبضات الكهربائية التي يولدها القلب ويجعل من الممكن اكتشاف التشوهات المحتملة. في هذا المشروع، نركز على تصنيف إشارات تخطيط القلب باستخدام تقنيات التعلم العميق. يتضمن نهجنا طريقتين: تصنيف ثنائي لتحديد ما إذا كان المريض يعاني من أمراض القلب أم لا، متبوعاً بتصنيف متعدد الفئات لتحديد النوع المحدد للمرض. اخترنا استخدام اثنين من خوارزميات التعلم العميق، وهي متعدد الطبقات بيرسيبترون (MLP) و CNN، لمقارنة أدائها واختيار الأكثر الأمثل واحد في سياقنا. الهدف هو تحسين تشخيص أمراض القلب والأوعية الدموية عن طريق أتمتة تصنيف إشارات تخطيط القلب.

الكلمات المفتاحية: تخطيط القلب، التصنيف، التعلم العميق، التشخيص، MLP ، CNN .

Table des Matières

Table des matières

Remerciements

Dédicaces

Résumés

Table des matières

Liste des Figures

Liste des Tableaux

Liste des Abréviations

Introduction Générale 1

Chapitre 1 : Généralités sur l'Électrocardiogramme

1.1	Introduction	3
1.2	Anatomie du Cœur	3
1.3	L'électrocardiogramme	5
1.3.1	L'Électrocardiogramme	6
1.3.2	Les ondes du signal ECG	8
1.3.3	Les segments et intervalles qui caractérisent un ECG normal	9
1.3.4	Les pathologies cardiaques	10
1.4	Conclusion	12

Chapitre 2 : Le Deep Learning

2.1	Introduction	13
2.2	Deep Learning : définition	13
2.3	Applications du Deep Learning	15
2.4	Le réseau neuronal	16
2.4.1	Le neurone	17
2.4.2	Fonctions d'activation	18
2.4.3	Les types des réseaux de neurones	20
2.5	Apprentissage en Deep Learning	27
2.5.1	Variantes de la descente du gradient	27
2.5.2	Algorithmes d'optimisation de la descente du gradient	28
2.6	Conclusion	28

Chapitre 3 : État de l'art

3.1	Introduction	29
-----	--------------	----

3.2	Approche 1 : ECG arrhythmia classification using a 2-D convolutional neural network 29	
3.3	Approche 2: Classification of ECG arrhythmias using recurrent neural networks ...	32
3.4	Approche 3 : Application of Deep convolutional neural network for automated detection of myocardial infarction using ECG signals	34
3.5	Approche 4 : ECG Heartbeat Classification : A Deep Transferable Representation	37
3.6	Approche 5 : ECG Classification Using Wavelet Packet Entropy and Random Forests 38	
3.7	Approche 6 : Application of Higher Order Cumulant Features for Cardiac Health Diagnosis Using ECG Signals.....	40
3.8	Approche 7: Classification of Cardiac Arrhythmias by Artificial Neural Networks.	42
3.9	Comparaison des approches	45
3.10	Conclusion.....	45

Chapitre 4 : Système Réalisé

4.1	Introduction	46
4.2	Justification de la méthodologie suivie	47
4.3	Base de données utilisée.....	47
4.3.1	Prétraitement de données	48
4.3.2	Aperçu sur la base de données	48
4.4	Les mesures de performances.....	48
4.5	Présentation des outils utilisés.....	51
4.5.1	Visual Studio Code	51
4.5.2	Python	51
4.5.3	TensorFlow	52
4.5.4	Keras	53
4.5.5	Scikit-learn.....	53
4.6	Algorithmes utilisés.....	54
4.6.1	Algorithme MLP	54
4.6.2	Algorithme CNN.....	61
4.7	Comparaison des algorithmes utilisés	70
4.7.1	Comparaison pour la classification binaire	70
4.7.2	Comparaison pour la classification multi-classe.....	71
4.8	Comparaison entre les méthodes de l'état d'art et celles du système utilisé	72
4.9	Conclusion.....	72
	Conclusion Générale	74
	Bibliographie.....	75

Liste des Figures

&

Liste des Tableaux

Liste des Figures

Figure 1.1: Schéma général de cœur	4
Figure 1.2: Circulation du sang dans le cœur	5
Figure 1.3: Les ondes de signal ECG	9
Figure 1.4: Les intervalles et les segments du signal ECG	10
Figure 1.5: Les types des résultats ECG (1 normal, 4 anormaux).....	12
Figure 2.1: La Relation entre IA, ML et DL	13
Figure 2.2 : Représentation d'un neurone biologique et d'un neurone artificiel	17
Figure 2.3: Neurone artificiel	17
Figure 2.4: Représentation graphique de la fonction Sigmoidale	18
Figure 2.5: Représentation Graphique de la fonction ReLu	19
Figure 2.6 : Représentation graphique de la fonction Softmax	19
Figure 2.7 : Perceptron Multicouche	20
Figure 2.8 : Architecture du Modèle CNN	21
Figure 2.9: Types de séquences d'entrée pour un réseau récurrent	22
Figure 2.10 : Exemple d'un réseau récurrent qui se déroule	22
Figure 2.11 : Chaîne de Cellules LSTM	24
Figure 2.12 : Porte d'oubli.....	24
Figure 2.13 : Porte d'entrée	25
Figure 2.14 : La porte de sortie	25
Figure 2.15: Résumé des types d'architectures de réseaux de neurones.....	26
Figure 2.16 : Illustration du processus de recherche de l'optimum	27
Figure 3.1: Procédures globales traitées dans la classification des arythmies ECG.	29
Figure 3.2: L'architecture globale de modèle CNN	31
Figure 3.3: Les résultats des différents modèles CNN	32
Figure 3.4 : Les étapes de détection d'arythmie ECG proposées	33
Figure 3.5 : Les résultats des différents modèles RNN.....	34
Figure 3.6 : Les caractéristiques des données ECG obtenues à partir de la base de données PTB.....	35
Figure 3.7: L'architecture du CNN proposée	35
Figure 3.8: les résultats obtenus dans le travail	36
Figure 3.9: Comparaison des résultats de la classification du rythme cardiaque.....	37
Figure 3.10: Comparaison des résultats de la classification MI.....	37

Figure 3.11: Organigramme de la méthode proposée	39
Figure 3.12 : Résultats (en %) obtenus par la méthode proposée et certains travaux antérieurs suite à la Recommandations de l'AAMI et schéma inter-patients.....	39
Figure 3.13 : Système proposé pour la classification des battements ECG	41
Figure 3.14 : Résultats de classification des différents classificateurs pour.	41
Figure 3.15: Résultats de classification des différents classificateurs pour	42
Figure 3.16: Patients et nombre de battements utilisés	43
Figure 3.17: (a) Taux de reconnaissance en% du classifieur, (b) : Matrice de confusion entre classes	44
Figure 4.1: L'architecture globale de la classification des signaux ECG.....	46
Figure 4.2 : Les 5 premières lignes de la base de données.....	48
Figure 4.3 : Matrice de confusion	49
Figure 4.4 : Logo Python.....	52
Figure 4.5 : Logo TensorFlow.....	52
Figure 4.6 : Logo Keras	53
Figure 4.7 : Logo Scikit-Learn	53
Figure 4.8 : Différents algorithmes utilisés.....	54
Figure 4.9 : Organigramme de L'architecture MLP	55
Figure 4.10 : Sous-organigramme pour le traitement des donnée,(a) :binaire ,(b) : multiclasse.....	56
Figure 4.11: Le modèle MLP binaire.	56
Figure 4.12: Le modèle MLP multiclasse.	58
Figure 4.13 : La matrice de confusion MLP Binaire.....	59
Figure 4.14: Résultats MLP Binaire	60
Figure 4.15 : La matrice de confusion MLP Multiclasse.	60
Figure 4.16 : Résultats MLP Multiclasse.	61
Figure 4.17 : Organigramme de L'architecture CNN.	62
Figure 4.18: Le modèle CNN binaire.	63
Figure 4.19: Le modèle CNN multiclasse.	65
Figure 4.20 : La matrice de confusion CNN Binaire.....	66
Figure 4.21 : Résultat CNN Binaire.	67
Figure 4.22 : La matrice de confusion CNN Multi-classe.....	67
Figure 4.23 : Résultat CNN Multi-classe.	68

Figure 4.24 : La matrice de confusion CNN Multi.	69
Figure 4.25 : Résultat CNN Multi.	69
Figure 4.26: L'architecture finale de la classification des arythmies ECG.	72

Liste des Tableaux

Tableau 2.1 : Résumé de l'histoire du Deep Learning	15
Tableau 4.1 : Comparaison entre le MLP et le CNN (cas de classification binaire).....	70
Tableau 4.2 : Comparaison pour classification multi-classe	71

Liste des Abréviations

Liste des Abréviation

AG :	Algorithmes Génétiques
AAMI :	Association pour l'Avancement de l'Instrumentation Médicale
AV :	Auriculo-Ventriculaire
APC :	Contraction Auriculaire Préaturée
BBD :	Bloc de Branche Droit
BBG :	Bloc de Branche Gauche
CNN :	Réseaux de Neurones Convolutifs
CSV :	valeurs séparées par des virgules
DL :	Deep Learning
EV :	Extrasystole Ventriculaire
EA :	Extrasystole Auriculaire
ECG :	Electrocardiogramme
IA :	Intelligence Artificiel
GRU :	Gated Recurrent Unit
LF :	Logique Floue
LSTM:	Long Short Term Memory
LVQ :	Learning Vector Quantization
ML :	Machine Learning
MLP :	Multi Layer Perceptron
MCV :	Maladies Cardiovasculaires
MI :	Myocardie Infraction

OMS : Organisation Mondiale de la Santé

PVC : Contraction Ventriculaire Prématurée

PAB : Battement du rythme

PTB: Physikalisch-Technische Bundesanstalt diagnostic

RN : Réseaux de Neurones

RBB : Bloc de branche droit

RNN : Réseaux de Neurones Récurrents

RF : Forêts Aléatoires

SAMU : Service d'Aide Médicale Urgente

VPC : Contraction Prématurée Ventriculaire

WPE : l'Entropie par Paquets d'Ondelettes

WPD : Décomposition en Paquets d'Ondelettes

Introduction

Générale



À l'heure actuelle, le domaine médical a besoin de nouvelles techniques et technologies pour évaluer l'information de façon objective. Cela est dû à l'évolution récente de l'électronique qui a poussé l'informatique à un stade de plus en plus avancé. Ceci a permis d'avoir des machines de plus en plus performantes permettant d'exécuter des algorithmes complexes et de tester de nouvelles approches de l'intelligence artificielle (IA) qui s'avéraient impossibles auparavant.

Ainsi, la médecine, comme de nombreux autres domaines, a bénéficié de cette révolution dans l'informatique, en particulier l'IA. Dans ce contexte, l'informatique est devenue un outil incontournable dans la pratique médicale moderne en général et comme support d'aide au diagnostic en particulier.

Plusieurs techniques de l'IA, de la Logique Floue (LF), des Réseaux de Neurones (RN), les Algorithmes Génétiques (AG), sont couramment utilisées dans des applications médicales en vue d'améliorer la performance des systèmes d'aide au diagnostic médical.

Selon l'Organisation mondiale de la santé (OMS), les Maladies CardioVasculaires (MCV) sont la première cause de décès aujourd'hui. Plus de 17,7 millions de personnes sont mortes de MCV en 2017 dans le monde entier, ce qui représente environ 31% de tous les décès, et plus de 75% de ces victimes arrivent dans des pays à revenu faible ou intermédiaire [1].

Ainsi, les praticiens surveillent la fonction du cœur par l'enregistrement d'un signal électrique variant avec le temps et qui s'appelle électrocardiogramme (ECG). Celui-ci est distingué par un rythme périodique ou quasi périodique pouvant être représenté graphiquement et qui fournit donc les informations nécessaires en rapport avec le cœur d'un patient. Il présente l'apport le plus important pour diagnostiquer et traiter les pathologies cardiovasculaires. C'est aussi un instrument important qui permet de surveiller les patients chez eux. Ce qui a conduit à l'avancement des applications en télémédecine.

En réalité, l'examen ECG est un outil non invasif effectué par le médecin en vue d'explorer le fonctionnement du cœur par l'emploi des électrodes externes mises en contact de la peau. Il s'agit d'un signal qui reflète l'activité électrique du cœur. À partir du signal ECG, certains paramètres importants peuvent être extraits. En règle générale, les durées et les formes des différentes ondes sont considérées comme des signes indicateurs de certaines anomalies cardiaques.

En raison du taux de mortalité élevé des maladies cardiaques, une détection et une classification fidèles des arythmies ECG sont essentielles pour le traitement des patients. Ces dernières années, de nombreux algorithmes ont été développés pour la détection et la classification des signaux

La classification des signaux cardiaques est considérée comme une aide à la décision médicale, elle englobe plusieurs disciplines, comme l'intelligence artificielle, le traitement du signal, l'ingénierie du logiciel et des connaissances, avec en plus, l'aspect médical.

Dans le cadre de ce travail, nous proposons une approche pour la classification des signaux ECG. Cette approche est basée sur des méthodes de Deep Learning.

Le présent document est reparti en quatre chapitres :

- le premier chapitre concerne la présentation du principe de fonctionnement de l'électrocardiogramme,
- le deuxième chapitre décrit ce qu'est l'apprentissage profond,
- le troisième chapitre présente un état de l'art qui décrit les études récentes déjà réalisées dans ce domaine et ayant un lien avec notre problématique,
- puis on aborde le dernier chapitre qui explique en détails les différentes étapes et méthodes utilisées pour la classification des signaux ECG.
- Et enfin, une conclusion générale clôture ce mémoire.

Chapitre 1 : Généralités sur l'Électrocardiogramme



1.1 Introduction

L'électrocardiographie, communément appelée ECG, est une technique médicale utilisée pour enregistrer l'activité électrique du cœur. C'est un outil précieux et largement utilisé dans le domaine de la cardiologie pour évaluer la santé cardiaque d'un individu. L'ECG permet de détecter et d'analyser les anomalies de l'activité électrique du cœur, ce qui aide les médecins à diagnostiquer divers problèmes cardiaques tels que les maladies coronariennes, les troubles du rythme cardiaque, les infarctus du myocarde et d'autres affections cardiaques.

L'enregistrement d'un ECG est réalisé à l'aide d'un appareil appelé électrocardiographe, qui mesure et amplifie les impulsions électriques générées par le cœur. Des électrodes sont placées stratégiquement sur la poitrine, les bras et les jambes du patient pour capturer ces signaux électriques. L'activité électrique est ensuite tracée sous forme d'un graphique appelé électrocardiogramme.

Un électrocardiogramme typique comporte plusieurs ondes distinctes qui correspondent à différents événements électriques du cycle cardiaque. Ces ondes, connues sous les noms de P, Q, R, S et T, fournissent des informations sur la dépolarisation et la repolarisation des différentes parties du cœur. L'analyse de ces ondes permet aux médecins d'identifier les problèmes potentiels et d'évaluer la fonction cardiaque globale.

L'ECG est une procédure non invasive, rapide et relativement simple. Il est couramment utilisé lors des examens médicaux de routine, des urgences, des évaluations préopératoires et du suivi des patients atteints de maladies cardiaques. Les résultats de l'ECG peuvent aider les médecins à prendre des décisions éclairées concernant le traitement des patients, y compris la prescription de médicaments, l'intervention chirurgicale ou la recommandation de procédures supplémentaires.

1.2 Anatomie du Cœur

Le cœur est un organe musculaire creux situé dans la cavité thoracique, légèrement sur la gauche du sternum.

La structure interne du cœur se compose de 3 couches avec de l'intérieur vers l'extérieur : **l'endocarde, le myocarde et le péricarde.**

Le péricarde est un sac à double paroi qui enveloppe le cœur. **Le myocarde** constitue le muscle cardiaque proprement dit, c'est un muscle strié c'est-à-dire qu'il possède la même structure que les muscles des membres et la même force de contraction. **L'endocarde** est une membrane endothéliale qui tapisse la face interne du cœur.

Le cœur est divisé en 4 cavités (Figure 1.1) :

- **cavités supérieures** : les oreillettes droite et gauche séparées par le septum inter auriculaire.
- **cavités inférieures** : les ventricules droit et gauche séparés par le septum intraventriculaire

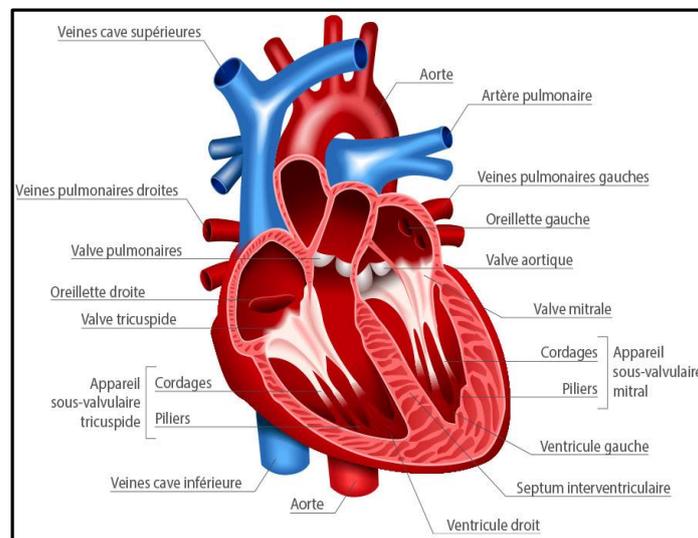


Figure 1.1: Schéma général de cœur [2]

a) Fonctionnement du cœur

Le cœur fonctionne comme une pompe qui assure la circulation du sang dans tout le corps. Il se contracte et se détend de manière rythmique pour maintenir le flux sanguin.

Le processus commence par la contraction des oreillettes, ce qui permet au sang de passer dans les ventricules. Ensuite, les ventricules se contractent, propulsant le sang hors du cœur. Le ventricule droit envoie le sang vers les poumons pour être oxygéné, tandis que le ventricule gauche le pompe vers le reste du corps pour fournir de l'oxygène et des nutriments aux organes et aux tissus.

Les valves cardiaques s'ouvrent et se ferment au bon moment pour contrôler le flux sanguin. Elles s'ouvrent lorsque les cavités se relâchent pour permettre au sang de passer, puis se ferment lors des contractions pour empêcher le reflux.

Le système électrique du cœur régule cette séquence de contractions et de relaxations. Il génère des impulsions électriques qui se propagent à travers les cellules musculaires cardiaques, provoquant leur contraction synchronisée.

Ce cycle se répète sans relâche, assurant une circulation continue du sang dans le corps, fournissant ainsi l'oxygène et les nutriments nécessaires aux tissus et éliminant les déchets métaboliques (Figure 1.2).

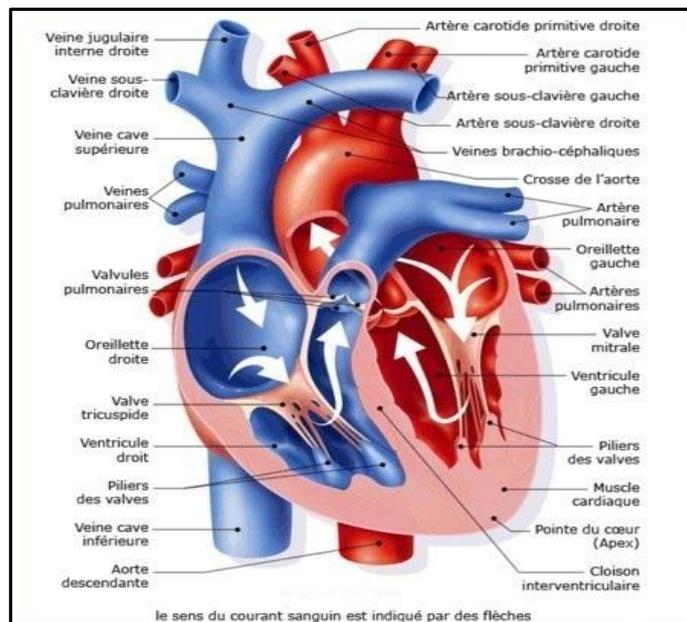


Figure 1.2: Circulation du sang dans le cœur [3].

1.3 L'électrocardiogramme

Un électrocardiogramme est un appareil médical utilisé pour enregistrer l'électrocardiogramme du cœur. Cet appareil a été inventé par Willem Einthoven en 1895 (Prix Nobel en 1924). Il mesure les impulsions électriques cardiaques à l'aide d'électrodes, les amplifie, les convertit en signaux numériques et produit un électrocardiogramme qui permet aux médecins d'évaluer la santé cardiaque et de diagnostiquer les problèmes cardiaques.

1.3.1 L'Électrocardiogramme

Un électrocardiogramme (ECG) est un test qui étudie le fonctionnement du cœur en mesurant son activité électrique. À chaque battement cardiaque, une impulsion électrique (ou « onde ») traverse le cœur. Cette onde fait contracter le muscle cardiaque afin qu'il expulse le sang du cœur. Un ECG représente de ce fait un outil de mesure et enregistre l'activité électrique du cœur [4].

a) Pourquoi passer un électrocardiogramme ?

L'ECG est un examen essentiel en cardiologie. Il est peut être prescrit dans de nombreuses situations :

- en cas de douleurs thoraciques.
- pour détecter des arythmies (troubles du rythme cardiaque).
- pour détecter d'autres troubles cardiaques comme une insuffisance coronarienne (blocage des artères irriguant le cœur), la présence de zones endommagées dans le cœur (en raison d'un manque d'irrigation ou d'une crise cardiaque récente), la dilatation du cœur, etc.
- pour surveiller l'activité cardiaque en cas de maladie cardiaque connue.
- en cas d'admission en urgence pour une douleur thoracique.
- lors d'un bilan pré-opératoire.

b) Types d'électrocardiogrammes

L'électrocardiogramme constitue un examen indolore, sans risques, Il existe plusieurs types d'électrocardiogrammes :

- **L'électrocardiogramme de repos** : est le cas le plus fréquent en pratique :
 - Il est réalisé au cours d'une consultation médicale lorsque le patient est allongé sur le dos.
 - Mais il peut aussi être effectué lors d'une intervention d'urgence par les pompiers ou le SAMU grâce à des appareils portatifs.
- **L'électrocardiogramme en continu pendant un effort** : est enregistré pendant toute la durée d'un effort physique du patient (sur un vélo ou sur un tapis roulant) :
 - Cet ECG est souvent appelé le test à l'effort.

- Cet électrocardiogramme est prescrit lorsque le patient décrit des palpitations ou des douleurs à la poitrine, mais que son électrocardiogramme de repos est normal.
- **L'Holter-ECG** : est enregistré pendant 24 heures au cours des activités quotidiennes du patient.
 - Cet électrocardiogramme est notamment utilisé pour déterminer à quel moment de la journée les troubles cardiaques du patient sont les plus importants.

c) Comment se préparer à passer un électrocardiogramme ?

Aucune préparation n'est requise avant l'examen :

- Il est inutile d'être à jeun.
- Aucun produit n'est injecté ni avant, ni au cours de l'examen.
- Si le patient prend des médicaments, il n'est pas nécessaire d'interrompre le traitement, mais le patient doit absolument indiquer au médecin l'ensemble des médicaments pris. En effet, certains médicaments peuvent modifier le tracé de l'électrocardiogramme.
- Il est toutefois recommandé de s'abstenir de fumer.

d) Le déroulement de l'électrocardiogramme

Pour effectuer l'électrocardiogramme, le médecin applique sur la peau une dizaine de petites électrodes (petits disques métalliques collés sur la peau grâce à des patchs) disposées au niveau des bras, des jambes et de la poitrine. Pour une meilleure adhérence des électrodes, il peut être nécessaire de raser quelques zones de la peau. Puis l'examen se déroule de la manière suivante selon le type d'ECG :

- L'enregistrement de l'électrocardiogramme de repos dure **entre 5 et 10 minutes** :
 - Il est important pendant l'enregistrement de ne pas parler, de rester calme et détendu, pour éviter de perturber le tracé.
 - Le médecin peut demander au patient de retenir son souffle à certains moments de l'enregistrement.

