

République Algérienne Démocratique Et Populaire  
Ministère De l'Enseignement Supérieure Et De La Recherche Scientifique  
Université Abderahmane Mira - Bejaia -  
Faculté Des Sciences Exactes  
Département d'Informatique



*Mémoire de fin d'études en vue de l'obtention d'un diplôme de master en  
informatique*

## thème

Développement d'un système de détection d'anomalies  
des rythmes cardiaques en utilisant le deep learning

Réalisé par :

M<sup>elle</sup> ABBAS Lydia

M<sup>elle</sup> HAMMOUCHE Nesrine

Membres de jury :

Présidente	M <sup>me</sup> El BOUHISSI Houda Épouse BRAHAMI	M.C.A
Examinatrice	M <sup>me</sup> TAHAKOURT Zineb	M.A.A
Encadrant	M <sup>r</sup> AMROUN Kamal	M.C.A

promotion 2019/2020

## *- Remerciements -*

Nous remercions tout d'abord Dieu le tout puissant de nous avoir donné la volonté et la persévérance pour réaliser ce travail. Nous tenons à saisir cette occasion et adresser nos profonds remerciements et nos profondes reconnaissances à monsieur AMROUN Kamal, notre promoteur, pour ses précieux conseils et son orientation ficelée tout au long de notre recherche. Nos vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail Et de l'enrichir par leurs propositions. Nous souhaitons adresser nos remerciements les plus sincères aux personnes qui nous ont apporté leur aide et qui ont contribué à l'élaboration de ce mémoire ainsi qu'à la réussite de cette formidable année universitaire. Nous remercions tous nos enseignants pour leur dévouement, leur patience et leur contribution à notre formation. Enfin, nous adressons nos plus sincères remerciements à tous nos proches et amis, qui nous ont toujours encouragés au cours de la réalisation de ce mémoire.

## *- Dedicaces -*

Je dédie ce travail aux deux êtres les plus chères au monde, les bougies qui m'ont toujours guidé vers le bon chemin : Ma mère, la lumière de ma vie, qui a tout fait pour ma réussite et mon bonheur. Mon père, à lui, je devais offrir tout le respect et l'amour pour son soutien et sa tendresse.

A mes frères Fahem, kamel et Lyes et a mes sœur Fadila, Safia, Nassima, Nora et Souad qui m'ont encouragé sans cesse et cru en moi.

A mon mari Billal qui a été toujours près de moi et m'a soutenu tout au long de mon projet.

A mes beaux-parents.

A tous les membres de la famille et à tous les amis en particulier **Samia et Reda**.

**-ABBAS Lydia-**

## - *Dedicaces* -

Je dédie ce mémoire :  
À MES CHERS PARENTS

Aucune dédicace ne saurait exprimer mon respect, mon amour éternel et ma considération pour les sacrifices que vous avez consenti pour mon instruction et mon bien être Je vous remercie pour tout le soutien et l'amour que vous me portez depuis mon enfance et j'espère que votre bénédiction m'accompagne toujours .Que ce modeste travail soit l'exaucement de vos vœux tant formulés, le fruit de vos innombrables sacrifices, bien que je ne vous en acquitterai jamais assez .Puisse Dieu, le Très Haut, vous accorder santé, bonheur et longue vie et faire en sorte que jamais je ne vous déçoive . Je voudrais réserver une place spéciale dans mon cœur pour remercier ma mère qui m'a toujours soutenue moralement et financièrement, de manière inconditionnelle, et qui ne cesse de m'encourager pour aller toujours de l'avant ,Affable, honorable, aimable : Tu représentes pour moi le symbole de la bonté par excellence, la source de tendresse et l'exemple du dévouement qui n'a pas cessé de m'encourager et de prier pour moi. Ta prière et ta bénédiction m'ont été d'un grand secours pour mener à bien mes études. Aucune dédicace ne saurait être assez éloquente pour exprimer ce que tu mérites pour tous les sacrifices que tu n'as cessé de me donner depuis ma naissance, durant mon enfance et même à l'âge adulte. Tu as fait plus qu'une mère puisse faire pour que ses enfants suivent le bon chemin dans leur vie et leurs études. Je te dédie ce travail en témoignage de mon profond amour. Puisse Dieu, le tout puissant, te préserver et t'accorder santé, longue vie et bonheur.

A mes très chers frères Dada Salah , Dada Kamel et Dada Djamel.

Mes conseiller, et amis fidèles, qui m'ont assisté dans les moments difficiles et m'ont pris doucement par la main pour traverser ensemble des épreuves pénibles. . . .Je te suis très reconnaissante, et je vous remercierai jamais assez pour votre amabilité, votre générosité, votre aide précieuse.