- L'électrocardiogramme en continu pendant un effort dure **entre 10 et 30 minutes** :
 - L'intensité de l'effort demandé au patient est augmentée par paliers.
 - L'examen dure jusqu'à ce que le patient signale l'apparition de la fatigue au médecin.
 - Le patient ne doit pas parler pendant l'enregistrement, sauf pour signaler tout symptôme anormal.
 - L'enregistrement est suivi par une période de récupération pendant laquelle le patient reste sous la surveillance du médecin.

- L'Holter-ECG dure **au minimum 24 h** :
 - Un petit appareil portatif d'enregistrement de l'ECG est relié aux électrodes et accompagne le patient dans ses activités quotidiennes.
 - Le patient doit noter les moments où il ressent des symptômes particuliers pour faciliter l'interprétation des résultats de l'examen.

1.3.2 Les ondes du signal ECG

Le signal ECG regroupe un ensemble d'ondes connues par l'onde P, Q, R, S, et l'onde T qui sont en corrélation directe avec l'activité cardiaque. Le signal (Figure 1.3) représente la morphologie et l'amplitude de ces ondes mais la durée change selon les dérivations ECG utilisées.

- **Onde P** : C'est la première onde détectable, Elle représente la dépolarisation auriculaire. Cette onde peut être positive ou négative avec une durée de l'ordre de 90 ms. Généralement son observation est difficile, spécialement dans des conditions bruitées. Il faut noter que la repolarisation auriculaire n'est pas visible sur l'ECG car elle coïncide avec le complexe QRS d'amplitude plus importante [5].
- **Le complexe QRS** : correspond à un ensemble de déflexions positives et négatives dues à la dépolarisation des ventricules, Pour un cas normal, il a une durée inférieure à 0.12 secondes et son amplitude variable est comprise entre 5 et 20 mV.
- **L'onde Q** est la première : c'est une onde dirigé vers le bas, qui n'est pas toujours visible sur le tracé
- **l'onde R** : elle est de grande amplitude et dirigée vers le haut

- **l'onde S** : c'est la dernière et est dirigée vers le bas
- **L'onde T** : Elle correspond à la repolarisation des ventricules, qui peut être négative, positive ou bi phasique et qui a normalement une amplitude plus faible que le complexe QRS. Bien que la dépolarisation et la repolarisation des ventricules soient des événements opposés, l'onde T est normalement du même signe que l'onde R, ce qui indique que la dépolarisation et la repolarisation ne sont pas symétriques [6].

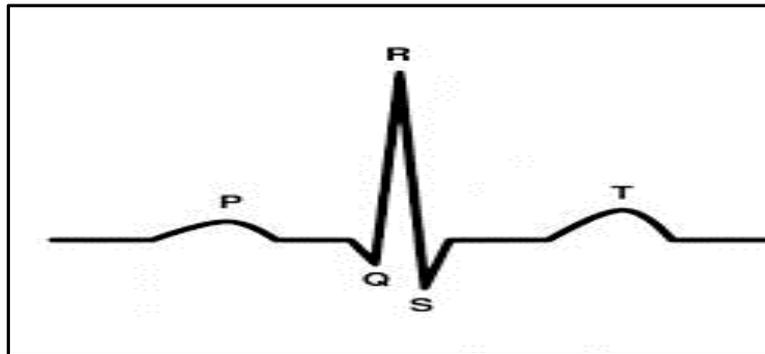


Figure 1.3: Les ondes de signal ECG [7].

1.3.3 Les segments et intervalles qui caractérisent un ECG normal

En plus des différentes ondes qui sont les paramètres de base pour une bonne caractérisation d'un signal ECG, il existe un certain nombre d'intervalles et de segments qui portent des informations très utiles sur la vitesse de conduction de l'impulsion électrique dans les différentes parties du cœur.

Les intervalles et les segments les plus importants sont les suivants (Figure 1.4) :

- **Intervalle RR** : qui correspond à la durée de propagation de l'onde de dépolarisation du nœud sinusal jusqu'aux cellules myocardiques ventriculaires.
- **Segment PR** : (pause du nœud AV) qui correspond au délai entre la fin de la dépolarisation des oreillettes et le début de celle des ventricules. C'est le temps pendant lequel l'onde de dépolarisation est bloquée au niveau du nœud AV.
- **Intervalle PR** : (durée de conduction auriculo-ventriculaire) qui correspond à la durée de propagation de l'onde de dépolarisation du nœud sinusal jusqu'aux cellules myocardiques ventriculaires [8].

- **Intervalle QT** : (durée de systole ventriculaire) Cet intervalle correspond au temps de systole ventriculaire, qui va du début de l'excitation des ventricules jusqu'à la fin de leur relaxation.
- **Segment ST** : (durée de stimulation complète des ventricules) qui correspond à la phase pendant laquelle les cellules ventriculaires sont toutes dépolarisées, le segment est alors isoélectrique.

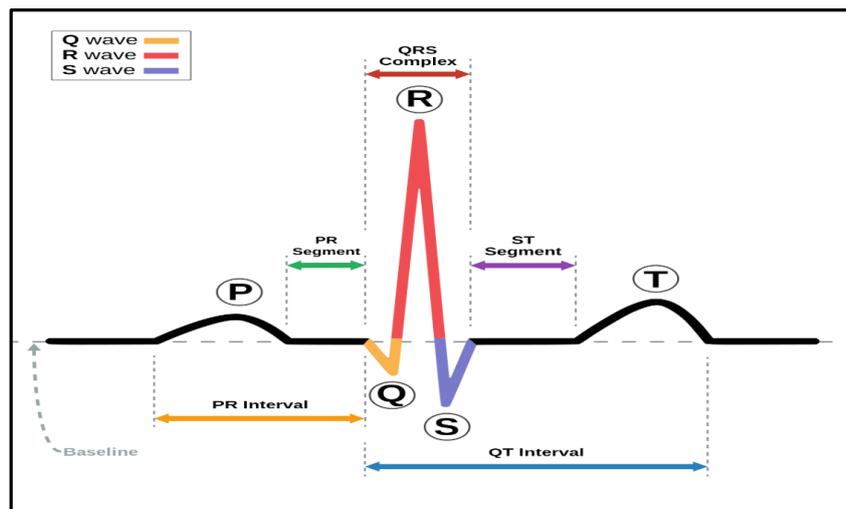


Figure 1.4: Les intervalles et les segments du signal ECG [9].

1.3.4 Les pathologies cardiaques

Cette section décrit brièvement les différentes pathologies cardiaques susceptibles d'être repérées en étudiant le signal ECG.

L'objectif, ici, n'est pas d'analyser précisément leurs origines et leurs conséquences sur le fonctionnement cardiaque, ni de décrire les traitements que ces pathologies nécessitent, mais simplement de mettre en relation certaines observations anormales du tracé ECG avec les pathologies les plus courantes. Parmi les maladies les plus rencontrées, ceux qui affectent le rythme cardiaque et qui sont appelées les arythmies cardiaques. Mais avant de parler des arythmies, il est intéressant de connaître les caractéristiques du rythme normale appelé aussi **rythme sinusal**.

- **Rythme sinusal** : Le rythme sinusal est le rythme normal cardiaque. Il correspond à une activation physiologique des oreillettes, puis des ventricules, à

partir du nœud sinusal. Son rythme est compris entre 60 à 80 battements par minute avec un intervalle régulier entre des battements normaux.

- **Les arythmies cardiaques :** L'arythmie est une anomalie qui affecte le rythme cardiaque normal. Plusieurs formes d'arythmie ne présentent aucun problème de santé ; cependant, elles peuvent entraîner divers symptômes inconfortables, comme des étourdissements ou une douleur dans la poitrine. D'autres types d'arythmie, plus dangereux, ont un impact sur la circulation sanguine et nécessitent donc une prise en charge médicale.

Dans notre projet, nous allons utiliser 4 catégories d'arythmie (Figure 1.5).

- **Contraction ventriculaire prématurée (PVC) :** représentée par des battements cardiaques anormaux qui proviennent des ventricules du cœur. Elles sont causées par une irritabilité électrique du muscle cardiaque des ventricules et perturbent le rythme normal du cœur, provoquant une bascule ou un battement sauté dans la poitrine. Les contractions ventriculaires prématurées sont très fréquentes et sont susceptibles d'apparaître chez les personnes âgées, les personnes hypertendues et les personnes souffrant de maladies cardiaques.
- **Contraction auriculaire prématurée (APC) :** qui est une dépolarisation prématurée prenant naissance en un point quelconque des oreillettes, habituellement mais non constamment propagée aux ventricules. La morphologie de l'onde P extrasystolique est différente de celle de l'onde P sinusale et dépend de la situation du foyer auriculaire ectopique.
- **Bloc de branche droit (RBB) :** Lorsqu'il existe un bloc de branche droit, le ventricule droit n'est pas directement activé par les impulsions électriques progressant au travers la branche droite du faisceau de His. Le ventricule gauche, quant à lui, est normalement activé par la branche gauche. Ces impulsions électriques progressent alors au travers du myocarde, du ventricule gauche vers le ventricule droit, activant finalement celui-ci. La prévalence du bloc de branche droit augmente avec l'âge.
- **Battement du rythme (Paced Beat : PAB) :** qui est en général la manifestation d'un dysfonctionnement du muscle cardiaque. Il peut apparaître à tout âge, mais il est fréquemment la conséquence d'un infarctus du myocarde. Il se manifeste par la désynchronisation des battements qui conduit à une diminution du pompage du sang [10].

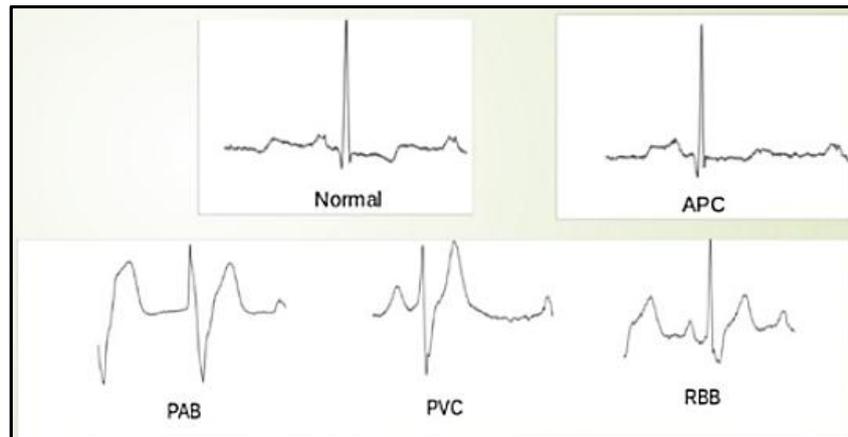


Figure 1.5: Les types des résultats ECG (1 normal, 4 anormaux) [11].

1.4 Conclusion

Dans ce chapitre, nous avons parlé de l'électrocardiogramme. Nous avons commencé par donner une brève introduction sur la structure anatomique du cœur, puis nous avons expliqué ce qu'est un électrocardiogramme, les différents types d'électrocardiogrammes, comment il est réalisé, et les différentes ondes, segments et intervalles qui le composent. Enfin, nous avons abordé quelques exemples d'arythmies cardiaques.

Chapitre 2 : Le Deep Learning



2.1 Introduction

Avant de présenter ce qu'est l'apprentissage profond (DL : Deep Learning), nous devons tout d'abord introduire le concept de l'intelligence artificielle.

L'intelligence artificielle ou encore IA, est un ensemble d'algorithmes conférant à une machine des capacités d'analyse et de décision lui permettant de s'adapter intelligemment aux situations en faisant des prédictions à partir de données déjà acquises.

Le Machine Learning et le Deep learning font partie de l'intelligence artificielle. Ces approches ont toutes deux pour résultat de donner aux ordinateurs la capacité de prendre des décisions intelligentes. La figure 2.1 montre la relation entre les trois concepts IA, ML et DL .

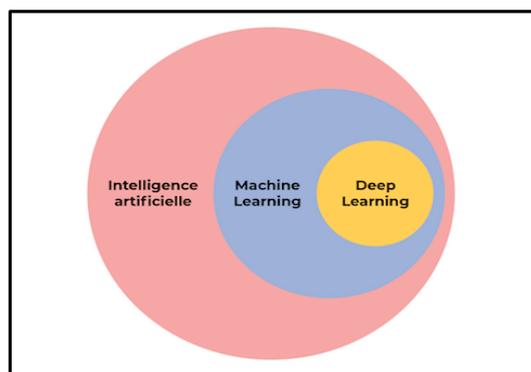


Figure 2.1: La Relation entre IA, ML et DL [12].

2.2 Deep Learning : définition

Le Deep Learning qui est un type d'intelligence artificielle, est une branche de l'informatique fondamentale permettant de simuler des comportements du cerveau humain.

Plus précisément, le Deep Learning est une technologie d'apprentissage et de classification basée sur des réseaux de neurones artificiels permettant d'apprendre à reconnaître des images, des voix, des textes.

Le Deep Learning a été formalisé en 2007 à partir de nouvelles architectures de réseaux de neurones. Mais c'est seulement, depuis 2012 que les recherches sur le Deep Learning ont abouti à l'élaboration de méthodes efficaces permettant d'avoir une plus grande précision au niveau de la reconnaissance grâce à l'élaboration de nouveaux algorithmes, à la réalisation de grands corpus appelés aussi big data, mais surtout grâce à la puissance des machines.

L'article rédigé en [13], représente un aperçu du contexte historique du Deep Learning. Il présente les grandes étapes qui mènent à la situation actuelle et qui sont résumées dans le tableau suivant.

<i>L'année</i>	<i>Contributeur</i>	<i>Contribution</i>
300 AC	Aristote	Introduction de l'associationnisme, début de l'histoire des humains qui essayent de comprendre le cerveau
1873	Alexander Bain	Introduction du Neural Groupings comme les premiers modèles de réseaux de neurones
1943	McCulloch et Pitts	Introduction du McCulloch Pitts (MCP) modèle considéré comme L'ancêtre des réseaux de neurones artificielles
1949	Donald Hebb	Considérer comme le père des réseaux de neurones, il introduit la règle d'apprentissage de Hebb qui servira de fondation pour les réseaux de neurones modernes.
1958	Frank Rosenblatt	Introduction du premier perceptron
1974	Paul Werbos	Introduction de la retro propagation
1980	Teuvo Kohonen	Introduction des cartes auto organisatrices
1980	Kunihiko Fukushima	Introduction du Neocognitron, qui a inspiré les réseaux de neurones convolutif
1982	John Hopfield	Introduction des réseaux de Hopfield
1985	Hilton et Sejnowski	Introduction des machines de Boltzmann
1986	Paul Smolensky	Introduction de Harmonium, qui sera connu plus tard comme machines de Boltzmann restreintes

1986	Michael I. Jordan	Définition et introduction des réseaux de neurones récurrent
1990	Yann LeCun	Introduction de LeNet et montra la capacités des réseaux de neurones profond
1997	Schuster et Paliwal	Introduction des réseaux de neurones récurrent bidirectionnelles
1997	Hochreiter et Schmidhuber	Introduction de LSTM, qui ont résolu le problème du vanishing gradient dans les réseaux de neurones récurrent
2006	Geoffrey Hinton	Introduction des Deep Belief Network
2009	Salakhutdinov et Hinton	Introduction des Deep Boltzmann Machines
2012	Geoffrey Hinton	Introduction de AlexNet qui remporta le challenge ImageNet

Tableau 2.1 : Résumé de l'histoire du Deep Learning [13].

2.3 Applications du Deep Learning

Parmi les applications du Deep Learning, on peut citer [14] :

- **La reconnaissance faciale** : Les yeux, le nez, la bouche, tout autant de caractéristiques qu'un algorithme de Deep Learning va apprendre à détecter sur une photo. Il va s'agir en premier lieu de donner un certain nombre d'images à l'algorithme, puis à force d'entraînement, l'algorithme va être en mesure de détecter un visage sur une image.
- **Recherche Vocale et Assistants à Commande Vocale** : L'un des domaines d'utilisation les plus populaires de l'apprentissage en profondeur est la recherche vocale et l'activation vocale assistants intelligents. Avec les grands géants de la technologie qui ont déjà fait des investissements importants dans ce domaine, des assistants à commande vocale peuvent être trouvés sur presque tous les

smartphones. Siri d'Apple est sur le marché depuis octobre 2011. Google Now, l'assistant à commande vocale pour Android, a été lancé moins d'un an après Siri. Le plus récent des assistants intelligents à commande vocale est Microsoft Cortana.

- **Reconnaissance d'image** : Un autre domaine populaire concernant l'apprentissage en profondeur est la reconnaissance d'images. Il vise à reconnaître et à identifier les personnes et les objets dans les images ainsi qu'à comprendre le contenu et le contexte. La reconnaissance d'image est déjà utilisée dans plusieurs secteurs comme les jeux, les médias sociaux, la vente au détail, le tourisme, etc.
- **la Détection du Cancer du Cerveau** : Une équipe de chercheurs français a noté qu'il est difficile de repérer les cellules cancéreuses invasives du cerveau pendant la chirurgie, en partie à cause des effets de l'éclairage dans les salles d'opération. Ils ont découvert que l'utilisation de réseaux de neurones en conjonction avec la spectroscopie Raman pendant les opérations leur permettait de détecter plus facilement les cellules cancéreuses et de réduire les résidus de cancer après l'opération.
- **Génération automatique de texte** : C'est une tâche intéressante, où un corpus de texte est appris et à partir de ce modèle, un nouveau texte est généré, mot par mot ou caractère par caractère. Le modèle est capable d'apprendre à épeler, ponctuer, former des phrases et même capturer le style du texte dans le corpus. De grands réseaux de neurones récurrents sont utilisés pour apprendre la relation entre les éléments dans les séquences de chaînes d'entrée, puis générer du texte.

2.4 Le réseau neuronal

Dans la pratique, tous les algorithmes DL sont des réseaux neuronaux [15], Les réseaux neuronaux, aussi appelés ANN, sont des modèles de traitement de l'information qui simulent le fonctionnement d'un système nerveux biologique. C'est similaire à la façon dont le cerveau manipule l'information au niveau du fonctionnement. Tous les réseaux neuronaux sont constitués de neurones inter connectés qui sont organisés en couches [15].

2.4.1 Le neurone

Ce qui forme les réseaux de neurones, ce sont les neurones artificiels inspirés du vrai neurone qui existe dans notre cerveau. La figure suivante montre une représentation d'un neurone réel (biologique) et d'un neurone artificiel :

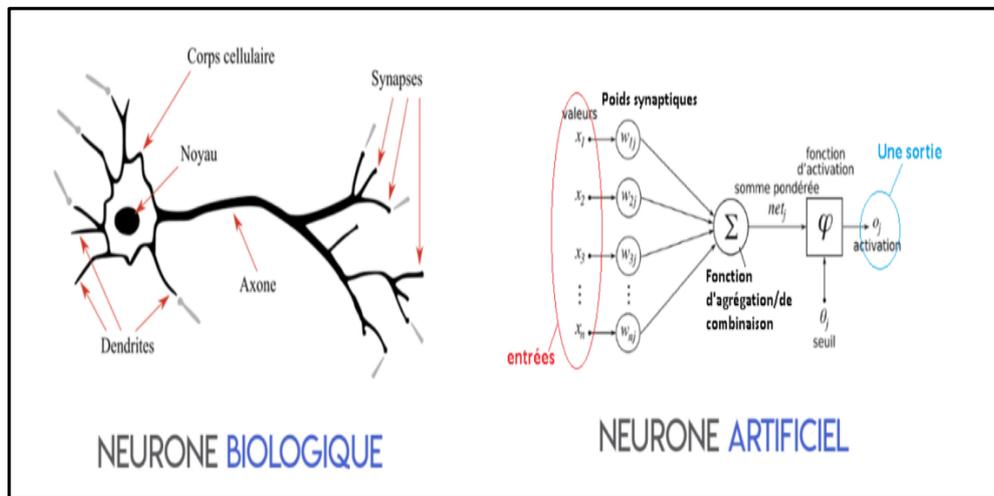


Figure 2.2 : Représentation d'un neurone biologique et d'un neurone artificiel [16].

Où les x_i sont des valeurs numériques qui représentent soit les données d'entrée, soit les valeurs sorties d'autres neurones. Et où les poids w_i sont des valeurs numériques qui représentent soit la valeur de puissance des entrées, soit la valeur de puissance des connexions entre les neurones. Il existe des opérations qui se passent au niveau du neurone artificiel.

Le neurone artificiel fera un produit entre le poids (w) et la valeur d'entrée (x), puis ajouter un biais (b), le résultat est transmis à une fonction d'activation (f) qui ajoutera une certaine non-linéarité (Figure 2.3).

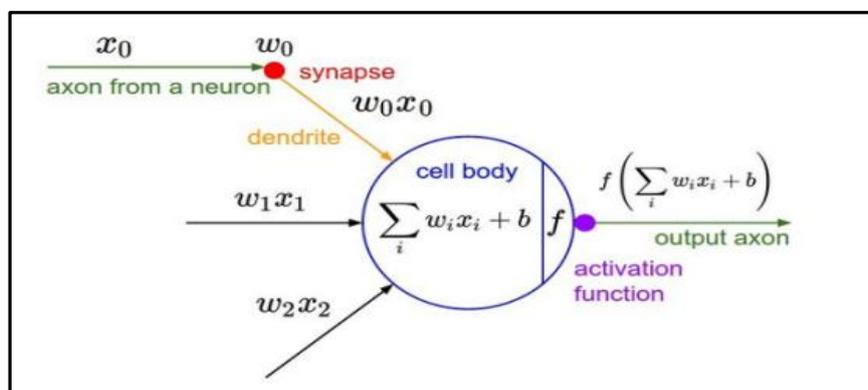


Figure 2.3: Neurone artificiel [17].

2.4.2 Fonctions d'activation

Après que le neurone a effectué le produit entre ses entrées et ses poids, il applique également une non-linéarité sur ce résultat. Cette fonction non linéaire s'appelle la fonction d'activation.

La fonction d'activation est une composante essentielle du réseau neuronal. Ce que cette fonction a décidé est si le neurone est activé ou non. Elle calcule la somme pondérée des entrées et ajoute le biais. C'est une transformation non linéaire de la valeur d'entrée.

Après la transformation, cette sortie est envoyée à la couche suivante. La non-linéarité est si importante dans les réseaux de neurones, sans la fonction d'activation, un réseau de neurones est devenu simplement un modèle linéaire. Il existe de nombreux types de ces fonctions, parmi lesquelles nous trouvons [18] [19] :

a) Fonction Sigmoid

Cette fonction est l'une des plus couramment utilisées. Elle est bornée entre 0 et 1, et peut être interprétée stochastiquement comme la probabilité que le neurone s'active (Figure 2.4). Elle est généralement appelée la fonction logistique ou le sigmoïde logistique.

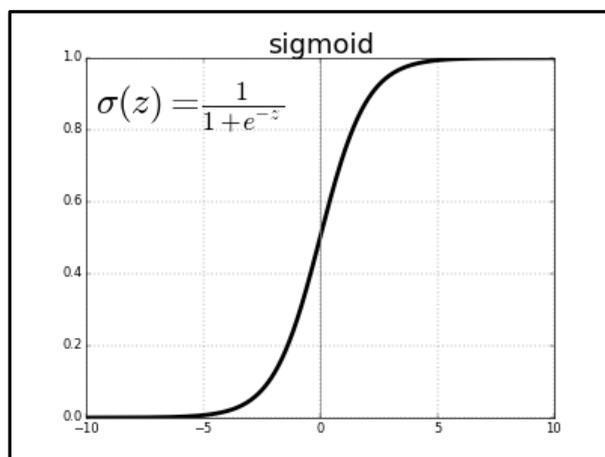


Figure 2.4: Représentation graphique de la fonction Sigmoid[20].

b) Fonction ReLu

La fonction ReLu est probablement la plus proche de sa correspondante biologique [19]. Cette fonction est récemment devenue le choix de nombreuses tâches (notamment en computer vision) [18]. Cette fonction renvoie 0 si l'entrée z est inférieure à 0 et retourne z lui-même si il est plus grand que 0 (Figure 2.5).