A mes très chères sœurs Nana Maissa et Nana Mounia.

En témoignage de l'attachement, de l'amour et de l'affection que je porte pour toi Nana Maissa Malgré la distance, tu es toujours dans mon cœur. Je te remercie pour ton hospitalité sans égal et ton affection si sincère. Je vous dédie ce travail avec tous mes vœux de bonheur, de santé et de réussite et Aucune dédicace ne peut exprimer mon amour et ma gratitude de t'avoir comme sœur Nana Mounia , Je ne pourrais jamais imaginer la vie sans toi, tu comptes énormément pour moi, tu es la sœur qui assure son rôle comme il faut, je n'oublierais jamais ton encouragement et ton soutien le long de mes études, je t'estime beaucoup et je t'aime beaucoup .Je te souhaite beaucoup de succès, de prospérité et une vie pleine de joie et de bonheur.

A mon très Cher neveu Fatah Baba-aissa Avoir un neveu est le plus beau cadeau qu'une sœur puisse vous faire. ton enthousiasme, tes sourires, tes yeux brillants sont incomparables. Tu as apporté beaucoup de bonheur à notre famille. Je t'aime

---

À MES AUTRES CHERS PETITS NEVEUX ET NIECES Ghania , Lamiss, Didouche, Mousline  
 , kenzi, hanna , ines, sofia,imene,mordjane ,aylan,émelle,éline.

Aucune dédicace ne saurait exprimer tout l'amour que j'ai pour vous, Votre joie et votre gaieté  
me comblent de bonheur. Puisse Dieu vous garder, éclairer votre route et vous aider à réaliser à  
votre tour vos vœux les plus chers.

**-HAMMOUCHE Nesrine-**

# Table des matières

Table des matières	i
liste des figures	iii
liste des tableaux	v
liste des abréviations	vi
Introduction générale	1
<b>1 L'Electrocardiogramme</b>	<b>3</b>
1.1 Introduction . . . . .	3
1.2 Le rôle du cœur . . . . .	3
1.3 Structure et rôle du coeur . . . . .	5
1.4 L'arythmie . . . . .	5
1.4.1 Les différents types d'arythmie . . . . .	6
1.4.1.1 Bradycardies . . . . .	6
1.4.1.2 La tachycardies . . . . .	6
1.5 L'Electrocardiographie . . . . .	7
1.5.1 But de l'Electrocardiogramme . . . . .	9
1.5.2 Les ondes du signal ECG . . . . .	10
1.5.3 Les segments et intervalles du signal ECG . . . . .	10
1.5.3.1 Les dérivations d'un ECG . . . . .	12
1.5.3.2 Interprétation de l'ECG . . . . .	13
1.6 Conclusion . . . . .	15
<b>2 Techniques de l'intelligence artificielle</b>	<b>16</b>
2.1 Introduction . . . . .	16
2.2 L'Intelligence artificielle . . . . .	16
2.3 techniques d'intelligence artificielle . . . . .	17
2.3.1 Le Machine Learning – Apprentissage automatique . . . . .	17
2.3.1.1 Quelques types de Machine Learning . . . . .	17
2.3.1.2 Quelques algorithmes du Machine Learning . . . . .	18

2.3.2	Le Deep Learning – Apprentissage profond . . . . .	21
2.3.3	Les réseaux de neurones . . . . .	21
2.3.4	Domaine d’application du deep learning . . . . .	25
2.3.5	Différences entre Deep Learning et Machine Learning . . . . .	25
2.3.6	Avantages des réseaux de neurones . . . . .	26
2.3.7	Inconvénients des réseaux de neurones . . . . .	26
2.4	Medecine et IA . . . . .	27
2.5	Les domaines les plus impactés par l’IA . . . . .	27
2.6	Conclusion . . . . .	28
<b>3</b>	<b>Deep learning pour la détection d’anomalies</b>	<b>29</b>
3.1	Introduction . . . . .	29
3.2	Etat d’art . . . . .	29
3.2.1	Analyse et critique . . . . .	30
3.3	CNN . . . . .	30
3.3.1	L’entraînement d’un nouveau CNN . . . . .	34
3.4	LSTM . . . . .	37
3.4.1	RNN (recurrent neural networks) . . . . .	37
3.4.1.1	Le problème La descente de gradient . . . . .	38
3.5	CNN pour la détection d’anomalies . . . . .	44
3.6	LSTM pour la détection d’anomalies . . . . .	44
3.7	Conclusion . . . . .	45
<b>4</b>	<b>Expérimentation</b>	<b>47</b>
4.1	Introduction . . . . .	47
4.2	Présentation des outils . . . . .	48
4.2.1	Outils logiciels . . . . .	48
4.2.1.1	Environnement . . . . .	48
4.2.1.2	Framework . . . . .	48
4.2.1.3	Bibliothèques utilisées . . . . .	48
4.2.1.4	Langages de programmation . . . . .	49
4.3	Classification avec Réseau Convolutif . . . . .	49
4.4	La base de données MIT-BIH Arrhythmia . . . . .	50
4.5	Techniques d’échenillage . . . . .	51
4.6	Fractionnement de données . . . . .	51
4.7	Modèle CNN . . . . .	52
4.7.1	Phase d’entrainements . . . . .	53
4.7.2	Phase de test . . . . .	55
4.7.3	Critère d’évaluation . . . . .	56
4.8	Modèle LSTM sous Tensorflow . . . . .	58
4.8.1	Phase d’entraînement avec le modèle LSTM . . . . .	59

4.9	Limites . . . . .	61
4.10	Tensorflow et pytorch . . . . .	61
4.11	Comparaison entre les 2 modèles . . . . .	62
4.12	Conclusion . . . . .	63
	<b>Conclusion générale</b>	<b>64</b>



# Table des figures

1.1	Coeur [3]. . . . .	4
1.2	Les composants du coeur [4]. . . . .	5
1.3	Rythme cardiaque [8]. . . . .	7
1.4	Willem Einthofen, prix Nobel pour l'invention de dispositif ECG [6]. . . . .	9
1.5	Ecg Normal [6]. . . . .	9
1.6	Les ondes d'un ECG [7]. . . . .	10
1.7	Segment et intervalles de L'ECG [8]. . . . .	11
1.8	Les intervalles RR,PR,QRS,QT respectivement. . . . .	12
1.9	Interprétation de L'ECG [12]. . . . .	15
2.1	Processus d'apprentissage automatique [13]. . . . .	17
2.2	Nuage de point des valeurs prédites et observées [17]. . . . .	19
2.3	Affectation d'un nouvel élément x à une classe par K-NN [17]. . . . .	19
2.4	Exemple d'arbre de décision [17] . . . . .	20
2.5	La relation entre l'intelligence artificielle, le machine Learning et le deep Learning [19]. . . . .	21
2.6	Réseau de neurone simple [20]. . . . .	22
2.7	Famille de réseau de neurones [27]. . . . .	22
2.8	Schéma du perceptron simple [26]. . . . .	23
2.9	Réseaux du perceptron multicouche [28]. . . . .	24
2.10	Machine Boltzmann restreinte [22]. . . . .	25
2.11	Comparaison entre la machine Learning et le Deep Learning [22]. . . . .	26
2.12	Schéma de fonctionnement d'un système expert [25]. . . . .	28
3.1	Architecture de CNN . . . . .	31
3.2	Feature Map [30] . . . . .	32
3.3	Fonction Relu [31] . . . . .	33
3.4	Max pooling [33] . . . . .	34
3.5	Implémentation du code de CNN . . . . .	36
3.6	Architecture du RNN [36] . . . . .	38
3.7	Schéma d'un neurone LSTM comprenant une mémoire interne contrôlée par les trois portes « Schéma issu de la thèse de Bruno Stuner (2018) » [43] . . . . .	41

---

3.8	Exemple d'une implémentation de LSTM . . . . .	42
3.9	Régularisation Dropout dans le deep learning [43] . . . . .	43
3.10	Classification des erreurs avec ou sans Dropout [43] . . . . .	43
3.11	Architecture du modèle en utilisant le CNN . . . . .	44
3.12	Architecture du modèle en utilisant LSTM . . . . .	45
4.1	Processus de traitement et de préparation des enregistrements. . . . .	50
4.2	Échantillon tracé à partir de la base de données MIT-BIH[51]. . . . .	52
4.3	Fractionnement de données de train et test. . . . .	52
4.4	Signale autour de l'un des battements anormaux. . . . .	53
4.5	API séquentielle et API fonctionnelle. . . . .	54
4.6	Entraînement du modèle. . . . .	54
4.7	Précision et perte tout au long de l'exécution du modèle. . . . .	55
4.8	Implémentation de la phase de Test . . . . .	56
4.9	Précision du modèle. . . . .	57
4.10	Modèle de perte. . . . .	57
4.11	Formation du modèle avec LSTM. . . . .	58
4.12	Préparation des entrées pour l'entraînement du modèle. . . . .	59
4.13	Entraînement du modèle. . . . .	60
4.14	Précision et perte tout au long de l'exécution du modèle LSTM. . . . .	60
4.15	évaluation du modèle. . . . .	61
4.16	Courbe ROC . . . . .	62

# Liste des tableaux

4.1	Types et occurrences des battements cardiaques dans le dataset MIT-BIH. . . . .	51
4.2	Comparaison entre les résultats des deux modèles. . . . .	63