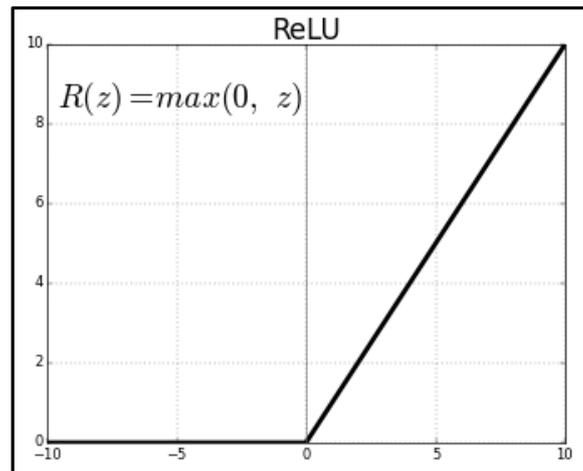


Figure 2.5: Représentation Graphique de la fonction ReLu [21].

c) Fonction Softmax

Régression Softmax (synonymes : Logistique multinomiale, Classificateur d'entropie maximale, ou simplement Régression logistique multi-classe) est une généralisation de la régression logistique que nous pouvons utiliser pour la classification multi-classe [22].

Contrairement à d'autres types de fonctions, la sortie d'un neurone d'une couche utilisant la fonction softmax dépend des sorties de tous les autres neurones de sa couche. Cela s'explique par le fait qu'elle nécessite que la somme de toutes les sorties soit égale à 1. Le vecteur de sortie souhaité est de la forme ci-dessous (Figure 2.6) :

$$[p_0, p_1, p_2, \dots, p_n] \text{ où } \sum_{i=0}^n p_i = 1$$

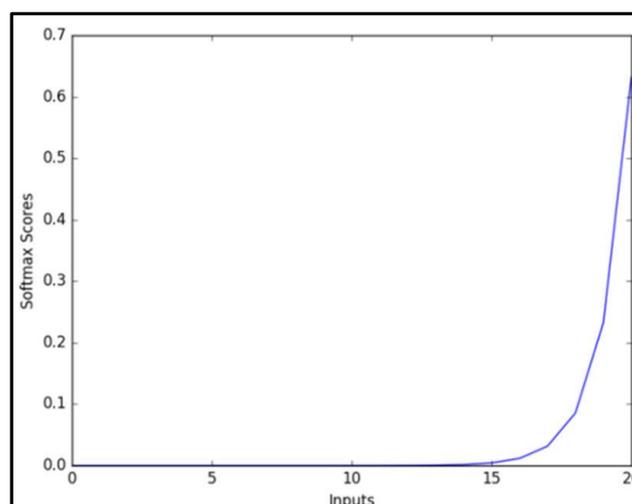


Figure 2.6 : Représentation graphique de la fonction Softmax [23].

2.4.3 Les types des réseaux de neurones

Il existe plusieurs types de réseaux neuronaux :

a) Perceptron multicouche (Multi Layer Perceptron : MLP)

Le perceptron multicouche (multilayer perceptron MLP) est un type de réseau neuronal artificiel organisé en plusieurs couches (Figure 2.7) : couche d'entrée, couche cachée et couche de sortie.

- **La couche d'entrée (input layer)** : composée d'un ensemble de neurones qui portent le signal d'entrée. Tous les neurones de cette couche sont ensuite reliés à ceux de la couche suivante.
- **La couche cachée (hidden layer)** : Ou plus souvent les couches cachés (couche cachée 1, couche cachée 2, . . .). Il s'agit du cœur du perceptron.
- **La couche de sortie (output layer)** : qui représente le résultat final du réseau, c'est-à-dire sa prédiction.

Mis à part les nœuds d'entrée, chaque nœud est un neurone qui utilise une fonction d'activation non linéaire. Le MLP utilise une technique d'apprentissage supervisé appelée rétro-propagation pour la formation. Ses multiples couches et son activation non linéaire distinguent le MLP d'un perceptron linéaire. Il peut distinguer des données qui ne sont pas linéairement séparables.

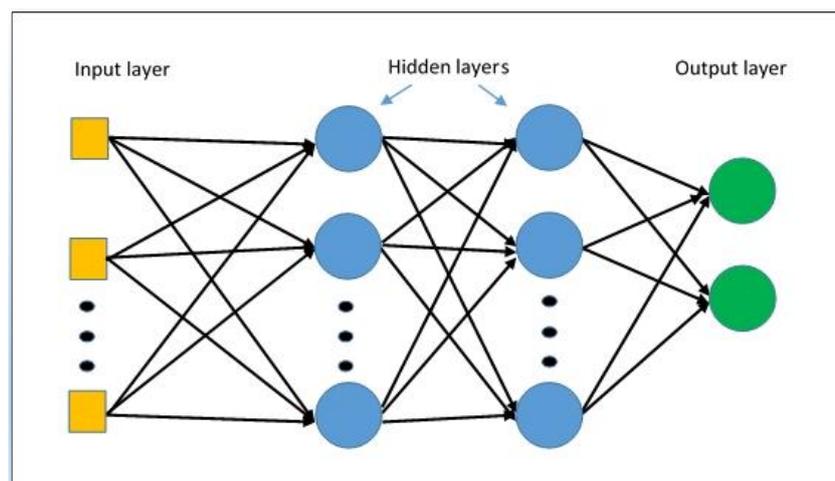


Figure 2.7 : Perceptron Multicouche [24].

b) Réseaux de neurones convolutifs

Les réseaux de neurones convolutifs (CNN : Convolutional Neural Networks) supposent une structure spatiale particulière dans leurs entrées. En particulier, ils supposent que les entrées qui sont proches les unes des autres dans l'entrée originale sont sémantiquement liées [25].

Autrement dit, le CNN est une séquence de couches, où chaque couche transforme un volume d'activations en un autre par une fonction différentiable [26].

Les trois principaux types de couches (Figure 2.8) pour construire ce type de réseau sont : la couche convolutive, la couche de pooling et la couche entièrement connectée [26].

- **Couche convolutive** : C'est la couche la plus importante et le cœur des éléments constitutifs du réseau convolutif, et c'est aussi elle qui effectue le plus de calculs lourds.
- **Couche de pooling** : Il est courant d'insérer périodiquement une couche de Pooling dans ce type d'architecture. Sa fonction est de réduire progressivement la taille spatiale de la représentation pour réduire le nombre de paramètres et de calculs dans le réseau, et donc de contrôler également l'overfitting.
- **Couche entièrement connectée** : Comme nous l'avons mentionné précédemment, les neurones d'une couche entièrement connectée ont des connexions complètes à toutes les activations de la couche précédente.

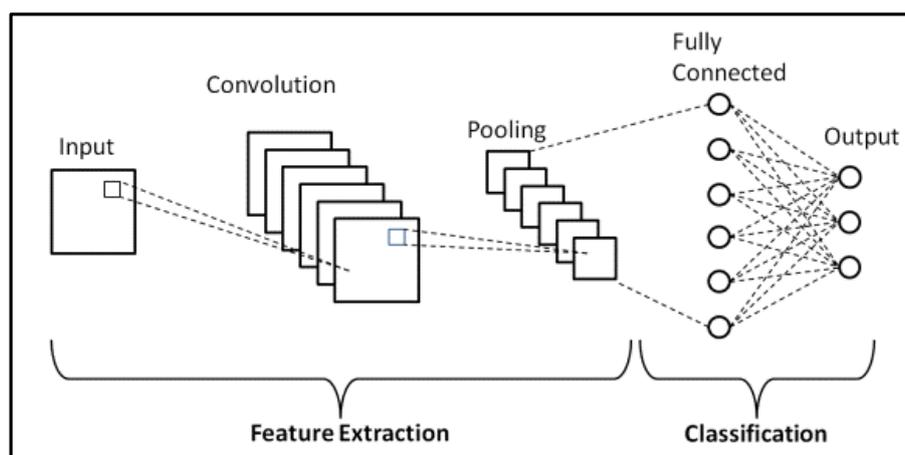


Figure 2.8 : Architecture du Modèle CNN [27].

c) Réseaux neuronaux récurrents

Les réseaux de neurones récurrents (RNN : Recurrent Neural Networks) est un type de réseau neuronal. Les couches de réseaux neuronaux récurrentes sont des entités primitives qui permettent aux réseaux neuronaux d'apprendre à partir de séquences d'entrées [25]. La figure ci-dessous représente les types de séquences d'entrée que le réseau traite.

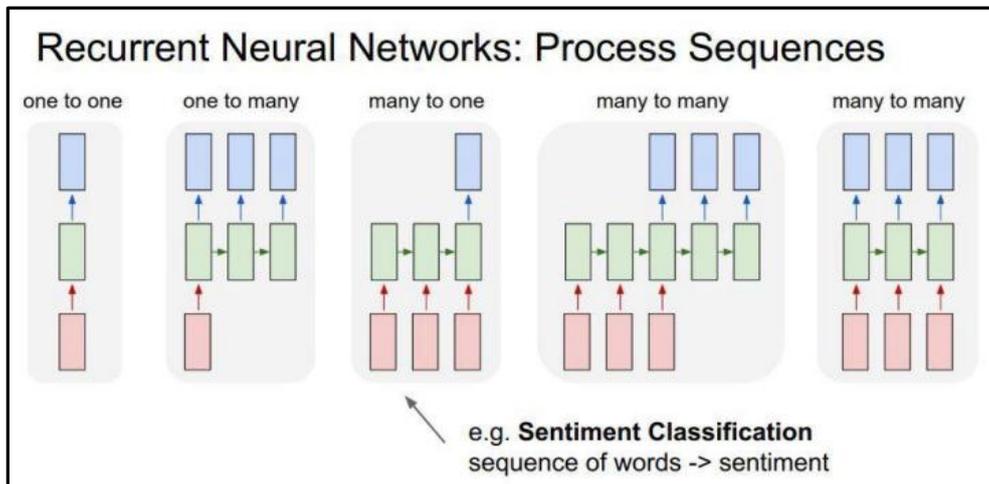


Figure 2.9: Types de séquences d'entrée pour un réseau récurrent [25].

Si la séquence que nous traitons est une phrase de 3 termes par exemple, le réseau va être déroulé en un réseau neuronal à 3 couches, une couche pour chaque mot. La figure suivante représente cette idée [25] :

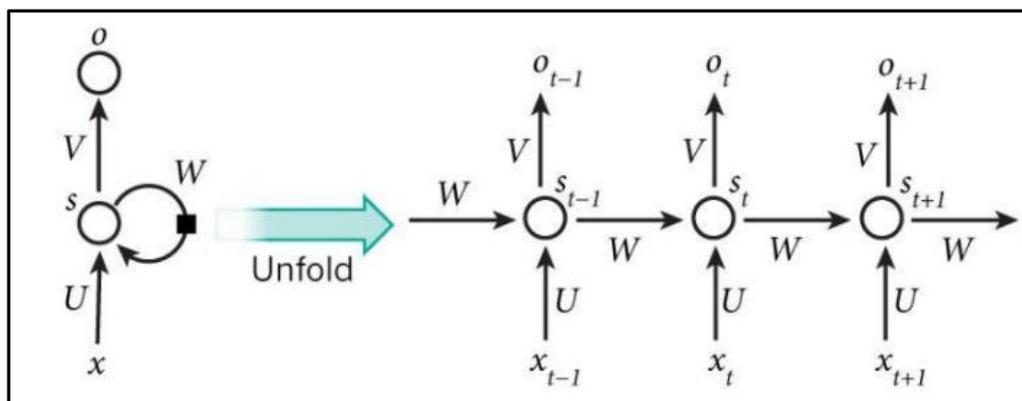


Figure 2.10 : Exemple d'un réseau récurrent qui se déroule [28].

Les formules qui dirigent les calculs dans un RNN sont les suivantes :

- x_t est l'entrée au temps t .
- s_t est l'état caché à l'instant t , qui est calculé en fonction de l'état caché précédent et de l'entrée à l'étape courante avec : $s_t = f(Ux_t + W s_{t-1})$.

- U, V, W sont les paramètres que le réseau va apprendre des données de l'apprentissage.
- Où f est une fonction non linéaire telle que : ReLu ou Hyperbolic tangent (\tanh).
- O_t est la sortie au moment t . Par exemple, si on veut prédire le prochain mot dans une phrase, ce serait un vecteur de probabilités dans un vocabulaire :
$$O_t = \text{softmax}(V s_t)$$

Les RNN ont des difficultés à apprendre des dépendances sur le long terme et les interactions entre des mots qui sont éloignés les uns des autres [18]. Cela pose problème, car la signification d'une phrase est souvent déterminée par des mots qui ne sont pas très similaires.

Dans ce cas si la fonction d'activation est une \tanh ou une sigmoïde alors on aura le problème de disparition du gradient. C'est pour cette raison que nous avons recours à la fonction d'activation ReLu qui ne souffre pas de ce problème, mais il existe une solution encore plus utilisée c'est l'utilisation des architectures Long Short term Memory (LSTM) ou Gated Recurrent Unit (GRU).

➤ RNN LSTM (Long Short-Term Memory) :

Les réseaux à mémoire longue à court terme généralement appelés LSTM (Long Short Term Memory) sont un type spécial de RNN. Ils ont été introduits par Hochreiter Schmidhuber en 1997.

Les Réseaux neuronaux récurrents présentés dans la section précédente sont capables d'apprendre des règles de mise à jour de séquence arbitraire en théorie. Dans la pratique, cependant, ces modèles oublient rapidement le passé [25]. C'est ce qu'on appelle le problème de la disparition du gradient qui a poussé à inventer le LSTM.

La cellule LSTM est une adaptation de la couche récurrente qui permet aux signaux plus anciens des couches profondes de se déplacer vers la cellule du présent [25]. La figure suivante représente une chaîne de trois cellules LSTM :

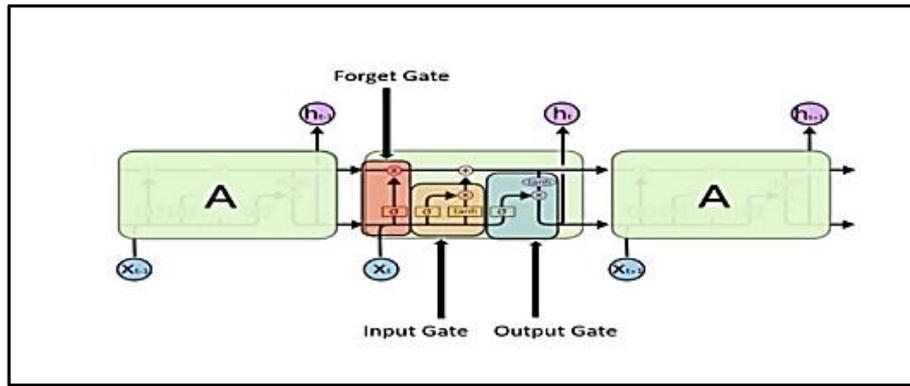


Figure 2.11 : Chaîne de Cellules LSTM [29].

Et où les calculs se déroulent comme suit :

- **Forget gate (Porte d’oubli)** : Cette porte décide quelles informations doivent être conservées ou jetées (Figure 2.12). Cette décision est faite par la fonction sigmoïde : elle regarde l’état précédent (h_{t-1}) et l’entrée (X_t), si la sortie de la sigmoïde est proche de 0, cela signifie que l’on doit oublier l’information et si on est proche de 1 alors il faut la mémoriser pour la suite.

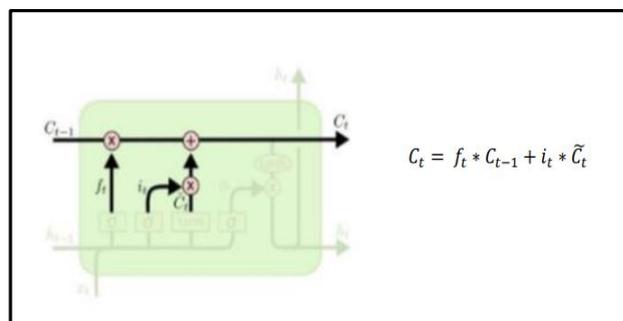


Figure 2.12 : Porte d’oubli [28].

Avec :

tanh : Fonction d’activation tangente hyperbolique.

C_t : Valeur candidate.

C_t : État interne

- **Input gate (Porte d’entrée)** : Cette porte décide si l’entrée doit modifier le contenu de la cellule (Figure 2.13). La fonction sigmoïde décide des valeurs à laisser passer par 0 et 1, et la fonction tanh donne un poids aux valeurs qui sont passées en décidant de leur niveau d’importance allant de -1 à 1.

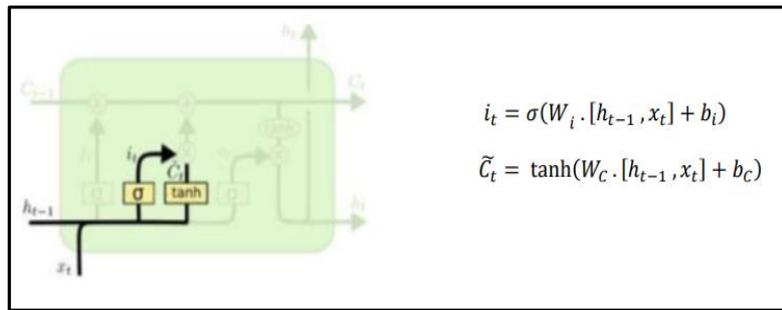


Figure 2.13 : Porte d'entrée [28].

Avec :

h_{t-1} : Sortie à l'instant $t - 1$.

x_t : Entrée courante à l'instant t .

b : Bais.

w : Poids.

σ : Fonction sigmoïde.

- **Output gate (porte de sortie)** : Cette porte détermine si une cellule peut envoyer une valeur aux autres cellules ou non (Figure 2.14). En effet, quand sa valeur est proche de zéro, la cellule ne fournit aucune information en sortie. Cette porte décide quelle partie de la cellule actuelle parvient à la sortie.

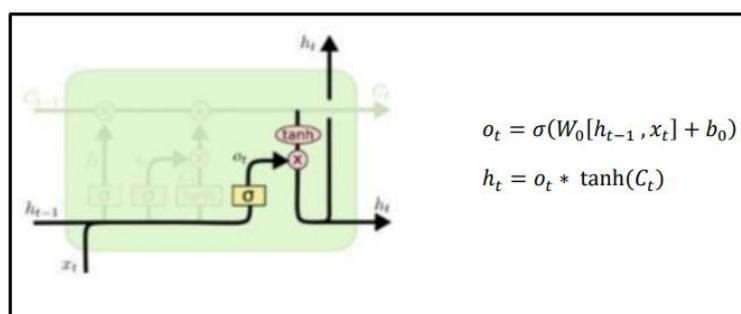


Figure 2.14 : La porte de sortie [28].

Avec :

h_t : Sortie.

Les réseaux neuronaux dont on parle, ne sont pas tous des réseaux qui existent, une excellente ressource qui résume peut-être toutes les architectures est en [30]. La figure suivante illustre leur œuvre.

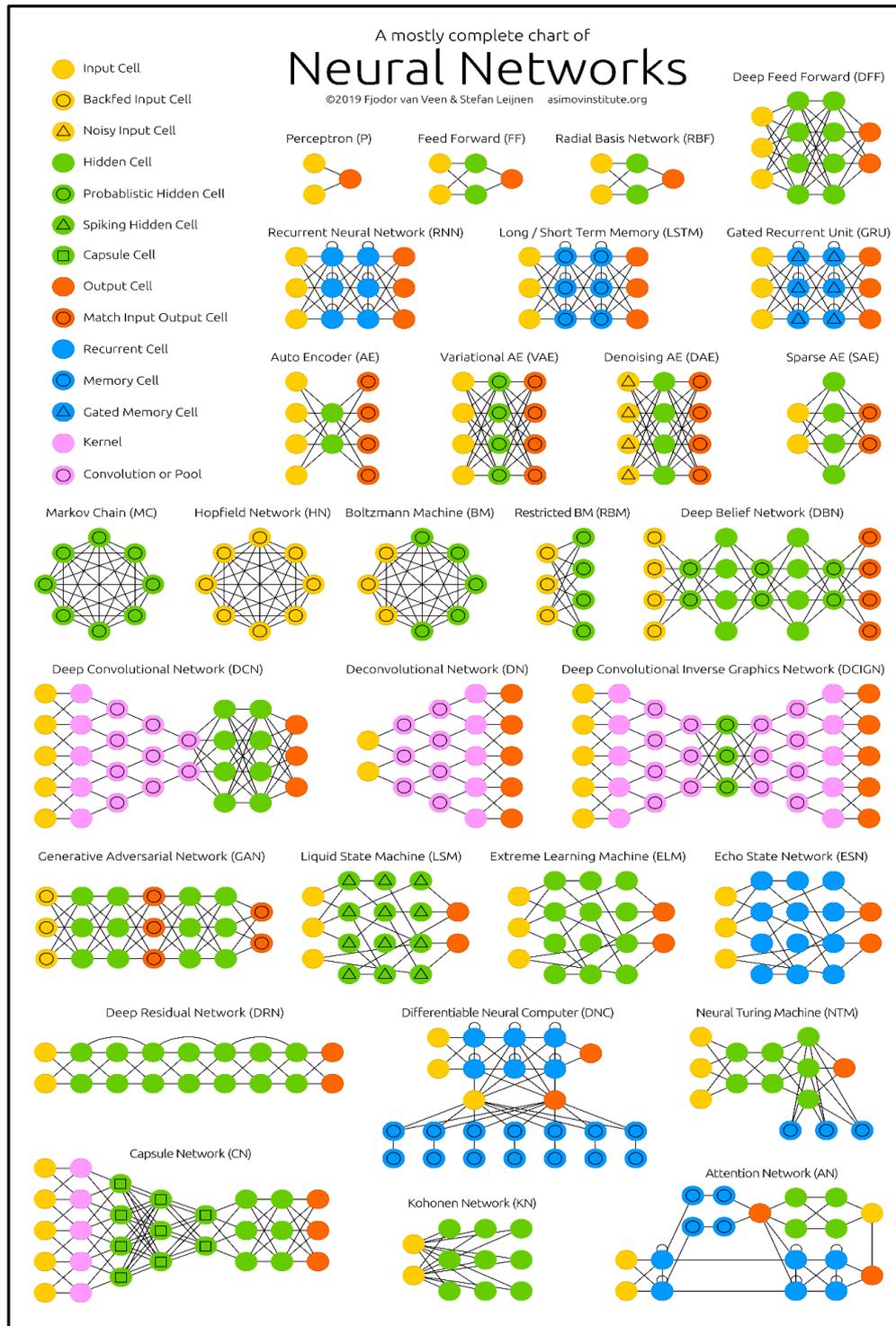


Figure 2.15: Résumé des types d'architectures de réseaux de neurones [30].

2.5 Apprentissage en Deep Learning

Le processus d'apprentissage dans le DL revient à l'entraînement du réseau neuronal en utilisant des optimiseurs itératifs, qui ne font que conduire la fonction de coût à une très faible valeur [31]. Nous pouvons utiliser différents algorithmes pour effectuer l'apprentissage, mais l'algorithme le plus utilisé est l'algorithme itératif d'optimisation par la descente du gradient qui est la méthode la plus utilisée presque sur tous les réseaux neuronaux avec ses différents modèles.

Le processus d'apprentissage revient au problème d'optimisation où il s'agit de minimiser ou de maximiser une fonction $f(x)$. Cette fonction est appelée la fonction objective, les auteurs dans [31] l'ont appelée aussi la fonction de coût, la fonction de perte et la fonction d'erreur.

Pendant l'entraînement, l'algorithme tente d'identifier le minimum global sans tomber dans le piège du minimum local. Le schéma suivant réalisé par les auteurs en [31] est une illustration de ce concept :

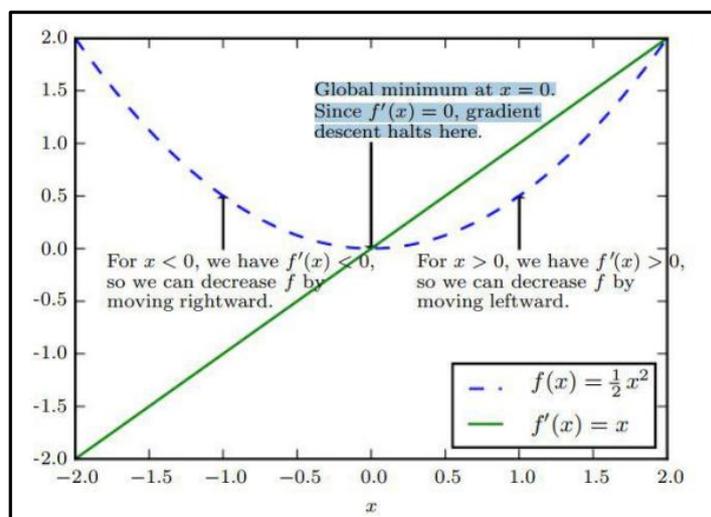


Figure 2.16 : Illustration du processus de recherche de l'optimum [31].