# Liste des abréviations

<b>IA</b>	<b>I</b> ntelligence <b>A</b> rtificielle
<b>ECG</b>	<b>E</b> lectrocardiogramme
<b>CNN</b>	<b>C</b> onvolutional <b>N</b> eural <b>N</b> etwork
<b>ANN</b>	<b>A</b> rtificial <b>N</b> eural <b>N</b> etwork
<b>ML</b>	<b>M</b> achine <b>L</b> earning
<b>DL</b>	<b>D</b> eep <b>L</b> earning
<b>SVM</b>	<b>S</b> upport <b>V</b> ector <b>M</b> achine
<b>ReLU</b>	<b>R</b> ectified <b>L</b> inear <b>U</b> nit
<b>RNN</b>	<b>R</b> eccurent <b>N</b> eural <b>N</b> etwork
<b>LSTM</b>	<b>L</b> ong <b>S</b> hort <b>T</b> erm <b>M</b> emory
<b>WFDB</b>	<b>W</b> aveform-database
<b>TPR</b>	<b>T</b> rue <b>P</b> ositive <b>R</b> ate
<b>TNR</b>	<b>T</b> rue <b>N</b> egative <b>R</b> ate

# Introduction générale

Les maladies cardio-vasculaires sont responsables de la moitié des décès dans le monde, mais encore plus les personnes qui souffrent d'hypercholestérolémie, de diabète, d'hypertension, ou de stress, le cœur est une pompe automatique qui ne s'arrête jamais de fonctionner les experts ont disposés un moyen pour aboutir et interprété les maladies cardiaques, les contractions cardiaques sont régulées par une activité électrique qui peut être enregistrée sur un électrocardiogramme (ECG). L'électrocardiogramme est une image de la conduction électrique du cœur, l'outil pour évaluer l'anomalies cardiaques en se basant sur la fréquence et les battements cardiaques, C'est une représentation graphique effectué par le médecin, ECG est devenu rapidement l'examen de première intention dans le cadre de l'urgence cardiovasculaire pour établir un diagnostic et une prise en charge des syndromes coronariens aigus et des arythmies mais aussi dans le cadre du dépistage ou du suivi de pathologies chroniques. Actuellement, il existe de nombreuses solutions d'intelligence artificielle ou IA qui peuvent être utilisées pour analyser et classer les données ECG. L'Intelligence Artificielle est un vaste domaine qui croise plusieurs techniques dont le but est de doter un système informatique de capacités de réflexion similaires à celles des humains, en d'autres termes, effectuer des tâches telles que raisonner, planifier, apprendre et comprendre notre langage. Dans notre projet nous allons utiliser l'une des meilleurs méthodes d'intelligence artificielle, qui est le deep learning, littéralement appelé apprentissage profond (DL) qui est un domaine de recherche sur l'apprentissage automatique basé sur un type particulier de mécanisme, offre des algorithmes très utiles pour faire face à l'apprentissage à partir de grande quantité de données, DL peuvent atteindre un niveau de précision exceptionnel, parfois supérieur aux performances humaines, il est récemment devenu un sujet d'étude dans différentes applications, y compris la santé, dans laquelle la détection rapide des anomalies sur l'électrocardiogramme (ECG) peut jouer un rôle essentiel dans la surveillance des patients.

## **Problématique**

La détection manuelle des ondes caractéristiques du signal ECG et la classification des battements cardiaques sont des tâches difficiles et ennuyeuses surtout pour l'analyse des enregistrements de longue durée. Par conséquent, les systèmes automatiques d'analyse du signal ECG, capables d'aider les médecins à faire le diagnostic, semble indispensable en raison du grand nombre de patients dans les unités de soins intensifs et de la nécessité d'une observation continue.

A cet effet, notre problématique consiste à apporter une solution qui peut aider les médecins dans l'analyse et l'interprétation d'ECG.

**Objectif**

L'objectif de ce travail est de proposer une méthode bien précise et facile pour l'interprétation des anomalies de rythmes cardiaques en utilisant le deep learning.

**Structure du mémoire**

Le mémoire est structuré comme suit :

- **Chapitre 1 :**

Le premier chapitre présente des notions générales sur le cœur, les anomalies pouvant toucher le cœur et sur le diagramme ECG.

- **Chapitre 2 :**

Le second chapitre est une description de quelques techniques de l'intelligence artificielle.

- **Chapitre 3 :**

Le troisième chapitre est consacré à la présentation des deux modèles de deep learning utilisée (CNN, LSTM).

- **Chapitre 4 :**

Le quatrième chapitre est une description des différents outils utilisés et les résultats obtenus.