2.5.1 Variantes de la descente du gradient

Il existe trois principaux types de variantes de l'algorithme de descente du gradient. La principale différence entre eux est la quantité de données que nous utilisons lorsque nous calculons le gradient pour chaque étape d'apprentissage [16] [18] [31]. Les algorithmes d'optimisation qui utilisent tout l'ensemble de l'apprentissage sont appelés les méthodes de gradients déterministes ou batch descente de gradient, car ce sont des méthodes où tous les

exemples d'apprentissage sont traités simultanément dans un grand batch, tandis que le terme "batch" est pour décrire un groupe d'exemples.

Les algorithmes d'optimisation qui n'utilisent qu'un seul exemple à la fois sont parfois appelés méthodes stochastiques ou en online descente du gradient.

La plupart des algorithmes utilisés pour l'apprentissage profond se situent quelque part entre les deux, utilisant plus d'un mais moins que tous les exemples d'entraînement. Ils sont appelés les méthodes stochastiques de minibatch ou de minibatch descente de gradient.

2.5.2 Algorithmes d'optimisation de la descente du gradient

Adam est un algorithme d'optimisation présenté en 2015 [32]. Le nom de cet algorithme est dérivé d'Adaptive Moment Estimation. Lors de l'introduction de cet algorithme, les auteurs ont présenté les avantages de l'utilisation d'Adam sur des problèmes d'optimisation non convexes, comme suit :

- Simplicité de mise en œuvre.
- Efficacité du calcul.
- Peu de mémoire requise.
- Bien adapté aux problèmes volumineux en termes de données et / ou de paramètres.
- Les hyper-paramètres nécessitent généralement peu de réglages.

Il existe d'autres optimiseurs avec différents mécanismes de fonctionnement, comme :

- Adagrad.
- RMSProp.
- Adadelta.

2.6 Conclusion

Nous avons présenté au cours de ce chapitre l'apprentissage profond, ses applications, les réseaux de neurones où on a parlé sur le neurone biologique et formel, les fonctions d'activation et on a cité quelques types des réseaux de neurones. À la fin de ce chapitre, nous avons parlé de l'apprentissage en Deep Learning.

Chapitre 3 : État de l'art



3.1 Introduction

Dans le but de bien mener et aborder notre problématique, on a commencé par chercher et lire plusieurs articles publiés qui traitent des problématiques sur la classification des signaux ECG avec la méthode de Deep Learning.

3.2 Approche 1 : ECG arrhythmia classification using a 2-D convolutional neural network

Dans cette approche, nous allons simplement expliquer la méthode utilisée et présenter leurs résultats.

- **Explication de la méthode utilisée :**

Dans le but de faire la classification des signaux ECG d'une façon automatique avec les outils de Deep Learning, dans [33], ils ont utilisé un modèle de DL qui est le CNN comprenant deux étapes : le prétraitement des données ECG et le classificateur des arythmies ECG.

Dans cet article, ils ont utilisé la base de données sur l'arythmie MIT-BIH [34] pour la formation et les tests du modèle CNN. Etant donné que le modèle CNN gère l'image bidimensionnelle en tant que données d'entrée, les signaux ECG sont transformés en images ECG pendant l'étape de prétraitement des données ECG. Avec ces images obtenues, la classification de huit types d'ECG est effectuée à l'étape du classificateur CNN. Les procédures globales sont illustrées à la figure 3.1.

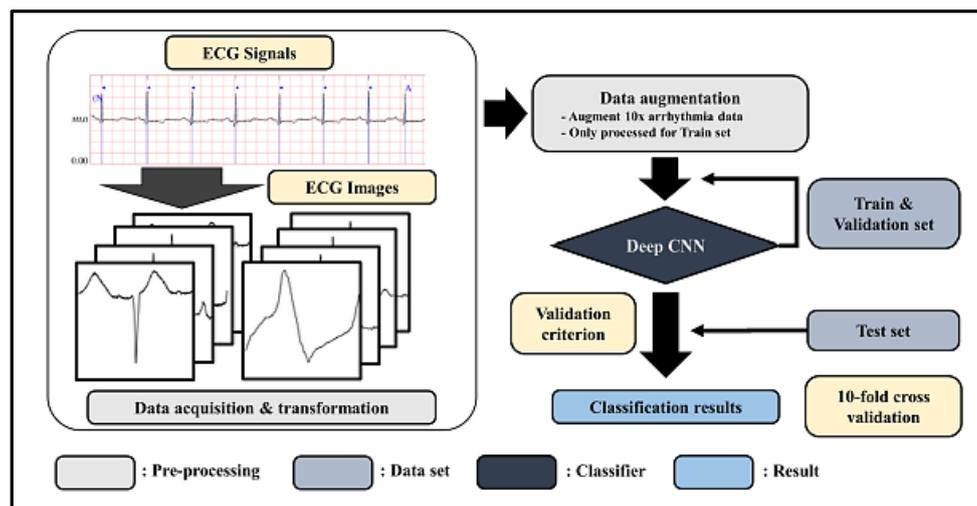


Figure 3.1: Procédures globales traitées dans la classification des arythmies ECG[33].

➤ **Étape 1 : Prétraitement des données ECG**

CNN bidimensionnel nécessite une image comme donnée d'entrée. Par conséquent, ils ont transformé les signaux ECG en images ECG en traçant chaque battement ECG sous la forme d'une image individuelle en niveaux de gris de 128 x 128. Dans la base de données sur l'arythmie MIT-BIH, chaque battement d'ECG est découpé en tranches en fonction de l'heure de pointe de l'onde Q. Plus précisément, le type d'arythmie est marqué à l'heure de pointe de l'onde Q de chaque battement ECG. Ainsi, ils ont défini une seule image de battement ECG en centrant le signal de crête d'onde Q tout en excluant le premier et les 20 derniers signaux ECG des signaux de crête d'onde Q précédents et suivants.

➤ **Étape 2 : Classificateur d'arythmies ECG**

Dans cet article, ils ont adopté le CNN comme classificateur d'arythmie ECG. La raison pour laquelle ils ont appliqué le CNN 2D en convertissant le signal ECG en forme d'image ECG. Les couches de convolution et de regroupement 2D sont plus appropriées pour filtrer la localité spatiale des images ECG.

Il existe plusieurs modèles CNN réussis du ImageNet Large Visual Perception Challenge (ILSVRC)[35], un concours pour détecter et classer des objets dans un ensemble donné d'images.

AlexNet [36], annoncé en 2012, a pris la première place avec des performances écrasantes en tant que premier modèle à utiliser l'arythmie ECG et le GPU du modèle CNN à ILSVRC.

En 2014, Google Net [37] et VGGNet [38] sont respectivement les premier et deuxième de l'ILSVRC. Bien que VGGNet occupe la deuxième place, la structure VGGNet avec une structure répétitive de filtres 3x3 et un sous-échantillonnage est plus souvent utilisé dans la reconnaissance d'images car la structure est beaucoup plus simple et les performances ne sont pas très différentes de Google Net.

ResNet [39] et Dense Net [40], apparus récemment, ont été proposés pour résoudre le problème selon lequel l'effet des premières caractéristiques de l'image sur la sortie finale est faible lorsque la profondeur du modèle CNN est plus profonde.

Dans cet article, Ils ont suivi la structure de base de VGGNet et optimisé le modèle CNN pour montrer des performances optimales pour la classification des arythmies ECG.

La comparaison des performances du modèle CNN proposé a été effectuée avec AlexNet et VGGNet sans comparaison avec GoogLeNet, ResNet et Réseau dense. En effet, l'image ECG de cet article est une image en niveaux de gris 128 x 128 relativement simple, il n'est donc pas nécessaire d'avoir une couche de profondeur profonde, et une augmentation des paramètres libres pourrait entraîner un sur ajustement et une dégradation des performances. La Figure 3.2 montre l'architecture globale de modèle CNN proposé.

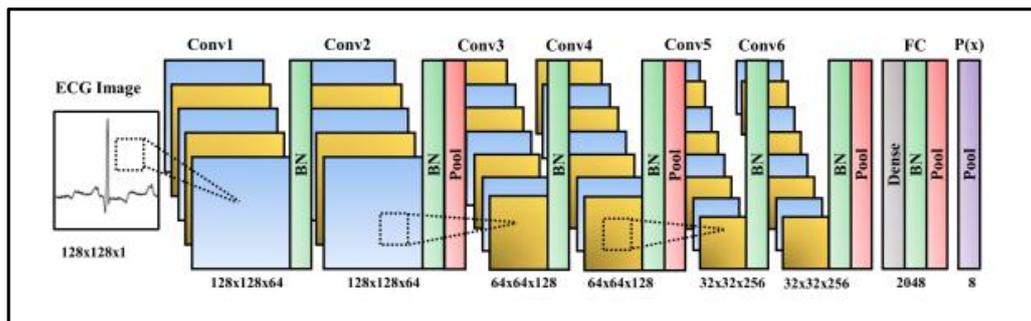


Figure 3.2: L'architecture globale de modèle CNN [33].

Data augmentation :

L'augmentation des données signifie l'augmentation du nombre de points de données. En termes d'images, cela peut signifier l'augmentation du nombre d'images dans le jeu de données. Nous pouvons atteindre à la fois une spécificité et une sensibilité élevées en augmentant et en équilibrant les données d'entrée. On augmente sept battements d'arythmie ECG (PVC, PAB, RBB, LBB, APC, VFW, VEB) avec neuf méthodes de recadrage différentes : gauche haut, centre haut, droite haut, centre gauche, centre, centre droite, gauche bas, centre bas et en bas à droite. Chaque méthode de recadrage donne deux des trois tailles d'une image ECG, c'est-à-dire 96 x 96. Ensuite, ces images augmentées sont redimensionnées à la taille originale de 128 x 128.

▪ Résultats obtenus :

Les résultats d'évaluation résumés des modèles CNN sont présentés dans la figure suivante où Natif fait référence aux données d'entraînement sans l'augmentation et Augmenté fait référence à l'augmentation. Le modèle CNN proposé a atteint une AUC de 0,989, une moyenne de 99,05%, une précision, une spécificité de 99,57%, une sensibilité moyenne de 97,85% et une valeur prédictive positive de 98,55%.

Method	Grade	AUC	Acc(%)	Sp(%)	Se(%)	+P(%)
Proposed	Native	0.986	98.90	99.64	97.20	98.63
	Augmented	0.989	99.05	99.57	97.85	98.55
AlexNet	Native	0.985	98.81	99.68	96.81	98.63
	Augmented	0.986	98.85	99.62	97.08	98.59
VGGNet	Native	0.984	98.77	99.43	97.26	98.08
	Augmented	0.984	98.63	99.37	96.93	97.86

Figure 3.3: Les résultats des différents modèles CNN [33].

À partir de la figure 3.3, le modèle CNN proposé avec augmentation des données montre les meilleurs résultats d' AUC, de précision et de sensibilité tout en AlexNet sans augmentation présente la meilleure spécificité et valeur prédictive positive. La discrimination de la sensibilité entre la méthode proposée et la deuxième méthode la plus élevée est de 0,59 % tandis que la discrimination de spécificité de la méthode proposée et de la méthode la plus élevée est de 0,11 %. Cela indique que le modèle proposé a obtenu la meilleure précision dans la classification de l'arythmie.

Le VGGNet montre le pire résultat par rapport aux deux autres modèles CNN. La principale différence entre le VGGNet et les autres modèles est le nombre de couches de regroupement. Après avoir contourné quatre couches de regroupement, la taille de l'image ECG est réduite à 8 x 8. Bien que le VGGNet ait deux fois plus de nombre de noyaux à la dernière couche convolutive, il semble que la dernière taille d'entité avec 16 x 16 soit plus stabilisée. Par conséquent, lorsque on utilise la taille d'image ECG avec 128 x 128, il est recommandé de ne pas utiliser plus de trois couches de regroupement dans le modèle CNN. Entre le modèle proposé et l'AlexNet avec augmentation des données, la différence entre les sensibilités moyennes est de 0,77 %. C'est une différence assez importante lorsque nous examinons les sensibilités détaillées des sept arythmies différentes.

3.3 Approche 2: Classification of ECG arrhythmias using recurrent neural networks

L'objectif de cet article [41] est d'appliquer les réseaux de neurones récurrents pour faire la classification des arythmies des signaux ECG. Le signal ECG se compose de seize types de battements cardiaques qui sont divisés en deux groupes de battements cardiaques normaux et d'arythmie.

La base de données sur l'arythmie MIT-BIH est utilisée pour examiner les meilleures performances de classification. À des fins de formation et de test, l'un des ensembles de données les plus populaires sur l'arythmie MIT-BIH a donc été utilisé. Il contient 47 enregistrements de 30 minutes chacun et 40 % des enregistrements étaient ceux de patients cardiaques. Les enregistrements contenaient différents types de signaux basés sur différents emplacements de fils : L2, V1, V2, V4 et V5 [42].

Tout d'abord, un fractionnement de 70 à 30 des données a été effectué. Les signaux ML2 ont été sélectionnés et pour chaque enregistrement, le signal a été segmenté en morceaux de longueur 720-720. Étant donné que la fréquence d'échantillonnage était de 360, 360 segments avant la valeur de crête R et 360 segments après celle-ci garantissaient que près de 3 battements seraient présents dans un morceau et la classification du battement central devait être effectuée par rapport à 15 étiquettes arythmiques possibles. L'architecture globale du système monté dans la Figure 3.4.

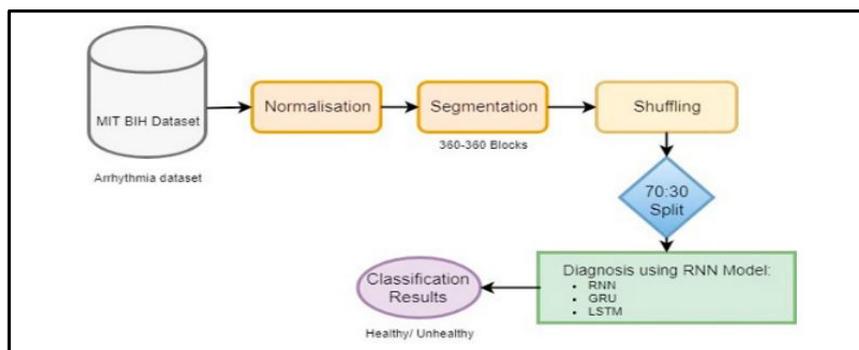


Figure 3.4 : Les étapes de détection d'arythmie ECG proposées[41].

▪ Résultats obtenus :

Le RNN LSTM a montré une précision de 88,1% lorsque on prend le nombre d'itérations égal à 5 et les couches cachées à 3 et il y a respectivement 64, 256 et 100 neurones par couche cachée, ce qui montre une meilleure détection de l'arythmie que RNN et GRU car la précision de RNN est de 85,4% et GRU est de 82,5%, ce qui est inférieur à la précision LSTM. Le modèle est implémenté directement en utilisant les signaux de la base de données MIT BIH et aucun prétraitement n'a été effectué. Par conséquent, la complexité du modèle implémenté est bien moindre que celle des algorithmes d'apprentissage automatique traditionnels.

Dans cet article, la classification binaire de l'arythmie a été réalisée et les résultats peuvent être améliorés en l'étendant à une classification multi classe. Depuis peu de travaux ont été faits dans ce domaine de classification binaire (détection de l'arythmie), le modèle

proposé permet la même chose et offre la possibilité de poursuivre les travaux dans ce domaine. La précision de la classification peut être augmentée plus loin en augmentant le nombre d'époques. L'article montre que le LSTM donne le meilleur résultat pour la classification binaire de l'arythmie ECG (Figure 3.5).

	Algorithm	Accuracy	Sensitivity	Specificity
ECG Classification	RNN	85.4	80.6	85.7
ECG Classification	RNN GRU	82.5	78.9	81.5
LSTM ECG Classification	RNN LSTM	88.1	92.4	83.35

Figure 3.5 : Les résultats des différents modèles RNN[41].

3.4 Approche 3 : Application of Deep convolutional neural network for automated detection of myocardial infarction using ECG signals

Dans cet article [43], ils proposent une nouvelle approche pour détecter automatiquement l'IM à l'aide de signaux ECG. Dans cette étude, ils ont mis en œuvre un algorithme de réseau neuronal convolutif (CNN) pour la détection automatisée d'un battement ECG normal et MI (avec bruit et sans bruit).

Ils ont obtenu une précision moyenne de 93,53% et 95,22% en utilisant des battements ECG avec bruit et sans suppression du bruit respectivement. De plus, aucune extraction ou sélection de caractéristiques n'est effectuée dans ce travail. Par conséquent, leur algorithme proposé peut détecter avec précision les signaux ECG inconnus même avec du bruit. Ainsi, ce système peut être introduit en milieu clinique pour aider les cliniciens dans le diagnostic de MI.

Dans ce travail, les signaux ECG ont été obtenus à partir de la base de données ECG (Physikalisch-Technische Bundesanstalt diagnostic ECG data base) [44]. Cette base de données fournit des données ECG de 200 sujets (148 MI et 52 sujets sains). De plus, 12 signaux de dérivation ont été enregistrés à partir de chaque sujet. La Figure 3.6 présente les caractéristiques des données ECG obtenues à partir de la base de données PTB.

	Normal	MI
Minimum age	17	36
Maximum age	81	86
Average age	43.43	60.37
Number of male	39	110
Number of female	13	38

Figure 3.6 : Les caractéristiques des données ECG obtenues à partir de la base de données PTB[43].

Dans ce travail, ils valident leur méthode proposée avec deux ensembles de données ECG. Les deux ensembles de données se composent du même nombre de battements ECG. Cependant, dans l'un des ensembles de données, ils ont débruités et supprimé l'errance de base du signal ECG en utilisant des fonction d'ondelettes mères de l'ondelette 6 de Daubechies [45]. Mais, dans l'autre ensemble de données, ils ont conservé les bruits présents dans les signaux ECG. Ensuite, ils ont effectué la détection du pic R sur les deux ensembles de données (avec et sans bruit) en utilisant l'algorithme Pan Tompkins [46].

L'architecture standard d'un CNN se compose de quatre étapes (i) Convolution, (ii) Fonction d'activation linéaire rectifiée, (iii) Fonction de mise en commun et (iv) Couche entièrement connectée.

La Figure 3.7 montre une représentation graphique de l'architecture de leur système proposé.

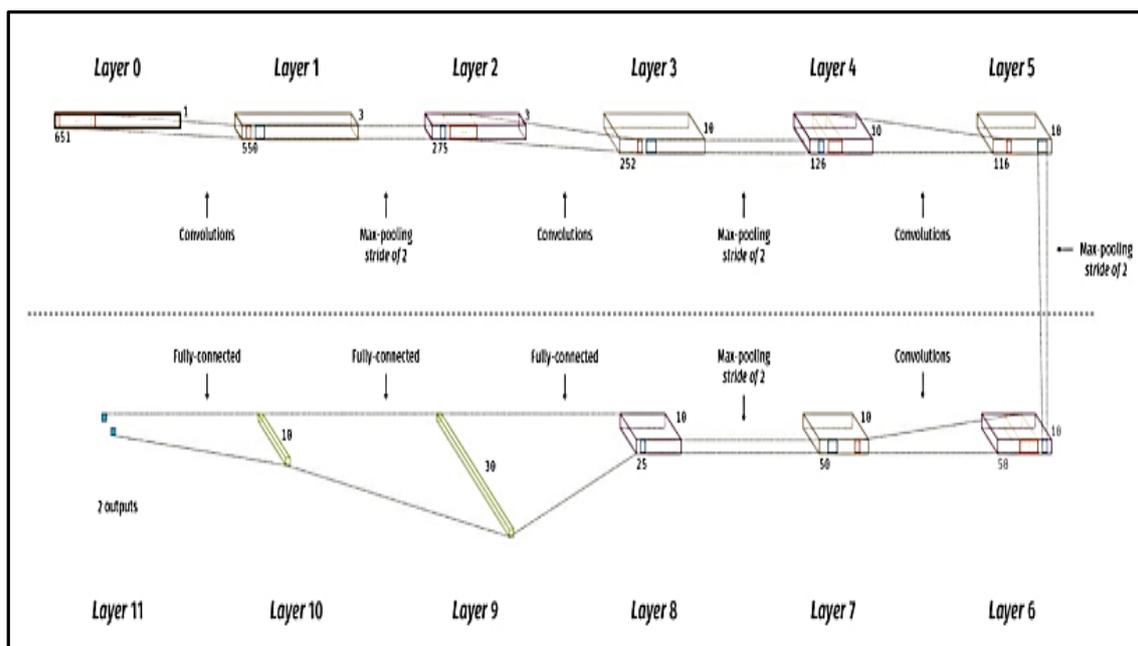


Figure 3.7: L'architecture du CNN proposée [43].

▪ Résultats obtenus :

Table 3 Confusion matrix of ECG beats with noise across 10-folds.							
		Predicted		ACC (%)	PPV (%)	SEN (%)	SPEC (%)
		Normal	MI				
Original	Normal	9790	756	93.53	79.48	92.83	93.71
	MI	2527	37,655	93.53	98.03	93.71	92.83

ACC = Accuracy, PPV = Positive Predictive Value, SEN = Sensitivity, SPEC = Specificity

Table 4 Confusion matrix of ECG beats without noise across 10-folds.							
		Predicted		ACC (%)	PPV (%)	SEN (%)	SPEC (%)
		Normal	MI				
Original	Normal	9933	613	95.22	84.56	94.19	95.49
	MI	1814	38,368	95.22	98.43	95.49	94.19

ACC = Accuracy, PPV = Positive Predictive Value, SEN Sensitivity, SPEC = Specificity

Table 5 The overall classification results for the classification of normal and MI classes across 10-folds.								
Beats Type	TP	TN	FP	FN	ACC (%)	PPV (%)	SEN (%)	SPEC (%)
Noise	37,655	9790	756	2527	93.53	98.03	93.71	92.83
Without Noise	38,368	9933	613	1814	95.22	98.43	95.49	94.19

TP = True Positive, TN = True Negative, FP = False Positive, FN = False Negative
ACC = Accuracy, PPV = Positive Predictive Value, SEN = Sensitivity, SPEC = Specificity

Figure 3.8: les résultats obtenus dans le travail [43].

La matrice de confusion pour les battements ECG avec bruit et sans bruit est présentée respectivement dans les tableaux 3 et 4 [figure 3.7]. On peut observer à partir du tableau 3 que, sur 10 546 battements ECG normaux, environ 7,17% des battements ECG sont classés à tort comme IM. De même, pour l'IM, un total de 6,29% des battements ECG sont classés à tort comme des battements ECG normaux. De même, dans le tableau 4, 94,19% des battements ECG sont correctement classés comme battements ECG normaux et 4,51% sont classés à tort comme battements ECG normaux.

De plus, les valeurs PPV pour chaque classe (normale et MI) sont enregistrées dans les tableaux 3 et 4. Dans le tableau 3, la PPV dans la classe normale est de 79,48% alors que la PPV dans la classe MI est de 98,03%. Cela montre que la probabilité de détecter correctement les signaux ECG MI des signaux ECG sont plus élevés par rapport à la détection correcte des signaux ECG normaux. De même, dans le tableau 4, la PPV dans les classes normales et IM est respectivement de 84,56% et 98,43%. Cela montre également que la probabilité d'identifier les signaux ECG MI est plus élevée que l'identification des signaux ECG normaux dans les signaux ECG avec suppression du bruit.

Le taux de performance des deux battements ECG avec et sans bruit est résumé dans le tableau 5. Une précision moyenne, une sensibilité et une spécificité de 93,53%, 93,71% et 92,83% sont obtenues en utilisant des battements ECG avec bruit introduit respectivement.

Thermore de fourrure, la précision moyenne la plus élevée de 95,22% de sensibilité de 95,49% et de spécificité de 94,19% est obtenue pour un battement ECG sans bruit.

3.5 Approche 4 : ECG Heartbeat Classification : A Deep Transferable Representation

Dans [47], ils ont proposé une méthode basée sur des réseaux neuronaux convolutifs profonds pour la classification des battements cardiaques qui est capable de classer avec précision cinq arythmies différentes conformément à la norme AAMI EC57. De plus, ils proposent une méthode pour transférer les connaissances acquises sur cette tâche à la tâche de classification de l'infarctus du myocarde (IM).

Dans cet article, ils utilisent les bases de données PhysioNet MIT-BIH sur l'arythmie et l'ECG diagnostique PTB comme source de données pour les enregistrements ECG étiquetés [48]. De plus, ils démontrent que les connaissances acquises à partir de la première base de données peuvent être transférées avec succès pour la formation de modèles d'inférence pour la seconde. Dans toutes leurs expériences, ils ont utilisé l'ECG lead II ré-échantillonné à la fréquence d'échantillonnage de 125 Hz comme entrée.

▪ Résultats Obtenus :

Selon les résultats, la méthode suggérée est capable de faire des prédictions avec des précisions moyennes de 93,4 % et 95,9 % sur la classification de l'arythmie et la classification de l'MI, respectivement (Voir les Figures 3.9 et 3.10) .

Work	Approach	Average Accuracy (%)
This Paper	Deep residual CNN	93.4
Acharya <i>et al.</i> [5]	Augmentation + CNN	93.5
Martis <i>et al.</i> [6]	DWT + SVM	93.8
Li <i>et al.</i> [7]	DWT + random forest	94.6

Figure 3.9: Comparaison des résultats de la classification du rythme cardiaque[47].

Work	Accuracy (%)	Precision (%)	Recall (%)
This Paper¹	95.9	95.2	95.1
Acharya <i>et al.</i> [8] ¹	93.5	92.8	93.7
Kojuri <i>et al.</i> [9] ²	95.6	97.9	93.3
Sun <i>et al.</i> [10] ³	–	82.4	92.6
Sharma <i>et al.</i> [11] ³	96	99	93

Figure 3.10: Comparaison des résultats de la classification MI[47].

3.6 Approche 5 : ECG Classification Using Wavelet Packet Entropy and Random Forests

Dans [49], ils proposent une méthode pour classer les signaux ECG en utilisant l'entropie par paquets d'ondelettes (WPE) et les forêts aléatoires (RF) suivant les recommandations de l'Association pour l'avancement de l'Instrumentation médicale (AAMI) et le schéma inter-patient. Plus précisément, ils décomposent d'abord les signaux ECG par décomposition en paquets d'ondelettes (WPD), puis calculent l'entropie à partir des coefficients décomposés en tant que caractéristiques représentatives, et enfin ils utilisent les RF pour construire un modèle de classification ECG.

La base de données sur l'arythmie MIT-BIH a été utilisée comme source de données pour les expériences [50]. La base de données contient 48 enregistrements de 30 minutes, échantillonnés à 360 Hz, provenant de 47 sujets différents (25 hommes âgés de 32 à 89 ans et 22 femmes âgées de 23 à 89 ans). Dix-huit types de battements cardiaques ont été étiquetés au total. Les enregistrements numérotés 201 et 202 sont du même sujet.

Cette base de données a été divisée en un ensemble de formation et un ensemble de tests suivant les recommandations de l'AAMI et le schéma inter-patients.

Le modèle de classification a été construit sur la formation définie par RF et enfin les performances ont été évaluées sur la base des tests définis par le modèle. L'organigramme du cadre proposé est présenté à la Figure 3.11. Puisque cette étude se concentre principalement sur l'extraction et la classification des caractéristiques, ils ont simplement appliqué un filtre à réponse impulsionnelle finie (FIR) à 12 tap, passe-bas, avec un point de 3 dB à 35 Hz aux données dans la phase de prétraitement, et ils ont pris les 71 points précédant et suivant le pic R, respectivement (143 points, y compris le point R au total), pour composer un échantillon segmenté dans la phase de segmentation.

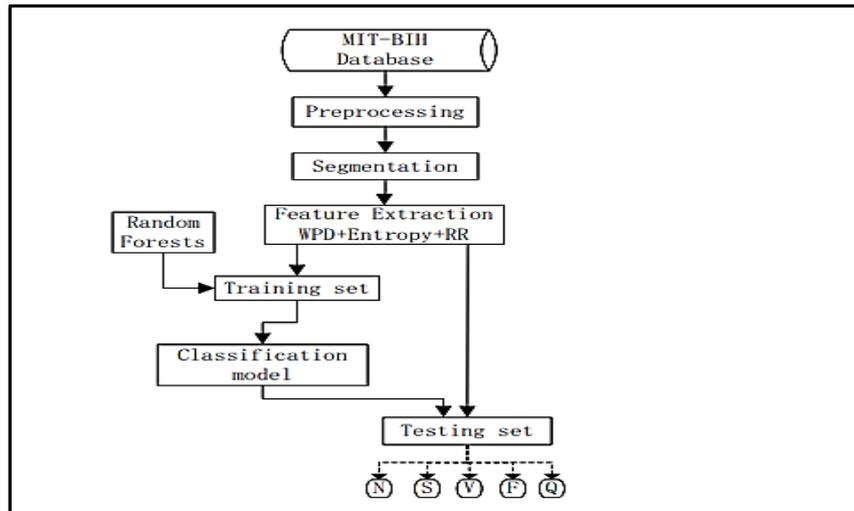


Figure 3.11: Organigramme de la méthode proposée [49].

▪ Résultats obtenus :

Tout d'abord, ils ont comparé la méthode proposée avec certains travaux antérieurs et les résultats sont présentés dans la figure 3.11. On peut voir à partir de ce tableau que la méthode proposée atteint la plus grande PRÉCISION de 94,61%, suivi des 94,22% de Song et des 92,17% de Ye [51], tandis que la méthode de Yu [52] (ICA + RR et PNN) atteint le plus bas ACC de 88,87%.

Method	Features	Classifier	ACC	N	S	V	F	Q
				SE+P/FPR	SE+P/FPR	SE+P/FPR	SE+P/FPR	SE+P/FPR
Yu [5]	ICA + RR	PNN	88.87	89.02/99.80/92.55	13.64/0.16/3.69	100.00/0.12/6.47	0.00/0.00/0.78	0.00/0.00/0.01
Song [9]	DWT + LDA + RR	SVM	94.22	95.79/98.73/13.66	0.00/0.00/3.70	79.03/97.55/0.17	0.00/0.00/0.78	0.00/0.00/0.01
Ye [11]	DWT + ICA + PCA + RR	SVM	92.17	92.18/99.71/7.01	0.00/0.00/3.69	91.90/52.50/3.20	0.00/0.00/0.78	0.00/0.00/0.01
The Proposed	WPE + RR	RF	94.61	94.67/99.73/3.92	20.00/0.16/3.69	94.20/89.78/0.71	50.00/0.52/0.78	0.00/0.00/0.01

ACC: Accuracy; S: Supraventricular ectopic beat (SVEB); V: Ventricular ectopic beat (VEB); F: Fusion beat; Q: Unknown beat; N: Any heartbeat not categorized as SVEB, VEB or Q; SE: Sensitivity; +P: positive predictivity; FPR: False positive rate; ICA: Independent component analysis; RR: the time between the R peaks of two heartbeats; PNN: Probabilistic neural network; DWT: Discrete wavelet transform; LDA: Linear discriminant analysis; PCA: Principal component analysis; SVM: Support vector machine; WPE: Wavelet packet entropy; RF: Random forests.

Figure 3.12 : Résultats (en %) obtenus par la méthode proposée et certains travaux antérieurs suite à la Recommandations de l'AAMI et schéma inter-patients[49].

Les résultats expérimentaux présentés dans ce travail montrent que la méthode proposée par la classification des signaux ECG en utilisant l'entropie du paquet d'ondelettes et les forêts aléatoires est prometteuse. Les auteurs ont évalué leur méthode sur l'ensemble de données MIT-

BIH Arythmie étant constaté qu'elle est capable de classifier avec précision différents types d'arythmies cardiaques. Les résultats montrent également que cette méthode est plus performance que les méthodes comparées dans l'étude.

3.7 Approche 6 : Application of Higher Order Cumulant Features for Cardiac Health Diagnosis Using ECG Signals

Diverses méthodes ont été proposées pour détecter les anomalies cardiaques dans les enregistrements ECG. L'application de caractéristiques de spectres d'ordre supérieur (HOS) est une approche apparemment prometteuse car elle peut capturer la nature non linéaire et dynamique des signaux ECG.

Dans cet article [53], ils ont automatiquement classé cinq types de battements à l'aide de caractéristiques HOS (higher order cumulants) en utilisant deux approches différentes. Les cinq types de battements ECG sont normaux (N), bloc de branche droit (RBBB), bloc de branche gauche (LBBB), prématuré auriculaire contraction (APC) et contraction prématurée ventriculaire (VPC). Dans la première approche, les caractéristiques cumulatives du signal ECG segmenté ont été utilisées pour la classification ; alors que dans la deuxième approche, les cumulatifs des coefficients de transformée en ondelettes discrètes (DWT) ont été utilisés comme caractéristiques pour les classificateurs. Dans les deux approches, les caractéristiques cumulatives ont été soumises à une réduction des données à l'aide de l'analyse en composantes principales (ACP) et classées à l'aide d'un réseau neuronal à trois couches (NN) et d'une machine à vecteurs de support des moindres carrés Classificateurs (LS-SVM).

Dans ce travail, ils ont utilisé 10 000 battements normaux et 24 989 battements anormaux (7250 RBBB, 8069 LBBB, 2544 APC et 7126 VPC) de la base de données d'arythmie MIT-BIH [54] pour l'analyse. L'ensemble de données est échantillonné à 360 Hz.

La Figure 3.13 montre le système proposé pour une classification auto-accouplée de cinq types de battements ECG.

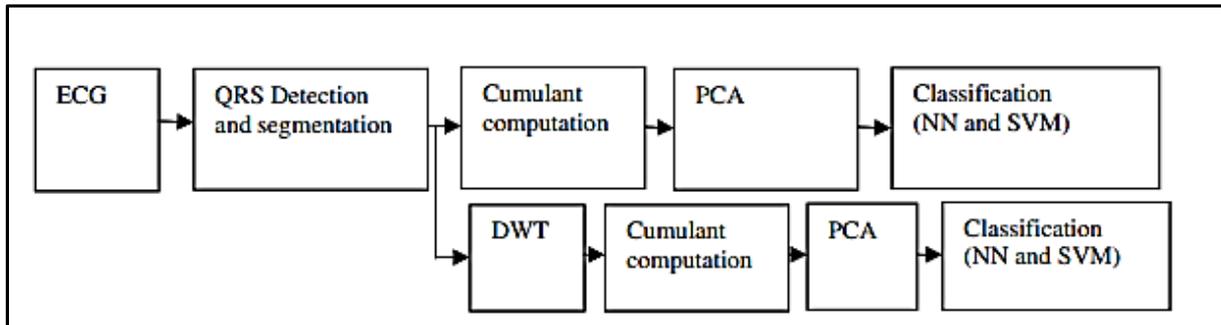


Figure 3.13 : Système proposé pour la classification des battements ECG[53].

▪ **Résultats obtenus :**

Deux méthodologies ont été étudiées dans ce travail. Dans la première méthodologie, le cumulatif de troisième ordre, a été calculé pour l'ECG segmenté. PCA réduit un grand nombre de coefficients cumulatifs en 12 composants. Dans la deuxième méthodologie, les coefficients DWT ont été utilisés pour le calcul des cumulants. Sur la base de la distribution des fréquences dans diverses bandes du tracé PSD, deux sous-bandes ont été considérées dans la décomposition DWT, approximation de quatrième niveau et détail de quatrième niveau. Un cumulatif de troisième ordre a été calculé sur chacune des deux sous-bandes. De chacune des sous-bandes, six composantes (12 caractéristiques au total) ont été utilisées pour la classification des motifs à l'aide de NN et LS-SVM. Les performances du système ont été évaluées en termes d'exactitude de classification.

Dans la première méthodologie, il ressort de la Figure 3.14 que le classificateur NN a fourni la précision de classification moyenne la plus élevée de 94,52 %, la sensibilité moyenne et la spécificité moyenne de 98,61% et 98,41%, respectivement.

	Average sensitivity (%)	Average specificity (%)	Average PPV (%)	Average accuracy (%)
NN	98.61	98.41	99.36	94.52
LS-SVM (linear)	96.48	81.41	93.07	74.30
LS-SVM (quadratic)	98.56	95.48	98.23	83.96
LS-SVM (polynomial)	98.32	96.69	98.70	90.59
LS-SVM (RBF)	99.72	96.69	98.70	94.30

Figure 3.14 : Résultats de classification des différents classificateurs pour[53].

La deuxième méthodologie implique le calcul des cumulants des coefficients DWT de l'ECG dans des sous-bandes sélectionnées. Il ressort de la Figure 3.15 que LS-SVM avec noyau RBF a fourni une précision maximale de 93,76% pour la deuxième méthodologie, une sensibilité moyenne et une spécificité moyenne de 99,46% et 97,36%, respectivement.

	Average sensitivity (%)	Average specificity (%)	Average PPV (%)	Average accuracy (%)
NN	98.51	97.80	99.12	93.61
LS-SVM (linear)	99.66	82.72	91.98	73.36
LS-SVM (quadratic)	97.98	94.94	99.31	81.47
LS-SVM (polynomial)	97.89	96.21	99.42	86.87
LS-SVM (RBF)	99.46	97.36	99.13	93.76

Figure 3.15: Résultats de classification des différents classificateurs pour [53].

La première méthode a donné de meilleurs résultats et a obtenu la précision de classification moyenne la plus élevée de 94,52%, une sensibilité moyenne de 98,61%, une spécificité moyenne de 98,41%, en utilisant le classificateur NN. Cette précision, qui a été obtenue en utilisant la première approche, s'est avérée supérieure à celles obtenues par des études antérieures utilisant d'autres caractéristiques non linéaires.

3.8 Approche 7: Classification of Cardiac Arrhythmias by Artificial Neural Networks

L'objectif de cet article [55], est de proposer un système d'analyse et de classification des battements cardiaques basé sur la combinaison des réseaux de neurones artificiels à apprentissage supervisé et non supervisé. Ce système reçoit en entrée 12 paramètres temporels et morphologiques caractérisant un battement ECG, et qui sont déterminés principalement en utilisant la transformée en ondelettes. Quatre types d'arythmies sont considérés dans cette étude : Extrasystole Ventriculaire (EV), Extrasystole Auriculaire (EA), Bloc de Branche Droit (BBD), et Bloc de Branche Gauche (BBG), en plus du battement normal (N).

Dans cette étude, et dans le but d'obtenir un système de classification efficace, ils proposent une structure neuronale hybride, combinant deux étages :

- Le premier étage est constitué d'une carte de Kohonen [56] à apprentissage non supervisé, qui réalise une tâche de pré classification.
- Le deuxième étage est un réseau MLP (Multi layer Perceptron) à apprentissage supervisé et qui est chargé de la classification finale.

Les signaux cardiaques utilisés dans cette étude ont été obtenus à partir de la base de données MIT/BIH qui est la norme employée par beaucoup de chercheurs pour l'évaluation des détecteurs d'arythmies [57, 58, 59, 60]. Cette base contient 48 enregistrements, chaque enregistrement a une durée de 30mn et une fréquence d'échantillonnage de 360HZ ; A chaque battement cardiaque est associé son type : normal, extrasystole ventriculaire, bloc de branche gauche, droit.

La base MIT-BIH a permis la création de deux bases de données : une pour l'apprentissage, l'autre pour le test qui serviront pour l'entraînement et l'évaluation du classifieur. Les battements sélectionnés sont donnés dans la figure suivante.

Type	Patients MIT-BIH	Base Apprentissage	Base test
NORMAL	100,103,105,108,	500	500
BBG	112,113,114,115,117	500	500
BBD	109,111,207,214	500	500
EV	118,124,212,231,232	500	500
EA	106,200,119, 214,203 ,208 100,209 ,118,202, 200,213,220	500	500
Total		2500	

Figure 3.16: Patients et nombre de battements utilisés[55].

Dans cette étude, pour minimiser le risque d'erreur du diagnostic, ils proposent un réseau de neurones hybride similaire à celui utilisé dans [61], combinant :

- Une carte SOM (Self Organizing Map) à apprentissage non supervisé qui permet d'effectuer une pré classification en regroupant les vecteurs d'entrée similaire en clusters.

- Un réseau LVQ (Learning Vector Quantization) à apprentissage supervisé, pour améliorer les frontières entre catégories.
- Un réseau MLP, chargé de la classification finale.

Le réseau simulé pour la classification des battements cardiaque est une SOM bidimensionnelle composée de $N \times N$ neurones artificiels.

Pour mieux affiner les frontières entre classes et diminuer le taux global d'erreur de reconnaissance, ils ont encore entraîné le réseau avec l'algorithme d'apprentissage LVQ2. Ainsi, pour la classification supervisé finale des battements ECG, ils ont opté pour un réseau MLP, qui est l'un des modèles connexionnistes le plus efficace et les utilisé pour les taches de régression et de discrimination. Dans cette étude ils ont utilisé un MLP à trois couches.

L'apprentissage du réseau a été réalisé en utilisant l'algorithme de la rétro propagation du gradient [62] avec une base d'apprentissage comportant 1100 battements tirés de la base MIT-BIH et représentant les cinq types de classes.

- **Résultat obtenus :**

Une fois l'apprentissage des réseaux terminé, il est nécessaire de les tester sur une base de données différente de celles utilisées pour l'apprentissage. La figure 3.17 illustre le taux de généralisation et la matrice de confusion du système de classification. On remarque que le nombre de mauvaises classifications est très bas sur l'échantillon d'étude, avec un taux global dépassant 95% en généralisation. On remarque aussi que la confusion entre les classes est faible ce qui démontre l'efficacité de la méthode proposée

	apprentissage	Test		NOR	BBG	BBD	EV	EA
NORM	100	99.2	NORM	99.2	0	0	0	0.8
BBG	99.7	92.9	BBG	3.3	92.9	0	0	3.3
BBD	99.8	99.2	BBD	0	0	99.2	0.6	0
EV	100	98.2	EV	0	0	1.8	98.2	0
EA	100	89.7	EA	5	5.3	0	0	89.7
Taux Global (%)	99.9	95.18						

(a)
(b)

Figure 3.17: (a) Taux de reconnaissance en% du classifieur, (b) : Matrice de confusion entre classes [55].

Dans ce travail, le système a été validé sur des enregistrements extraits de la base de données MIT-BIH, les résultats obtenus sont très encourageants.

3.9 Comparaison des approches

D'après les résultats obtenus des approches de Deep Learning utilisées pour faire la classification des données ECG, on remarque que les modèles de CNN donnent des meilleurs résultats avec un taux de précision qui dépasse 98% que les autres modèles qui ont donné un taux de précision de 93% et les modèles RNN qui ont donnés pour leur meilleur modèle un résultat de 88%.

3.10 Conclusion

Nous avons vu dans ces divers articles, qu'il existe plusieurs méthodes de classification des arythmies ECG. Plus généralement, cette problématique se traduit par un problème de classification multi classe avec des approches qui se basent sur des algorithmes de Deep Learning. Pour répondre à notre problématique, on va s'inspirer des articles lus en testant plusieurs algorithmes de Deep Learning et sélectionner le plus optimal pour notre problématique.

Chapitre 4 : Système Réalisé



4.1 Introduction

Notre projet porte sur la classification des signaux ECG avec la méthode de Deep Learning. Pour atteindre cet objectif et pour obtenir la meilleure performance possible, on propose le système suivant : pour faire la classification des données ECG, on va passer par deux étapes :

- ✚ **La classification binaire** : qui définit juste si la personne est malade ou pas, pour cela on utilise une base de données qui contient deux classes : 1 pour « malade », 0 pour « pas malade ». Puis si le résultat est égal à 0 la procédure est définie et la personne est saine, sinon si le résultat est égal à 1, on passe à la classification multi classe.
- ✚ **La classification multi classe** : on utilise une base de données qui contient 4 classes qui correspondent à des types de maladies cardiaques. La classification multi classe va nous permettre de trouver le type de la maladie donc soit : APC, PAB, PVC ou RBB.

On applique les mêmes étapes pour les algorithmes utilisés dans ce projet, par exemple avec les algorithmes de Deep Learning, on fait la classification binaire puis toujours avec les mêmes algorithmes on fait la classification multi classe.

La figure suivante montre l'architecture globale de la classification des signaux ECG :

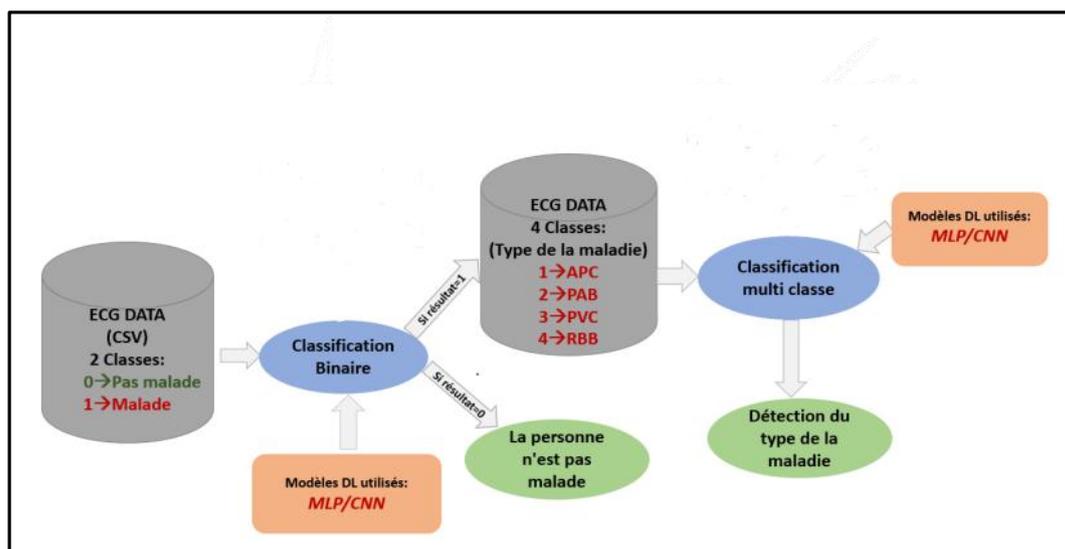


Figure 4.1: L'architecture globale de la classification des signaux ECG.

4.2 Justification de la méthodologie suivie

La raison pour laquelle nous choisissons de passer par deux étapes est liée au nombre de classes qui n'est pas équitable. Nous avons beaucoup plus de classes 0 que d'autres classes, donc si on fait juste une seule classification multi classe, le résultat des algorithmes sera faible et ne sera pas fiable pour un domaine qui est très sensible où la précision est très importante.

En plus de ça, en général la classification binaire donne toujours un meilleur résultat que la classification multi classe si on utilise la même base de donnée.

Il est donc plus prioritaire de déterminer d'abord si une personne est malade ou non, et ensuite, si elle est saine, on n'aura pas besoin d'aller plus loin.

4.3 Base de données utilisée

La base utilisée, soit l'ensemble de données Physionet MIT-BIH a été utilisé pour explorer la classification des battements cardiaques à l'aide d'architectures de réseaux neuronaux profonds et pour observer certaines des capacités d'apprentissage par transfert sur celui-ci. Les signaux correspondent aux formes d'électrocardiogramme (ECG) des battements cardiaques pour le cas normal et les cas affectés par différentes arythmies et infarctus du myocarde [62]

La base de données utilisée est donc un fichier CSV qui contient :

- Nombre d'échantillons : 109446
- Nombre de catégories : 5
- Fréquence d'échantillonnage : 125 Hz
- Source des données : Ensemble de données Physionet MIT-BIH sur l'arythmie
- 5 Classes : ["N": 0, "S": 1, "V": 2, "F": 3, "Q": 4]

Avec :

- **N** : Person Normal.
- **S** : Contraction auriculaire prématurée (APC).
- **V** : Battement du rythme (PAB).
- **F** : Contraction ventriculaire prématurée (PVC).
- **Q** : Bloc de branche droit (RBB).