Nous terminerons par une conclusion générale.

# Chapitre 1

## L'Electrocardiogramme

### 1.1 Introduction

L'électrocardiogramme (ECG) est l'enregistrement d'une activité électrique du cœur, un test médical qui détecte une anomalie Cardiaque par les électrodes fixées au surface de la peau Il peut aider à enregistrer et détecter les anomalies électrolytiques.

Le traitement du signal est aujourd'hui effectué dans la grande majorité des systèmes d'analyse et d'interprétation ECG son objectif est multiple et comprend l'amélioration et de la précision. L'ECG est enregistré dans des conditions ambulatoires ou épuisantes telles que le signal est corrompu par différents types de bruit, Dans ce chapitre, nous allons aborder en détail le signal ECG avec ces différentes ondes, segments Aussi, les différentes dérivations permettant le recueil du signal ECG seront étudiées [1][2].

### 1.2 Le rôle du cœur

Le cœur est un organe musculaire strié et creux assure la circulation sanguine dans l'organisme. En pompant le sang vers les vaisseaux sanguins et les cavités du corps à travers des contractions rythmique est composé d'un muscle cardiaque spécialisé, qui est strié et organisé en sarcomères pompe du sang vers tout le corps se déroule de manière synchronisée [3].

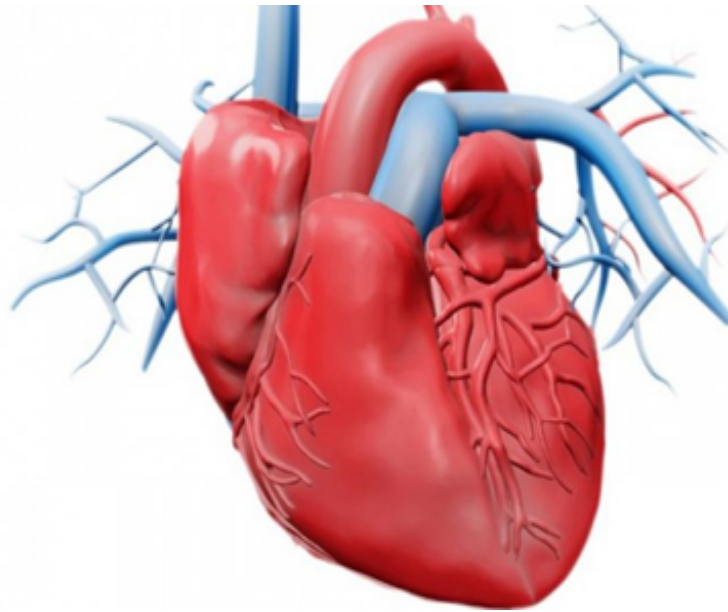


FIGURE 1.1 – Coeur [3].

Le cœur se compose de deux parties. La partie droite qui assure la récupération du sang veineux et la partie gauche qui propulse le sang dans tout l'organisme grâce à l'aorte et comme il est constitué de deux petites cavités supérieures du cœur (oreillettes), et de deux grandes cavités inférieures du cœur (ventricules) [4].

**L'oreillette droite :** reçoit le sang appauvri en oxygène qui a circulé dans le corps (des muscles, des organes, du cerveau et même du cœur). Lorsque l'oreillette est pleine elle se contracte et le sang entre alors dans le ventricule droit [4].

**Le ventricule droit :** Lorsque le ventricule droit est plein, il se contracte à son tour pour l'envoi du sang dans les poumons par la valvule pulmonaire [4].

**L'oreillette gauche :** reçoit le sang qui a été oxygéné dans les poumons véhiculé par les veines pulmonaires.

**Le ventricule gauche :** envoie le sang dans tout l'organisme par l'intermédiaire de l'aorte, la plus grosse artère de l'organisme [4].