4.3.1 Prétraitement de données

- ✚ **Cas de la classification binaire** : Nous savons que les classes 1, 2, 3 et 4 correspondent aux signaux des personnes malades, et pour effectuer la classification binaire, il est nécessaire qu'il y ait seulement deux classes : 0 et 1. Pour ça, on remplace toutes les classes qui ont les numéros de classe 2, 3 et 4 par 1, pour n'avoir qu'une seule classe des signaux des personnes malades. On aura donc deux classes 0 et 1 et on pourra faire une classification binaire.
- ✚ **Cas de la classification multi classe** : Le but de la classification multi classe est de déterminer le type de la maladie vu qu'on sait si on passe à la classification multi classe, donc la personne est malade. Donc on prend la base de données originale qui avait 5 classes et on supprime toutes les lignes qui contiennent la classe 0 puis on n'aura à la fin que 4 classes qui correspondent aux types des maladies cardiaques.

4.3.2 Aperçu sur la base de données

La base de données contient 109446 lignes et 188 colonnes (Figure 4.2).

- De 0 jusqu' à 186 se sont les attributs (features).
- La colonne 187 représente la classe cible (label) de la base de données qui peut être soit 0 ou 1 lors de la classification binaire et soit 1, 2, 3 ou 4 lors de la classification multi classe.

	0	1	2	3	4	5	6	7	...	180	181	182	183	184	185	186
187																
0	1.000000	0.606941	0.384181	0.254237	0.223567	0.276836	0.253430	0.184826	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1,0																
1	1.000000	0.951613	0.923963	0.853303	0.791859	0.734255	0.672043	0.685100	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1,0																
2	0.977819	0.899261	0.230129	0.032348	0.142329	0.223660	0.328096	0.367837	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1,0																
3	0.935618	0.801661	0.805815	1.000000	0.722741	0.480789	0.454829	0.319834	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1,0																
4	0.925265	0.433352	0.073620	0.079197	0.136643	0.182934	0.182934	0.182376	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1,0																

Figure 4.2 : Les 5 premières lignes de la base de données.

4.4 Les mesures de performances

Afin de mesurer les performances d'un modèle de Deep Learning, on fait appel généralement à la Confusion Matrix ou matrice de confusion.

a) Matrice de confusion

La matrice de confusion est en quelque sorte un résumé des résultats de prédiction pour un problème particulier de classification. Elle compare les données réelles pour une variable cible à celles prédites par un modèle. Les prédictions justes et fausses sont révélées et réparties par classe, ce qui permet de les comparer avec des valeurs définies (Figure 4.3).

Également connue sous le nom de tableau de contingence, la matrice de confusion permet d'évaluer les performances d'un modèle de classification. Elle montre donc à quel point un certain modèle peut être confus lorsqu'il fait des prédictions. Dans sa forme la plus simple, il s'agit d'une matrice 2X2. Pour des problèmes de classification plus complexes, il est toujours possible d'ajouter des lignes et des colonnes à la forme de base [63].

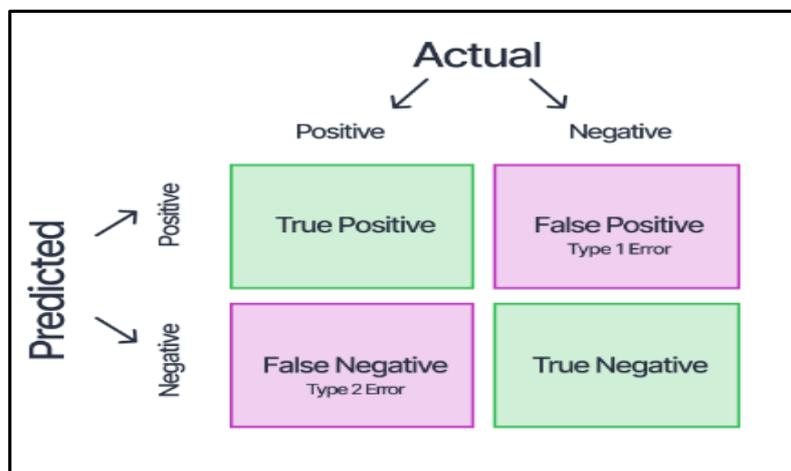


Figure 4.3 : Matrice de confusion [64].

Les résultats d'une matrice de confusion sont classés en quatre grandes catégories : les vrais positifs, les vrais négatifs, les faux positifs et les faux négatifs.

- **Les vrais positifs ou TP (True Positive)** : indiquent les cas où les prédictions et les valeurs réelles sont effectivement positives.
- **Les vrais négatifs ou TN (True Negative)** : indiquent par contre les cas où les prédictions et les valeurs réelles sont toutes les deux négatives.
- **Les faux positifs ou FP (False Positive)** : indiquent quant à eux une prédiction positive contraire à la valeur réelle qui est négative. Ils sont également considérés comme des erreurs de type 1.

- **Les faux négatifs ou FN (False Négatif)** : font référence aux cas où les prédictions sont négatives alors que les valeurs réelles sont positives. Ils sont également considérés comme des erreurs de type 2.

b) Les métriques

Différentes métriques peuvent être calculées à partir du tableau de contingence afin d'en faciliter l'interprétation. C'est par exemple le cas de la précision, du rappel et du F1 score. Ces indicateurs permettent de mieux apprécier la qualité de précision du modèle [64].

- **Accuracy** : Le nombre d'échantillons correctement classés parmi tous les échantillons présents dans l'ensemble d'essai.

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN} \quad (4.1)$$

- **Précision** : Le nombre d'échantillons appartenant réellement à la classe positive parmi tous les échantillons qui ont été prédits comme appartenant à la classe positive par le modèle.

$$\text{Precision} = \frac{TP}{TP+FP} \quad (4.2)$$

- **Recall (Rappel)** : Le nombre d'échantillons prédits correctement comme appartenant à la classe positive parmi tous les échantillons qui appartiennent réellement à la classe positive.

$$\text{Recall} = \frac{TP}{TP+FN} \quad (4.3)$$

- **F1-Score** : La moyenne harmonique des scores de précision et de rappel obtenus pour la classe positive.

$$\mathbf{F1-Score} = \frac{2 \times \mathbf{Precision} \times \mathbf{Recall}}{\mathbf{Precision} + \mathbf{Recall}} \quad (4.4)$$

4.5 Présentation des outils utilisés

Dans cette partie, nous aborderons les outils qui ont aidé à créer ce projet, tels que Visual Studio Code, Python, etc.

4.5.1 Visual Studio Code

VS Code est une version réduite de l'environnement officiel de développement Microsoft qui se concentre exclusivement sur l'éditeur de code. Il est multiplateforme et supporte des syntaxes pour un grand nombre de langages.

L'outil fournit du support et de l'assistance pour de nombreux langages différents : HTML, CSS, JavaScript, différentes variantes de python, C, JSON, Java, SQL, PHP, Ruby, Visual Basic, et beaucoup d'autres, supportant les highlights, l'auto-indentation, les snippets et l'auto-complétion [65].

4.5.2 Python

Python est un langage de programmation (Figure 4.4) largement utilisé dans les applications Web, le développement de logiciels, la science des données et le Machine Learning (ML). Les développeurs utilisent Python parce que c'est un langage efficace et facile à apprendre, et qui peut s'exécuter sur de nombreuses plateformes différentes. Le logiciel Python peut être téléchargé gratuitement, il s'intègre bien avec tous les types de systèmes et accélère le développement [66].

Les avantages de Python sont les suivants [66] :

- Les développeurs peuvent facilement lire et comprendre un programme Python, car il possède une syntaxe de base, semblable à celle de l'anglais.
- Python rend les développeurs plus productifs, car ils peuvent écrire un programme Python en utilisant moins de lignes de code par rapport à de nombreux autres langages.

- Python dispose d'une vaste bibliothèque standard qui contient des codes réutilisables pour presque toutes les tâches. Par conséquent, les développeurs ne doivent pas écrire le code à partir de zéro.
- Les développeurs peuvent facilement utiliser Python avec d'autres langages de programmation populaires comme Java, C et C++.
- Python est portable sur différents systèmes d'exploitation informatiques tels que Windows, macOS, Linux et Unix.



Figure 4.4 : Logo Python [67].

4.5.3 TensorFlow

TensorFlow est une bibliothèque open source de logiciels (Figure 4.5) pour le flux de données et la programmation différentielle pour diverses tâches. De même, TensorFlow est utilisée dans l'apprentissage automatique par les réseaux de neurones. Développée par Google en 2011 sous le nom de DistBelief, TensorFlow a été officiellement lancé en 2017 gratuitement.

La bibliothèque est capable de fonctionner sur plusieurs processeurs et est disponible sur plusieurs plateformes, y compris pour les mobiles. Le nom vient de tableaux multidimensionnels appelés tenseurs (Tensor), couramment utilisés dans les réseaux de neurones. Aujourd'hui, cette bibliothèque est utilisée par les grandes entreprises comme Intel, Twitter, Lenovo [68] .



Figure 4.5 : Logo TensorFlow [69].

4.5.4 Keras

Keras est une bibliothèque open source (Figure 4.6) écrite en Python (sous licence MIT) basée principalement sur les travaux du développeur de Google François Chollet dans le cadre du projet ONEIROS (**O**pen-ended **N**euro-**E**lectronic **I**ntelligent **R**obot **O**perating **S**ystem).

Une première version du logiciel multiplateforme a été publiée le 28 mars 2015. Le but de cette bibliothèque est de permettre la constitution rapide de réseaux neuronaux. Dans ce cadre, Keras ne fonctionne pas comme un Framework propre mais comme une interface de programmation applicative (API) pour l'accès et la programmation de différents frameworks d'apprentissage automatique. Theano, Microsoft Cognitive Toolkit (anciennement CNTK) et TensorFlow font notamment partie des frameworks pris en charge par Keras [70].



Figure 4.6 : Logo Keras [71].

4.5.5 Scikit-learn

Scikit-learn, encore appelé *sklearn*, est la bibliothèque la plus puissante et la plus robuste pour le machine learning en Python (Figure 4.7).

Elle fournit une sélection d'outils efficaces pour l'apprentissage automatique et la modélisation statistique, notamment la classification, la régression et le clustering via une interface cohérente en Python. Cette bibliothèque, qui est en grande partie écrite en Python, s'appuie sur NumPy, SciPy et Matplotlib [72].



Figure 4.7 : Logo Scikit-Learn [73].

4.6 Algorithmes utilisés

Après avoir préparé notre architecture de travail, et lu plusieurs articles qui traitent des problématiques qui ressemblent à la nôtre, on a choisi de tester deux algorithmes de Deep Learning qui sont le perceptron multicouche (**MLP**) et le **CNN**, afin de les comparer et de choisir le plus optimal pour notre problématique (Figure 4.8).

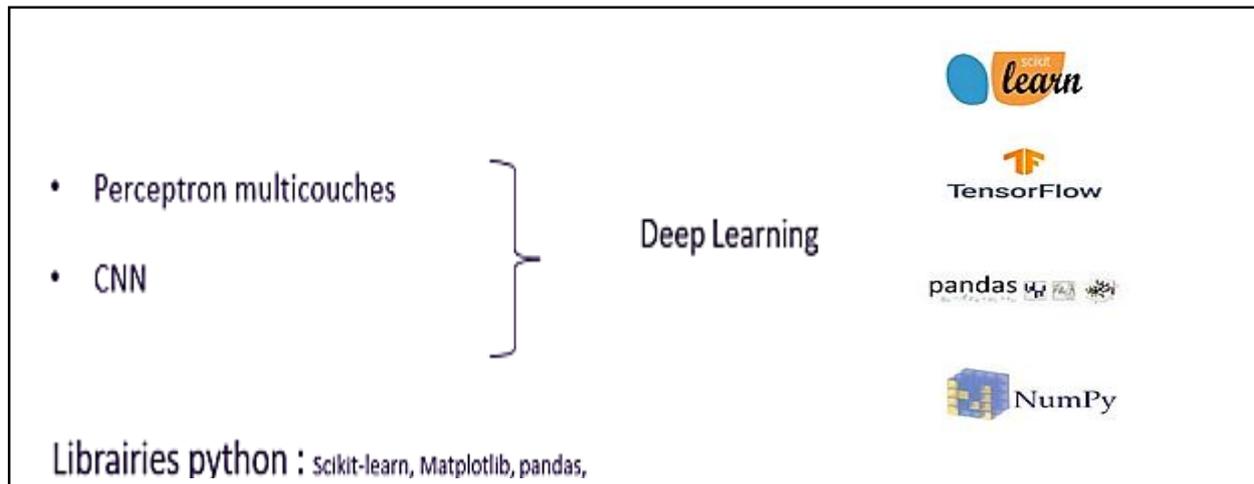


Figure 4.8 : Différents algorithmes utilisés.

4.6.1 Algorithme MLP

Dans le cas de la classification binaire et multi classe, nous utiliserons l'algorithme du MLP sur une base de données contenant deux classes : 0 correspondant à "pas malade" et 1 correspondant à "malade", ainsi qu'une base de données contenant quatre classes : 1, 2, 3 et 4 qui correspondent à un type de maladie, respectivement.

L'organigramme suivant décrit les étapes suivies pour créer un programme d'architecture MLP afin de réaliser la classification binaire et multiclasse.

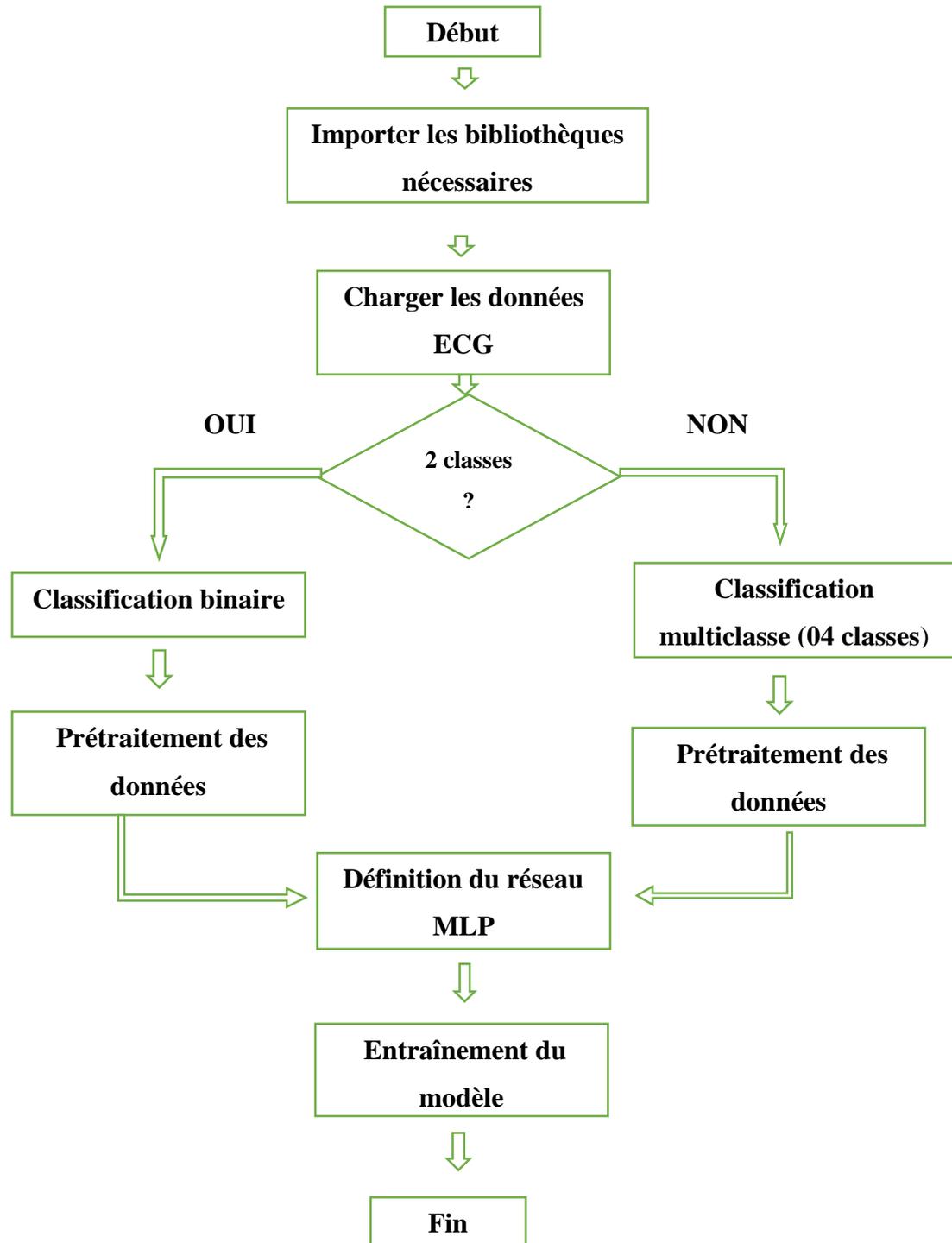


Figure 4.9 : Organigramme de L'architecture MLP

Voici un sous-organigramme qui explique les étapes de prétraitement des données de manière plus détaillée :

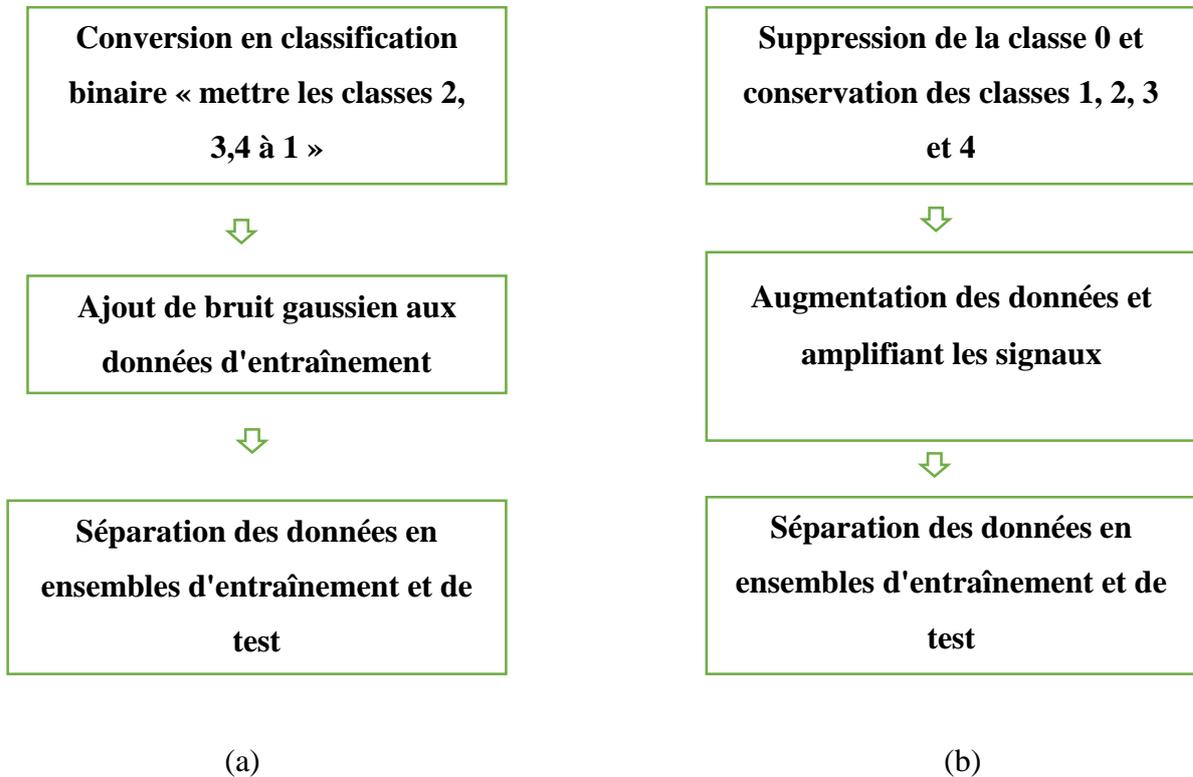


Figure 4.10 : Sous-organigramme pour le traitement des données, (a) : binaire, (b) : multiclasse.

a) Utilisation du MLP pour la classification binaire

La Figure ci-dessous donne un aperçu du code du programme MLP développé pour le cas d'une classification binaire.

```

50 def network(X_train, y_train, X_test, y_test):
51     im_shape = (X_train.shape[1],)
52     inputs_mlp = Input(shape=im_shape, name='inputs_mlp')
53     dense1 = Dense(64, activation='relu')(inputs_mlp)
54     dense2 = Dense(32, activation='relu')(dense1)
55     main_output = Dense(2, activation='softmax', name='main_output')(dense2) # Modification du nombre de clas
56
57     model = Model(inputs=inputs_mlp, outputs=main_output)
58     model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
59
60     callbacks = [
61         EarlyStopping(monitor='val_loss', patience=8),
62         ModelCheckpoint(filepath='best_model.h5', monitor='val_loss', save_best_only=True)
63     ]
64
65     history = model.fit(X_train, y_train, epochs=5, callbacks=callbacks, batch_size=32, validation_data=(X_tes
66     model.load_weights('best_model.h5')
67     return model, history
  
```

Figure 4.11: Le modèle MLP binaire.

Le modèle de MLP utilisé comporte trois couches :

- 1. Couche d'entrée :** Cette couche reçoit les signaux d'ECG en entrée avec une dimension de 186 (caractéristiques des signaux).
- 2. Couche cachée 1 :** Une couche dense avec 64 neurones et une fonction d'activation ReLU. Cette couche extrait des caractéristiques des données d'entrée.
- 3. Couche cachée 2 :** Une autre couche dense avec 32 neurones et une fonction d'activation ReLU. Cette couche continue à extraire des informations plus abstraites des caractéristiques extraites précédemment.
- 4. Couche de sortie :** La couche de sortie contient 2 neurones (pour les classes binaires) avec une fonction d'activation Softmax qui attribue des probabilités aux deux classes.

Le choix de ces couches et de leur taille est basé sur des considérations empiriques pour obtenir de bonnes performances de classification. Les fonctions d'activation Relu sont couramment utilisées car elles permettent au modèle d'apprendre des représentations non linéaires des données.

En résumé, ce code effectue la préparation des données, définit et entraîne un modèle de réseau de neurones à trois couches pour la classification binaire des signaux ECG, et évalue les performances du modèle à l'aide de différentes métriques. Le modèle a obtenu de bons résultats avec une précision de **92%** sur l'ensemble de test. La matrice de confusion a été affichée pour évaluer les performances du modèle.

b) Utilisation de MLP pour la classification multi classe

La Figure 4.12 donne un aperçu du code du programme MLP développé pour le cas d'une classification multiclasse.

```
109 n_obs, feature, depth = X_train.shape
110 batch_size = 500
111
112 K.clear_session()
113
114 inp = Input(shape=(feature, depth))
115 F1 = Flatten()(inp)
116 D1 = Dense(64)(F1)
117 A6 = Activation("relu")(D1)
118 D2 = Dense(64)(A6)
119 D3 = Dense(4)(D2) # Changed the number of classes from 5 to 4
120 A7 = Activation('softmax')(D3)
121
122 model = Model(inputs=inp, outputs=A7)
123 # Compilation et entraînement du modèle
124 model.summary()
```

Figure 4.12: Le modèle MLP multiclasse.

Le modèle de MLP utilisé comporte trois couches : une couche d'entrée, une couche cachée et une couche de sortie avec une activation Softmax pour la classification.

1. **La couche d'entrée** : est une couche de type Flatten qui transforme les données d'entrée en un vecteur 1D.
2. **La couche cachée 1** : est connectée à la couche d'entrée et contient 64 neurones.
3. **La couche cachée 2** : est connectée à la couche d'activation relu et contient également 64 neurones.
4. **La couche de sortie** : est également une couche dense avec 4 neurones correspondant aux classes cibles. avec une fonction d'activation Softmax qui attribue des probabilités aux 4 classes.

L'utilisation de deux couches cachées peut permettre au modèle d'apprendre des représentations plus complexes et de capturer des motifs plus profonds dans les données d'entrée.

Cela peut potentiellement améliorer les performances de classification du modèle en permettant une plus grande flexibilité dans la modélisation des relations entre les caractéristiques d'entrée et les classes cibles.

c) Résultats Obtenus

Après l'exécution des codes pour les deux types de classification, on a obtenu les résultats suivants :

- **Pour la classification binaire :**

La matrice de confusion permet d'évaluer la performance de notre modèle en reflétant les métriques du vrai positif, du vrai négatif, du faux positif et du faux négatif. La figure suivante montre cela.

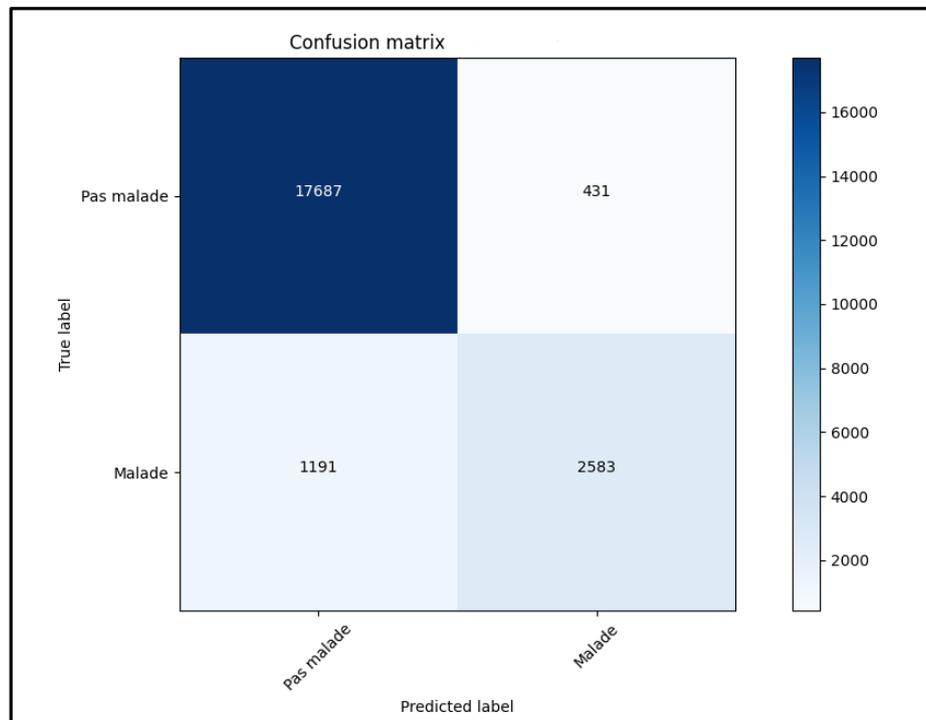


Figure 4.13 : La matrice de confusion MLP Binaire.

Les différentes valeurs de la matrice de confusion sont les suivantes :

Vrai positif (TP) = 2583 ; ce qui signifie que 2583 personnes de classe positive ont été correctement classés par le modèle.

Vrai négatif (TN) = 17687 ; ce qui signifie que 17687 personnes de classe négative ont été correctement classés par le modèle.

Faux positif (FP) = 1191 ; ce qui signifie que 1191 personnes de classe négative ont été incorrectement classés comme appartenant à la classe positive par le modèle.

Faux négatif (FN) = 431 ; ce qui signifie que 172 personnes de classe positive ont été incorrectement classés comme appartenant à la classe négative par le modèle.

➤ **Le rapport de classification :**

La figure suivante montre le taux de précision, de Recall et de f1-score des deux classes :

	precision	recall	f1-score	support
0	0.93	0.98	0.95	18118
1	0.88	0.63	0.74	3774
accuracy			0.92	21892
macro avg	0.91	0.81	0.85	21892
weighted avg	0.92	0.92	0.92	21892

Figure 4.14: Résultats MLP Binaire

Pour la classification binaire avec l'algorithme MLP, le taux de précision est de 88%, le Recall est de 63% et le f1-score est de 74% pour les personnes malades, ce qui est un bon résultat qui va nous permettre de savoir si la personne est malade.

▪ **Pour la classification multiclass :**

Afin de déterminer le type de maladie d'une personne précédemment identifiée comme malade lors de la classification binaire, nous utiliserons l'algorithme MLP pour réaliser une classification multiclass. La figure suivante montre le résultat que nous avons obtenu.

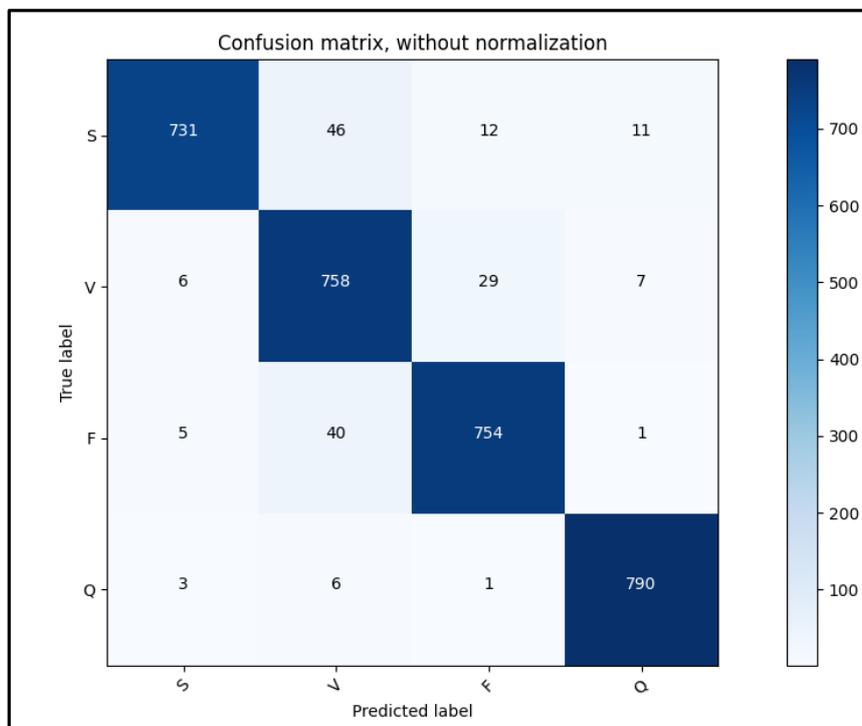


Figure 4.15 : La matrice de confusion MLP Multiclass.

➤ **Le rapport de classification :**

La figure suivante montre le taux de précision, de Recall et de f1-score de chaque type de maladie :

	precision	recall	f1-score	support
0	0.98	0.91	0.95	800
1	0.89	0.95	0.92	800
2	0.95	0.94	0.94	800
3	0.98	0.99	0.98	800
accuracy			0.95	3200
macro avg	0.95	0.95	0.95	3200
weighted avg	0.95	0.95	0.95	3200

Figure 4.16 : Résultats MLP Multiclasse.

Le rapport de classification indique que Notre modèle de classification multiclasse a une performance globalement élevée, avec de bonnes précisions, rappels et F1-scores pour chaque classe. L'Accuracy de ce modèle est de 94%, ce qui signifie qu'il prédit correctement la classe des échantillons dans environ 94% des cas. Ces résultats montrent que le modèle est capable de bien classer les données

4.6.2 Algorithme CNN

Pour déterminer si une personne est malade ou non et identifier le type de maladie, nous utiliserons l'algorithme CNN (Convolutional Neural Network) pour effectuer à la fois la classification binaire et la classification multi-classe. Pour cela, nous suivrons les mêmes étapes que celles utilisées dans les deux classifications précédentes avec l'algorithme MLP, à l'exception du modèle qui sera différent.

Voici l'organigramme qui décrit les étapes pour créer un programme d'architecture CNN afin de réaliser la classification binaire et multiclasse.

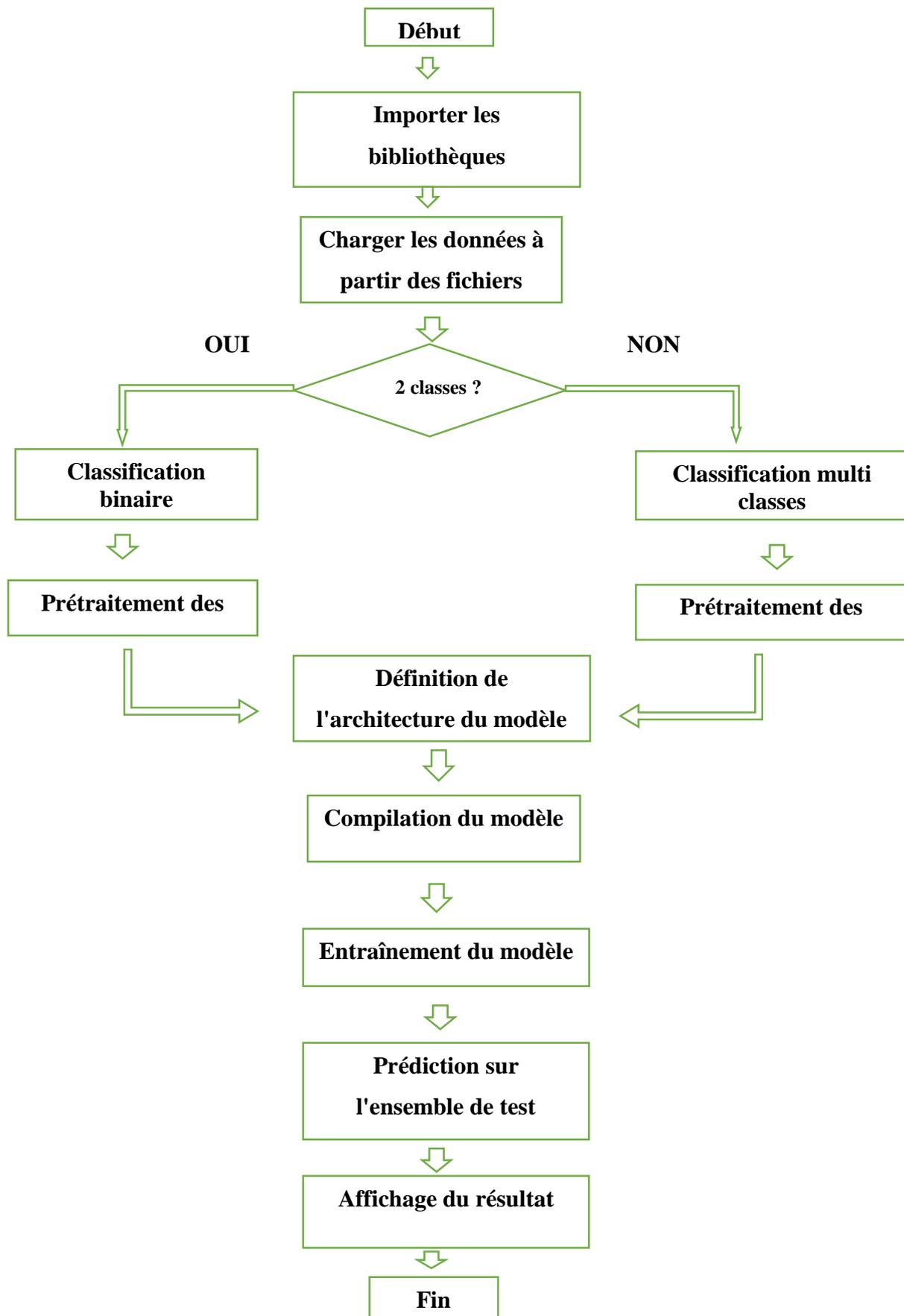


Figure 4.17 : Organigramme de L'architecture CNN.

Les étapes de prétraitement des données sont présentées dans la Figure 4.10.

a) Utilisation du CNN pour la classification binaire

```

89 # Définition de l'architecture du modèle
90 n_obs, feature, depth = X.shape
91 batch_size = 500
92 K.clear_session()
93 inp = Input(shape=(feature, depth))
94 C = Conv1D(filters=32, kernel_size=5, strides=1)(inp)
95 C11 = Conv1D(filters=32, kernel_size=5, strides=1, padding='same')(C)
96 A11 = Activation("relu")(C11)
97 C12 = Conv1D(filters=32, kernel_size=5, strides=1, padding='same')(A11)
98 S11 = Add()([C12, C])
99 A12 = Activation("relu")(S11)
100 M11 = MaxPooling1D(pool_size=5, strides=2)(A12)
101
102 C21 = Conv1D(filters=32, kernel_size=5, strides=1, padding='same')(M11)
103 A21 = Activation("relu")(C21)
104 C22 = Conv1D(filters=32, kernel_size=5, strides=1, padding='same')(A21)
105 S21 = Add()([C22, M11])
106 A22 = Activation("relu")(S21)
107 M21 = MaxPooling1D(pool_size=5, strides=2)(A22)
108
109 C31 = Conv1D(filters=32, kernel_size=5, strides=1, padding='same')(M21)
110 A31 = Activation("relu")(C31)
111 C32 = Conv1D(filters=32, kernel_size=5, strides=1, padding='same')(A31)
112 S31 = Add()([C32, M21])
113 A32 = Activation("relu")(S31)
114 M31 = MaxPooling1D(pool_size=5, strides=2)(A32)
115
116 C41 = Conv1D(filters=32, kernel_size=5, strides=1, padding='same')(M31)
117 A41 = Activation("relu")(C41)
118 C42 = Conv1D(filters=32, kernel_size=5, strides=1, padding='same')(A41)
119 S41 = Add()([C42, M31])
120 A42 = Activation("relu")(S41)
121 M41 = MaxPooling1D(pool_size=5, strides=2)(A42)
122
123 C51 = Conv1D(filters=32, kernel_size=5, strides=1, padding='same')(M41)
124 A51 = Activation("relu")(C51)
125 C52 = Conv1D(filters=32, kernel_size=5, strides=1, padding='same')(A51)
126 S51 = Add()([C52, M41])
127 A52 = Activation("relu")(S51)
128 M51 = MaxPooling1D(pool_size=5, strides=2)(A52)
129
130 F1 = Flatten()(M51)
131
132 D1 = Dense(32)(F1)
133 A6 = Activation("relu")(D1)
134 D2 = Dense(32)(A6)
135 D3 = Dense(2)(D2)
136 A7 = Softmax()(D3)

```

Figure 4.18: Le modèle CNN binaire.

Le modèle utilisé dans le code fourni dans la figure ci-dessus comprend un total de 16 couches. Voici une ventilation des différentes couches utilisées :

- 1. couche d'entrée (Input) :** Cette couche représente les données d'entrée du modèle et spécifie la forme des données.

2. **couches convolutives (Conv1D)** : Ces couches convolutives appliquent des filtres sur les données d'entrée pour extraire des caractéristiques pertinentes. Chaque couche convolutive est suivie d'une activation ReLU (Activation("relu")) pour introduire une non-linéarité.
3. **couches de fusion (Add)** : Ces couches de fusion relient les sorties des couches convolutives précédentes en les additionnant.
4. **couches de pooling (MaxPooling1D)** : Ces couches de pooling réduisent la dimensionnalité des caractéristiques extraites en prenant la valeur maximale dans des fenêtres de pooling.
5. **couche dense (Dense)** : Cette couche dense est utilisée pour transformer les caractéristiques extraites en une représentation plus compacte.
6. **couche de sortie (Dense)** : Cette couche de sortie génère les prédictions finales en produisant des scores pour chaque classe. Elle est suivie d'une activation Softmax (Softmax) pour obtenir des probabilités normalisées.

La raison d'utiliser plusieurs couches convolutives et de pooling dans le modèle est de permettre l'extraction de caractéristiques hiérarchiques à partir des données d'entrée.

Les couches convolutives capturent des motifs locaux tandis que les couches de pooling agrègent les informations pertinentes. Cela permet au modèle de détecter des motifs de plus en plus complexes à mesure que les données passent à travers les couches, améliorant ainsi les performances de la classification.

b) Utilisation du CNN pour la classification multiclasse

Le modèle utilisé dans le code fourni dans la Figure 4.19 comprend un total de 16 couches.

```

inp = Input(shape=(feature, depth))
C = Conv1D(filters=32, kernel_size=5, strides=1)(inp)

C11 = Conv1D(filters=32, kernel_size=5, strides=1, padding='same')(C)
A11 = Activation("relu")(C11)
C12 = Conv1D(filters=32, kernel_size=5, strides=1, padding='same')(A11)
S11 = Add()([C12, C])
A12 = Activation("relu")(S11)
M11 = MaxPooling1D(pool_size=5, strides=2)(A12)

C21 = Conv1D(filters=32, kernel_size=5, strides=1, padding='same')(M11)
A21 = Activation("relu")(C21)
C22 = Conv1D(filters=32, kernel_size=5, strides=1, padding='same')(A21)
S21 = Add()([C22, M11])
A22 = Activation("relu")(S21)
M21 = MaxPooling1D(pool_size=5, strides=2)(A22)

30 C31 = Conv1D(filters=32, kernel_size=5, strides=1, padding='same')(M21)
31 A31 = Activation("relu")(C31)
32 C32 = Conv1D(filters=32, kernel_size=5, strides=1, padding='same')(A31)
33 S31 = Add()([C32, M21])
34 A32 = Activation("relu")(S31)
35 M31 = MaxPooling1D(pool_size=5, strides=2)(A32)
36 C41 = Conv1D(filters=32, kernel_size=5, strides=1, padding='same')(M31)
37 A41 = Activation("relu")(C41)
38 C42 = Conv1D(filters=32, kernel_size=5, strides=1, padding='same')(A41)
39 S41 = Add()([C42, M31])
40 A42 = Activation("relu")(S41)
41 M41 = MaxPooling1D(pool_size=5, strides=2)(A42)
42 C51 = Conv1D(filters=32, kernel_size=5, strides=1, padding='same')(M41)
43 A51 = Activation("relu")(C51)
44 C52 = Conv1D(filters=32, kernel_size=5, strides=1, padding='same')(A51)
45 S51 = Add()([C52, M41])
46 A52 = Activation("relu")(S51)
47 M51 = MaxPooling1D(pool_size=5, strides=2)(A52)
48 F1 = Flatten()(M51)
49 D1 = Dense(32)(F1)
50 A6 = Activation("relu")(D1)
51 D2 = Dense(32)(A6)
52 D3 = Dense(4)(D2)
53 A7 = Softmax()(D3)

```

Figure 4.19: Le modèle CNN multiclasse.

Les différentes couches utilisées sont :

- **1 couche d'entrée (Input) :** Cette couche spécifie la forme des données d'entrée.
- **6 couches convolutives (Conv1D) :** Ces couches convolutives appliquent des filtres sur les données d'entrée pour extraire des caractéristiques. Chaque couche convolutive est suivie d'une activation ReLU (Activation("relu")) pour introduire une non-linéarité.
- **5 couches de fusion (Add) :** Ces couches de fusion relient les sorties des couches convolutives précédentes en les additionnant.

- **5 couches de pooling (MaxPooling1D)** : Ces couches de pooling réduisent la dimensionnalité des caractéristiques en prenant la valeur maximale dans des fenêtres de pooling.
- **1 couche dense (Dense)** : Cette couche dense est utilisée pour transformer les caractéristiques extraites en une représentation plus compacte.
- **1 couche de sortie (Dense)** : Cette couche de sortie génère les prédictions finales en produisant des scores pour chaque classe. Elle est suivie d'une activation Softmax (Softmax) pour obtenir des probabilités normalisées.

Les couches convolutives sont utilisées pour capturer les motifs locaux dans les données d'entrée, tandis que les couches de pooling réduisent la dimensionnalité pour conserver les informations les plus importantes. Les couches de fusion permettent d'intégrer les caractéristiques extraites à différents niveaux de l'architecture du modèle.

c) Résultats obtenus

Après l'exécution des codes pour les deux types de classification avec l'architecteur CNN, on a obtenu ces résultats.

- **Pour la classification binaire :**

La Figure 4.20 présente la matrice de confusion obtenue.

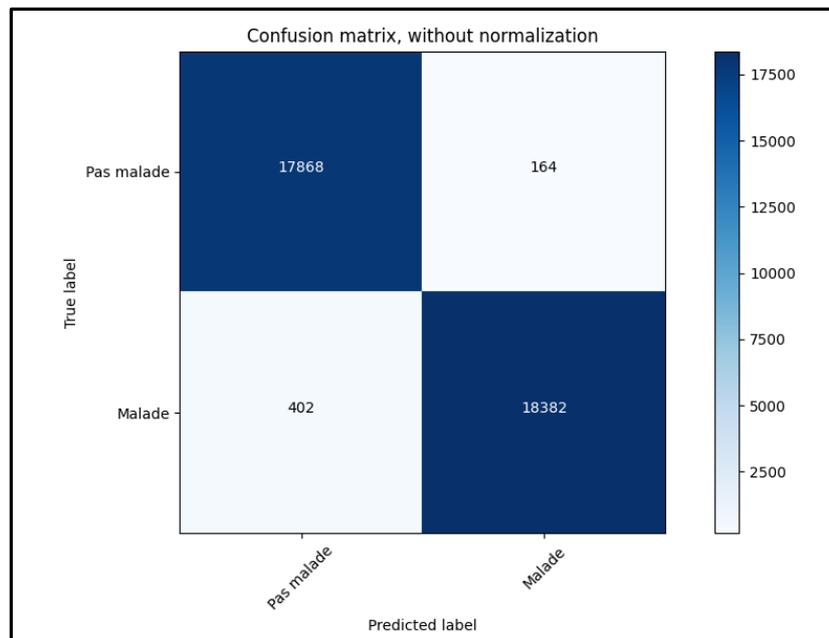


Figure 4.20 : La matrice de confusion CNN Binaire.

➤ **Le rapport de classification :**

Ce dernier est donnée par la Figure 4.21.

	precision	recall	f1-score	support
0	0.98	0.99	0.98	18032
1	0.99	0.98	0.98	18784
accuracy			0.98	36816
macro avg	0.98	0.98	0.98	36816
weighted avg	0.98	0.98	0.98	36816

Figure 4.21 : Résultat CNN Binaire.

Ainsi, pour la classification binaire avec l'algorithme CNN, le taux de précision est de 99 %, le Recall est de 98% et le f1-score est de 98% pour les personnes malades, ce qui est un très bon résultat qui va nous permettre de savoir si la personne est malade ou non.

▪ **Pour la classification multiclasse :**

Nous avons utilisé l'algorithme CNN pour réaliser une classification multiclasse. Les résultats obtenus sont illustrés dans la figure suivante.

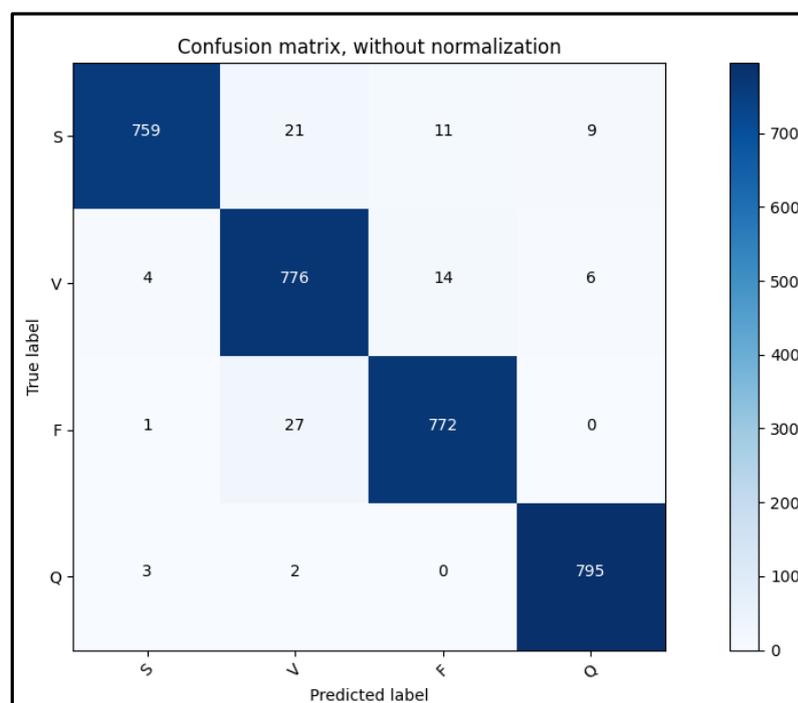


Figure 4.22 : La matrice de confusion CNN Multi-classe.

➤ **Le rapport de classification :**

Il est donné dans la figure ci-dessous.

	precision	recall	f1-score	support
0	0.99	0.95	0.97	800
1	0.94	0.97	0.95	800
2	0.97	0.96	0.97	800
3	0.98	0.99	0.99	800
accuracy			0.97	3200
macro avg	0.97	0.97	0.97	3200
weighted avg	0.97	0.97	0.97	3200

Figure 4.23 : Résultat CNN Multi-classe.