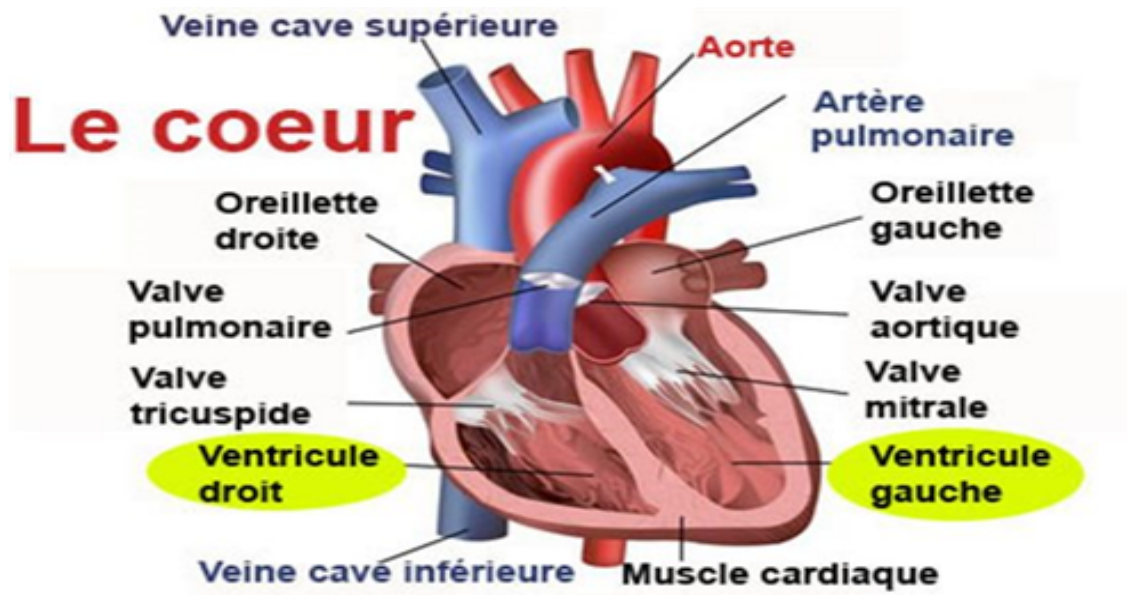


FIGURE 1.2 – Les composants du coeur [4].

### 1.3 Structure et rôle du coeur

Le cœur est formé par un tissu musculaire spécial appelé le myocarde et un muscle de couche épaisse composé de fibres musculaires striées. Le myocarde muscle spéciale du point de vue histologique et physiologique, les fibres musculaires striées du myocarde ne sont pas indépendantes les unes des autres comme c'est le cas des autres muscles striés du corps. Elles sont rattachées les unes aux autres et forment un réseau appelé un Syncytium et c'est le seul muscle strié de l'organisme qui échappe au contrôle de la volonté (Fonctionnement autonome) L'endocarde tapisse les cavités du coeur, il s'agit d'une couche fine et interne et le péricarde enveloppe extérieur du cœur [5].

### 1.4 L'arythmie

L'arythmie est le trouble cardiaque le plus fréquent. Un rythme cardiaque régulier est normal lorsqu'il est compris en journée entre 60 et 100 bpm, et entre 40 et 80 bpm pendant la nuit. Hors de ces limites, on parle de bradycardie lorsqu'il est trop lent, et de tachycardie lorsqu'il est trop rapide. Il est aussi normal que le nombre de pulsations cardiaques s'accélère en réponse à un effort physique, par exemple. Une arythmie cardiaque se produit lorsque le cœur bat irrégulièrement, sans que cela soit justifié.

Dans un coeur arythmique, les impulsions électriques qui contrôlent les battements du coeur se produisent de façon désordonnée ou ne passent pas par les circuits électriques habituels [6].

### 1.4.1 Les différents types d'arythmie

Les arythmies sont classées selon le lieu où elles prennent naissance, l'oreillette ou le ventricule et selon l'effet qu'elles produisent, soit l'accélération ou le ralentissement des battements du coeur [8].

#### 1.4.1.1 Bradycardies

Bradycardies à une diminution lorsque le rythme cardiaque est plus lent est caractérisée par le fait que la fréquence cardiaque est inférieure à 60 pulsations par minute ; elle est dite ventriculaire, bradycardies ne met pas nécessairement la vie en danger.

- la bradycardie sinusale :

La bradycardie sinusale est caractérisée par la présence systématique d'une onde P avant les complexes QRS. Les causes d'une telle arythmie sont multiples et souvent extérieures au système cardiovasculaire : traitement médicamenteux (bêtabloquant, digoxyne, ...).

- Bradycardie d'origine jonctionnelle :

En cas de dysfonctionnement sinusal, le nœud AV peut assurer le rôle de pacemaker de secours à une fréquence de l'ordre de 30 à 60 bpm. L'impulsion électrique prend alors naissance dans le nœud auriculo-ventriculaire (AV) et suit le chemin de conduction habituel : la morphologie des ondes QRS est identique à celle du battement normal ; en revanche, l'onde P peut être absente.

#### 1.4.1.2 La tachycardies

Les tachycardies À l'inverse de la bradycardie, correspondent à une augmentation du rythme cardiaque lorsque le coeur bat à un rythme supérieur à 100 pulsations à la minute, elle peut être d'origine sinusaleII, auriculaireII ou ventriculaire [8].