Le rapport de classification indique que Notre modèle de classification multiclasse a une performance globalement élevée, avec de bonnes précisions, recall et F1-scores pour chaque classe. L'Accuracy de ce modèle est de 97%, ce qui signifie qu'il prédit correctement la classe des échantillons dans environ 97% des cas. Ces résultats montrent que le modèle est capable de bien classifier les données

d) Classification multi-classe avec 5 classes :

Dans cette étape, nous utilisons la base de données originale qui comprend 5 classes. Parmi ces classes, 4 correspondent aux différents types de maladies cardiaques, tandis qu'une classe est réservée aux personnes considérées comme normales.

Nous utiliserons l'algorithme CNN pour effectuer ce type de classification. La figure suivante présente les résultats que nous avons obtenus.

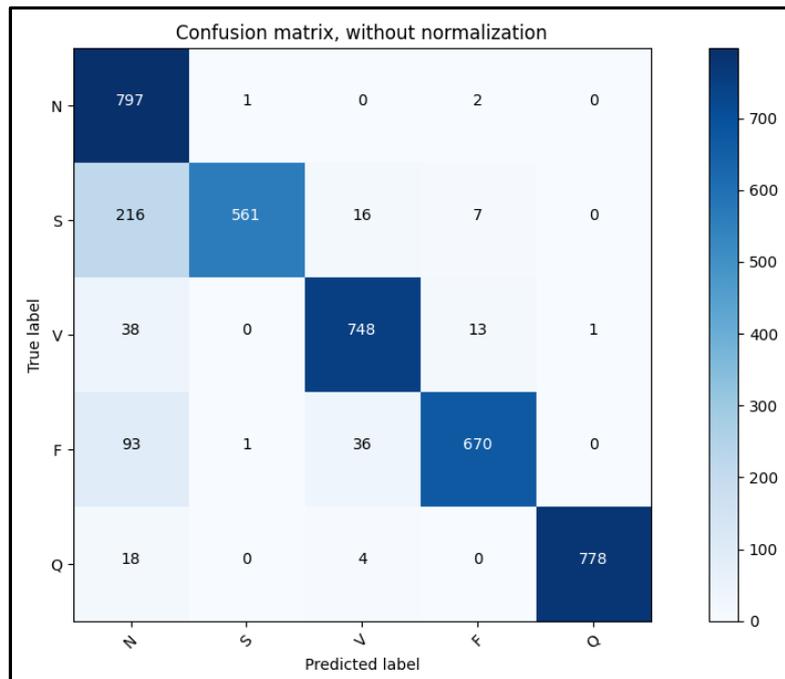


Figure 4.24 : La matrice de confusion CNN Multi.

➤ **Le rapport de classification :**

Il est donné dans la figure ci-dessous.

	precision	recall	f1-score	support
0	0.69	1.00	0.81	800
1	1.00	0.70	0.82	800
2	0.93	0.94	0.93	800
3	0.97	0.84	0.90	800
4	1.00	0.97	0.99	800
accuracy			0.89	4000
macro avg	0.92	0.89	0.89	4000
weighted avg	0.92	0.89	0.89	4000

Figure 4.25 : Résultat CNN Multi.

Selon le rapport de classification, le modèle présente des mesures de précision, de rappel et de F1-score compris entre 69% et 100%, avec une Accuracy de 89%. Ces résultats démontrent la capacité du modèle à effectuer une classification efficace des données.

4.7 Comparaison des algorithmes utilisés

Nous allons comparer le modèle de CNN (Convolutional Neural Network) et le modèle MLP (Multi-Layer Perceptron) afin de déterminer quelle architecture donne de meilleurs résultats pour les deux types de classification (binaire et multiclasse).

4.7.1 Comparaison pour la classification binaire

Le Tableau 4.1 présente la valeurs des différentes métriques obtenues par les 2 algorithmes dans le cas d'une classification binaire.

Algorithmes Métriques	MLP		CNN	
	PAS malade	Malade	PAS malade	Malade
Précision	0.93	0.88	0.98	0.99
Recall	0.98	0.63	0.99	0.98
F1-score	0.95	0.74	0.98	0.98
Accuracy	0.92		0.98	

Tableau 4.1 : Comparaison entre le MLP et le CNN (cas de classification binaire).

Après avoir testé les deux algorithmes de Deep Learning pour la classification binaire, il a été observé que les deux algorithmes donnent de bons résultats. Cependant, le CNN s'est révélé être plus optimal et performant, répondant mieux à notre problématique. Il a atteint une Accuracy de **98%**, tandis que l'algorithme MLP testé a obtenu une Accuracy de **93%**.

Pour la première étape de notre architecture, qui concerne la classification binaire, nous choisirons donc **le CNN**.

4.7.2 Comparaison pour la classification multi-classe

D'après les résultats de la classification multiclasse fournis pour les algorithmes MLP (Multi-Layer Perceptron) et CNN (Convolutional Neural Network) présentées dans le Tableau 4.2, une comparaison des performances et une recommandation sur la meilleure méthode s'impose donc.

Algorithmes	MLP				CNN								
					4 classes				5 classes				
	Métriques	1	2	3	4	1	2	3	4	0	1	2	3
Précision	0.98	0.89	0.95	0.98	0.99	0.94	0.97	0.98	0.69	1	0.93	0.97	1
Recall	0.91	0.95	0.94	0.99	0.95	0.97	0.96	0.99	1	0.70	0.94	0.84	0.97
F1-score	0.95	0.92	0.94	0.98	0.97	0.95	0.97	0.99	0.81	0.82	0.93	0.90	0.99
Accuracy	0.95				0.97				0.89				

Tableau 4.2 : Comparaison pour classification multi-classe

Après avoir testé les deux algorithmes de Deep Learning pour la classification multi-classe, il a été observé que les deux algorithmes donnent de bons résultats.

Cependant, le CNN s'est révélé être plus optimal et performant, répondant mieux à notre problématique. Il a atteint une Accuracy de **97%**, tandis que l'algorithme MLP testé a obtenu une précision de **95%**.

En outre, nous avons également fourni les résultats de la classification CNN pour 5 classes spécifiques. Les mesures de précision, de rappel et de score F1 varient entre 69% et 100%, avec une Accuracy 89%.

En se basant sur ces résultats, il semble que l'algorithme CNN ait obtenu de meilleures performances que l'algorithme MLP, tant en termes de précision globale que de précision par classe.

Par conséquent, la méthode recommandée serait d'utiliser le CNN pour la classification multiclasse, en tenant compte des performances globales et des performances spécifiques aux classes individuelles.

Nous proposons donc le CNN comme le meilleur algorithme pour classer les signaux ECG, qu'il s'agisse de classifications binaires ou multi-classe (Figure 4.26)

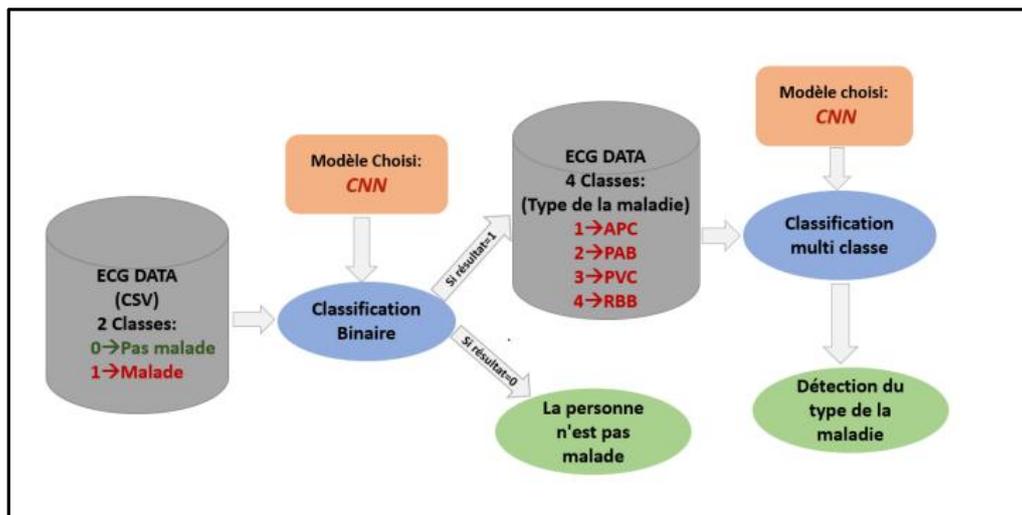


Figure 4.26: L'architecture finale de la classification des arythmies ECG.

4.8 Comparaison entre les méthodes de l'état d'art et celles du système utilisé

En comparant les méthodes utilisées dans l'état de l'art avec celles utilisées dans ce projet, nous avons constaté que les deux approches ont donné des résultats identiques en ce qui concerne la classification des données ECG. Cela renforce encore davantage l'efficacité du CNN pour résoudre ce type de problème.

Il est essentiel de tester plusieurs algorithmes pour une problématique donnée afin de déterminer celui qui convient le mieux. Il n'existe pas de règle générale pour affirmer qu'un algorithme est meilleur qu'un autre.

4.9 Conclusion

Dans ce chapitre, nous avons présenté notre système développé pour la classification des signaux ECG en utilisant deux étapes : la classification binaire et la classification multi-

classe, en utilisant des algorithmes de Deep Learning. Nous avons utilisé spécifiquement deux algorithmes, MLP et CNN, et nous avons affiché et analysé les résultats obtenus pour chaque algorithme.

La comparaison des résultats obtenus a clairement démontré que l'algorithme CNN offre les meilleurs résultats, que ce soit pour la classification binaire ou la classification multi-classe des signaux ECG.

Conclusion Générale



Conclusion Générale

Ce travail présente une approche pour la classification des signaux ECG en deux étapes : une classification binaire pour déterminer si une personne est malade ou non, et une classification multi-classe pour identifier le type spécifique de maladie cardiaque chez les personnes atteintes.

Différents algorithmes de Deep Learning ont été testés, et ceux qui se sont révélés les plus optimaux pour notre problématique ont été sélectionnés. Parmi ces algorithmes, le CNN a obtenu les meilleurs résultats tant pour la classification binaire que pour la classification multi-classe, comparés aux autres méthodes évaluées.

Pour parvenir à ces résultats, une grande quantité de temps a été consacrée à la lecture et à l'étude de publications et d'articles afin de comprendre les meilleures pratiques en matière de classification et de concevoir une approche appropriée. Selon les expériences décrites dans les articles et/ou acquises personnellement, il est clair qu'il est important de tester plusieurs algorithmes afin de déterminer celui qui convient le mieux à une problématique donnée. Il n'existe pas de règle générale pour affirmer qu'un algorithme est supérieur à un autre.

Grâce aux avancées de l'intelligence artificielle, telles que l'apprentissage automatique (Machine Learning) et le Deep Learning, nous sommes en mesure de traiter des problématiques complexes et d'obtenir de bons résultats, ce qui est particulièrement crucial dans le domaine de la santé où la précision peut sauver des vies. L'intelligence artificielle offre de nombreux avantages pour explorer le domaine de la santé, ouvrant la voie à des innovations qui amélioreront les systèmes de santé pour un avenir meilleur.

Notre système a été validé en utilisant des enregistrements extraits de la base de données MIT-BIH, et les résultats obtenus sont très encourageants, démontrant l'efficacité de notre approche.

Afin de permettre aux praticiens d'exploiter notre résultat, nous proposons pour le futur de développer une interface graphique interactive permettant la lecture d'un signal ECG puis l'affichage du type de l'arythmie si le patient en présente le cas.

Bibliographie



Bibliographie

- [1] « Cardiovascular diseases (CVDs) » World Health Organization, 17 May 2017. [En ligne]. Lien : <https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-cvds> (consulté le 25 juin 2023).
- [2] « Cardiologie : Généralités sur le coeur, Rythme cardiaque, Structure Anatomique », Santé sur le Net, l'information médicale au cœur de votre santé. <https://www.sante-sur-le-net.com/maladies/cardiologie/generalites-coeur/> (consulté le 25 juin 2023).
- [3] « Lecture d'un ECG (1) - Blog d'électrocardiographie vétérinaire ». <http://vet-ecg.ddns.net/index.php/lecture-dun-ecg-les-bases/> (consulté le 25 juin 2023).
- [4] L. Ary, et al. « Goldberger: clinical electrocardiography: a simplified approach. », Ed. Elsevier, 2013.
- [5] A .Belgacem, « Classification des signaux ECG avec un système-multi-agent neuronale », Magister en intelligence artificielle et aide à la décision, Université Abou Bekr Belkaid, Tlemcen, Algérie, 2012.
- [6] M .Talbi ,«Analyse et Traitement du Signal Electrocardiographique (ECG) », Doctorat en traitement de signal, Université Mentouri, Constantine, Algérie, 2011.
- [7] « Lecture d'un ECG (1) - Blog d'électrocardiographie vétérinaire ». <http://vet-ecg.ddns.net/index.php/lecture-dun-ecg-les-bases/> (consulté le 25 juin 2023).
- [8] R. BENALI, « Analyse du signal ECG par réseau adaptif d'ondelettes en vue de la reconnaissance de pathologies cardiaques », Doctorat en Sciences ,spécialité Electronique Biomédicale , Université Abou Bekr Belkaid, 2013.
- [9] « Électrocardiographie - L'électrocardiographe ». <https://www.techno-science.net/glossaire-definition/Electrocardiographie-page-2.html> (consulté le 25 juin 2023).
- [10] D. ATTIAS, N. LELLOUCHE « ikb CARDIOLOGIE VASCULAIRE »Edition VG,7e edition 2016.
- [11] L. Dekkiche, « Classification des arythmies ECG avec des méthodes de Machine Learning et de Deep Learning», Mémoire de master, UNIVERSITE MOULOUD MAMMERI DE TIZI-OUZOU, 2019-2020.
- [12] « Appréhendez le Deep Learning ou l'apprentissage profond », OpenClassrooms. <https://openclassrooms.com/fr/courses/6417031-objectif-ia-initiez-vous-a-lintelligence-artificielle/6823506-apprenez-le-deep-learning-ou-lapprentissage-profond> (consulté le 25 juin 2023).
- [13] H. Wang et B. Raj, « On the Origin of Deep Learning ». arXiv, 2 mars 2017. [En ligne]. Disponible sur : <http://arxiv.org/abs/1702.07800>
- [14] « TOP 15 Deep Learning applications that will rule the world in 2018 and beyond ». <https://www.linkedin.com/pulse/top-15-deep-learning-applications-rule-world-2018-beyond-mittal> (consulté le 25 juin 2023).

- [15] V. Zocca, G. Spacagna, D. Slater, et P. Roelants, « Python Deep Learning », Ed. Packt, 2019.
- [16] B. Maurice, « Fonctionnement du neurone artificiel », Deeply Learning, 15 septembre 2018. <https://deeplylearning.fr/cours-theoriques-deep-learning/fonctionnement-du-neurone-artificiel/> (consulté le 25 juin 2023).
- [17] « Figure 2. Activation function of neuron », ResearchGate. https://www.researchgate.net/figure/Activation-function-of-neuron_fig1_329799984 (consulté le 26 juin 2023).
- [18] N. Buduma et N. Locascio, « Fundamentals of Deep Learning: Designing Next-generation Machine Intelligence Algorithms », Ed. O'Reilly Media, 2017.
- [19] D. Slater, et P. Roelants, et al, « Python Deep Learning », Ed. Packt, 2017.
- [20] « What is a sigmoid function in neural networks? », Quora. <https://www.quora.com/What-is-a-sigmoid-function-in-neural-networks> (consulté le 26 juin 2023).
- [21] « A Basic Introduction to Activation Function in Deep Learning ». <https://www.analyticsvidhya.com/blog/2022/03/a-basic-introduction-to-activation-function-in-deep-learning/> (consulté le 26 juin 2023).
- [22] S. Raschka, « What is Softmax regression and how is it related to Logistic regression? », SebastianRaschka, *PhD*. https://sebastianraschka.com/faq/docs/softmax_regression.html (consulté le 26 juin 2023).
- [23] « Difference Between Softmax Function and Sigmoid Function - Dataaspirant », 7 mars 2017. <https://dataaspirant.com/difference-between-softmax-function-and-sigmoid-function/> (consulté le 26 juin 2023).
- [24] « Multilayer Perceptron Definition | DeepAI ». <https://deepai.org/machine-learning-glossary-and-terms/multilayer-perceptron> (consulté le 26 juin 2023).
- [25] R.B. Zadeh, B. Ramsundar, « TensorFlow for Deep Learning », Ed. O'Reilly Media, 2017
- [26] « CS231n Convolutional Neural Networks for Visual Recognition ». <https://cs231n.github.io/convolutional-networks/> (consulté le 26 juin 2023).
- [27] « Basic CNN Architecture: Explaining 5 Layers of Convolutional Neural Network | upGrad blog ». <https://www.upgrad.com/blog/basic-cnn-architecture/> (consulté le 26 juin 2023).
- [28] « Réseaux récurrents et LSTM - ppt télécharger ». <https://slideplayer.fr/slide/13237015/> (consulté le 27 juin 2023).
- [29] « LSTM RNN in Tensorflow - Javatpoint ». <https://www.javatpoint.com/long-short-term-memory-rnn-in-tensorflow> (consulté le 27 juin 2023).
- [30] « The Neural Network Zoo - The Asimov Institute ». <https://www.asimovinstitute.org/neural-network-zoo/> (consulté le 27 juin 2023).

- [31] J. Heaton, « Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning », Ed. The MIT Press, 2016.
- [32] D.P. Kingma et J. Ba, « Adam: A Method for Stochastic Optimization ». 3rd International Conference for Learning Representations, San Diego, 2015.
- [33] T. J. Jun, H. M. Nguyen et al. « ECG arrhythmia classification using a 2-D convolutional neural network »,2018.
- [34] G. B. Moody et R. G. Mark, « The impact of the MIT-BIH Arrhythmia Database », IEEE Engineering in Medicine and Biology Magazine , 2001.
- [35] O. Russakovsky et al. « ImageNet Large Scale Visual Recognition Challenge », Int J Comput Vis, 2015.
- [36] A. Krizhevsky, I. Sutskever, et G. E. Hinton, « ImageNet Classification with Deep Convolutional Neural Networks », in Advances in Neural Information Processing Systems, Curran Associates,2012.
- [37] C. Szegedy et al. « Going Deeper With Convolutions », IEEE Conference on Computer Vision and Pattern Recognition, 2015.
- [38] K. Simonyan et A. Zisserman, « Very Deep Convolutional Networks for Large-Scale Image Recognition »,2015.
- [39] K. He, X. Zhang, et al. « Deep Residual Learning for Image Recognition », IEEE Conference on Computer Vision and Pattern Recognition , 2016.
- [40] G. Huang, Z. Liu et al. « Densely Connected Convolutional Networks »,IEEE Conference on Computer Vision and Pattern Recognition ,2017.
- [41] S. Singh, S. K. Pandey, et al. « Classification of ECG Arrhythmia using Recurrent Neural Networks », Procedia Computer Science, 2018.
- [42] H. Lee, R. Grosse, et al.« Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations », in Proceedings of the 26th Annual International Conference on Machine Learning, Association for Computing Machinery, 2009.
- [43] U. R. Acharya, H. Fujita et al.« Application of deep convolutional neural network for automated detection of myocardial infarction using ECG signals », Information Sciences, 2017.

- [44] A.L. Goldberger, L.A.N. Amaral, et al, « PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals», 2000.
- [45] B. N. Singh et A. K. Tiwari, « Optimal selection of wavelet basis function applied to ECG signal denoising », Digital Signal Processing, 2006.
- [46] J. Pan et W. J. Tompkins, « A Real-Time QRS Detection Algorithm », IEEE Transactions on Biomedical Engineering, 1985.
- [47] M. Kachuee, S. Fazeli, et al. « ECG Heartbeat Classification: A Deep Transferable Representation », IEEE International Conference on Healthcare Informatics, 2018.
- [48] A. L. Goldberger, L. A. Amaral, et al. « Physiobank, physiotoolkit, and physionet, Circulation», 2000.
- [49] T. Li et M. Zhou, « ECG Classification Using Wavelet Packet Entropy and Random Forests », 2016.
- [50] G. B. Moody et R. G. Mark, « The impact of the MIT-BIH Arrhythmia Database », IEEE Engineering in Medicine and Biology Magazine, 2001.
- [51] C.Ye, M. T. Coimbra, et al. « Arrhythmia detection and classification using morphological and dynamic features of ECG signals », Annual International Conference of the IEEE Engineering in Medicine and Biology, 2010.
- [52] S.N. Yu et K.-T. Chou, « Integration of independent component analysis and neural networks for ECG beat classification », Expert Systems with Applications, 2008.
- [53] R. J. Martis, U. R. Acharya, et al.« Application of Higher Order Cumulant Features for Cardiac Health Diagnosis Using ECG Signals», 2013.
- [54] R. Mark and G. Moody, «MIT-BIH Arrhythmia Database». Lien: <http://ecg.mit.edu/dbinfo.html>.
- [55] M. Hendel, A.benyettou, et al. « Classification des Arythmies Cardiaques par les Réseau de Neurones Artificiels ». Université des Sciences et de Technologie d'Oran Département d'informatique, 2009.
- [56] T. Kohonen, «Self-Organizing Maps», Springer Series in Information Sciences, Berlin, Heidelberg: Springer, 2nd edition. 1997.

- [57] R. Ceylan et Y. Özbay, « Comparison of FCM, PCA and WT techniques for classification ECG arrhythmias using artificial neural network », Expert Systems with Applications, ,2007
- [58] Sung-Nien Yu, Ying-Hsiang Chen, « Electrocardiogram beat classification based on wavelet transformation and probabilistic neural network », Pattern Recognition Letters ,2007.
- [59] M. Hendel, A. Benyettou, et al. « Classification automatique des signaux ECG par les réseaux de neurones probabilistes », Applications Médicales de l'Informatique : Nouvelles Approches, 2008.
- [60] R. Wariar and D. Moreau. «Integer coefficient bandpass filter for the simultaneous removal of baseline wander», Med. & Biol. Eng. & Comput, 1999.
- [61] F. Alexandre, N. Kerkeni et al. «Supervised Neural networks approaches for EEG signal classification : experimental studies», From Proceeding (544) Artificial Intelligence and Soft Computing,2006.
- [62] « ECG Heartbeat Categorization Dataset ». <https://www.kaggle.com/datasets/shayanfazeli/heartbeat> (consulté le 15 Mars 2023).
- [63] « Matrice de confusion: comment la lire et l'interpréter | Jedha ». <https://www.jedha.co/formation-ia/matrice-confusion> (consulté le 30 juin 2023).
- [64] « Confusion Matrix: How To Use It & Interpret Results [Examples] ». <https://www.v7labs.com/blog/confusion-matrix-guide> (consulté le 30 juin 2023).
- [65] U. T. SL, « Visual Studio Code (Windows) », Uptodown. <https://visual-studio-code.fr.uptodown.com/windows> (consulté le 19 juin 2023).
- [66] « Qu'est-ce que Python? – Le langage Python expliqué – AWS », Amazon Web Services. <https://aws.amazon.com/fr/what-is/python/> (consulté le 30 juin 2023).
- [67] « The Python Logo », Python.org. <https://www.python.org/community/logos/> (consulté le 30 juin 2023).
- [68] « TensorFlow ». <https://www.tensorflow.org/?hl=fr> (consulté le 30 juin 2023).

- [69] « TensorFlow logo vector ». <https://brandeps.com/logo/T/TensorFlow-01> (consulté le 30 juin 2023).
- [70] « Keras : une bibliothèque open source pour la constitution de réseaux neuronaux », IONOS Digital Guide. <https://www.ionos.fr/digitalguide/web-marketing/search-engine-marketing/quest-ce-que-keras/> (consulté le 30 juin 2023).
- [71] « logo keras », Pensée Artificielle. <https://penseeartificielle.fr/tp-reconnaissance-faciale/logo-keras/> (consulté le 30 juin 2023).
- [72] J. JVC, « Scikit-Learn : guide de démarrage rapide en Machine Learning avec Python », Data Transition Numérique.<https://www.data-transitionnumerique.com/scikit-learn-python/> (consulté le 30 juin 2023).
- [73] Haider,« Scikit learn Logo Vector »,VectorSeek. https://vectorseek.com/vector_logo/scikit-learn-logo-vector/ (consulté le 30 juin 2023).