- La Tachycardie Sinusale :

La tachycardie sinusale correspond à un rythme sinusal dont la fréquence est comprise entre 100 et 180 battement pulsations par minute, on observe, avant chaque complexe QRS, une onde P de forme identique à celle observée lors du rythme normal. Les causes de cette pathologie sont habituellement extra-cardiaques, et incluent tous les facteurs de stimulations du système nerveux,l'influence de différentes substances comme ,m mmml'adrénaline ou la caféine.[8]

- Tachycardie auriculaire et nodale AV :

Tachycardie auriculaire peut avoir pour origine un foyer ectopique, ou une voie qui court-

circuite, il s'agit d'un groupe de cellules situées dans les oreillettes, qui se dépolarisent spontanément et plus rapidement que le sinus, prenant ainsi sa place. La dépolarisation des oreillettes n'étant pas d'origine sinusale, on observe une onde P de forme inhabituelle. La fréquence typique de décharge de ce type de foyer est comprise entre 120 et 200 ; en l'absence de problèmes de conduction AV, les ventricules sont entraînés au même rythme la tachycardie peut aussi être causé par La décharge régulière d'un foyer ectopique localisé dans le nœud AV, appelée tachycardie nodale AV ; dans ce cas, la fréquence des battements peut atteindre 250 bpm. Le principal risque de ce type de pathologie est le manque d'efficacité des ventricules qui, contraints de se contracter très fréquemment, n'ont pas le temps de se remplir correctement de sang : l'alimentation du corps en oxygène peut en être altérée [8].

- Tachycardie ventriculaire (TV) :

La tachycardie ventriculaire a pour origine un ou plusieurs foyer(s) ectopique(s) ventriculaire(s). Les battements ont donc la forme d'extrasystoles ventriculaires très rapprochées. Ce type de rythme est dangereux à cause de sa possible évolution en fibrillation ventriculaire qui conduit au décès[8].

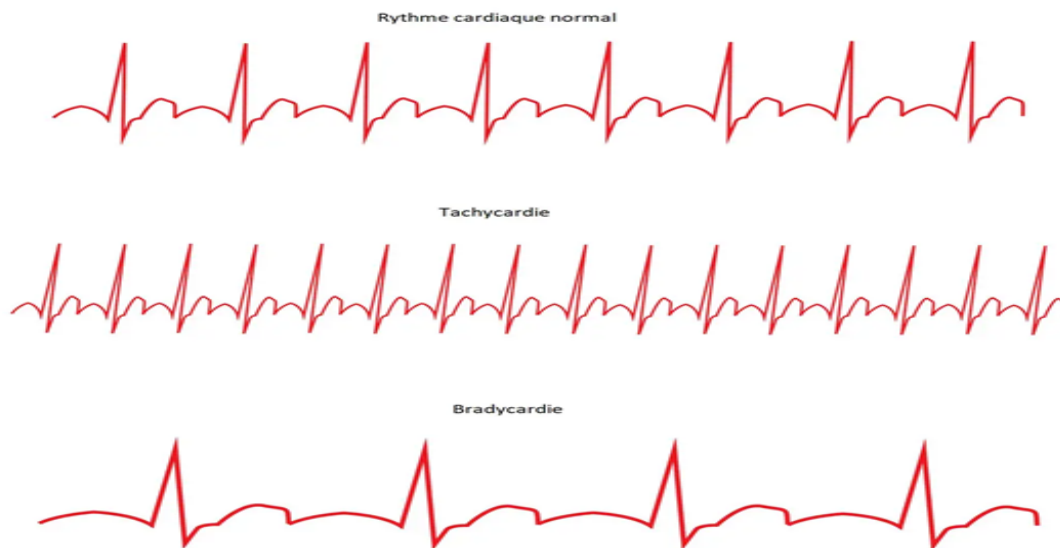


FIGURE 1.3 – Rythme cardiaque [8].

## 1.5 L'Electrocardiographie

L'évolution de l'ECG a connu plusieurs étapes historiques pour atteindre sa forme actuelle La période 1842 - 1942 constitue le premier centenaire de l'électrocardiographie, des balbutiements, des premières découvertes scientifiques à l'ECG 12 dérivations. Nous allons essayer de "refaire"

l'histoire à travers 11 dates.

**1842 : 1er courant électrique d'origine cardiaque**

Carlo Matteucci, physicien italien, montre qu'un courant électrique passe à chaque battement cardiaque.

**1856 : 1er potentiel d'action d'origine cardiaque**

Rudolph von Koellicker Henrich Mueller enregistrent un potentiel d'action, enregistrement fait en mettant en contact un nerf de patte de grenouille avec un cœur isolé : à chaque battement cardiaque, on observait une contraction de la patte de grenouille.

**1887 : 1er ECG humain**

Augustus D. Waller, physiologiste anglais, publie le 1er électrocardiogramme humain. En liant à un tube capillaire dans un champ magnétique une électrode posée sur le thorax d'un " cobaye " humain (T. Goswell, son technicien de laboratoire), on voyait varier le niveau liquide dans le tube à chaque battement de cœur.

**1893 : " Electrocardiogramme " (presque) pour la 1ère fois**

Willem Einthoven utilise le terme électrocardiogramme " pour la 1ère fois " à une réunion de la Deutsch Medical Association. (Il confiera plus tard que A.D. Waller l'avait utilisé avant lui. )

**1895 : PQRST**

W. Einthoven met en évidence 5 ondes sur le tracé électrocardiographique, ondes qu'il nomme respectivement P, Q, R, S, T.

**1910 : 1ère revue d'électrocardiographie**

Walter James et Williams H. publient la 1ère revue américaine d'électrocardiographie.

**1912 : Le triangle d'Einthoven**

W. Einthoven présente pour la 1ère fois devant une société savante, la Chelsea Clinical Society, le célèbre triangle équilatéral formé par les dérivations DI, DII et DIII.

**1920 : L'onde de Pardee**

Harold Pardee publie le 1er ECG d'un infarctus aigu du myocarde chez un homme, avec la repolarisation caractéristique dite "onde de Pardee".

**1924 : Einthoven Nobel**

W. Einthoven reçoit le prix Nobel de physiologie et médecine pour la mise au point du 1er électrocardiogramme.

**1938 : V1 à V6 : position officielle**

La position des dérivations précordiales V1 à V6 est définie par convention par l'American Heart Association et la Cardiac Society of Great Britain.

**1942 : aVR, aVL, aVF**

E. Goldberger ajoute 3 dérivations périphériques aux 3 dérivations déjà décrites par Einthoven, à savoir aVR, aVL et aVF.

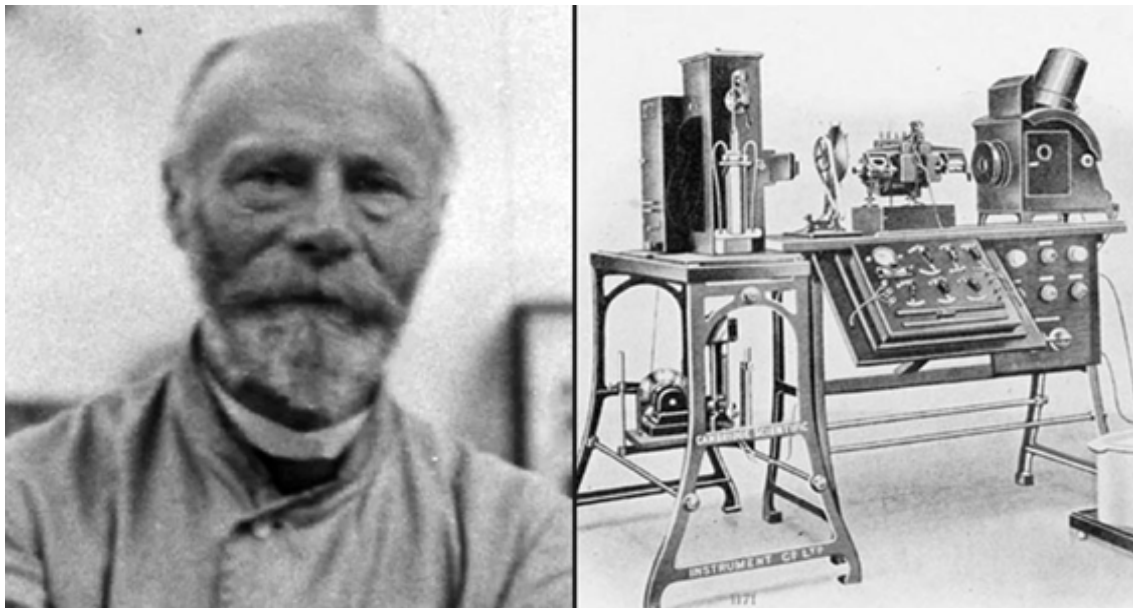


FIGURE 1.4 – Willem Einthoven, prix Nobel pour l'invention de dispositif ECG [6].

### 1.5.1 But de l'Electrocardiogramme

L'électrocardiogramme est un examen majeur en cardiologie inventés au 18ième siècle qui permet qui permet la mesure d'une activité électrique du cœur permettant ainsi de déceler toute anomalie inhérente au fonctionnement cardiaque. Et parmi Les symptômes principale qui nécessite son utilisation sont : la palpitation, les étourdissements, Symptômes ou signes associés aux maladies cardiaques, y compris la tachycardie, les conditions cliniques, le choc et l'hypotension [6].

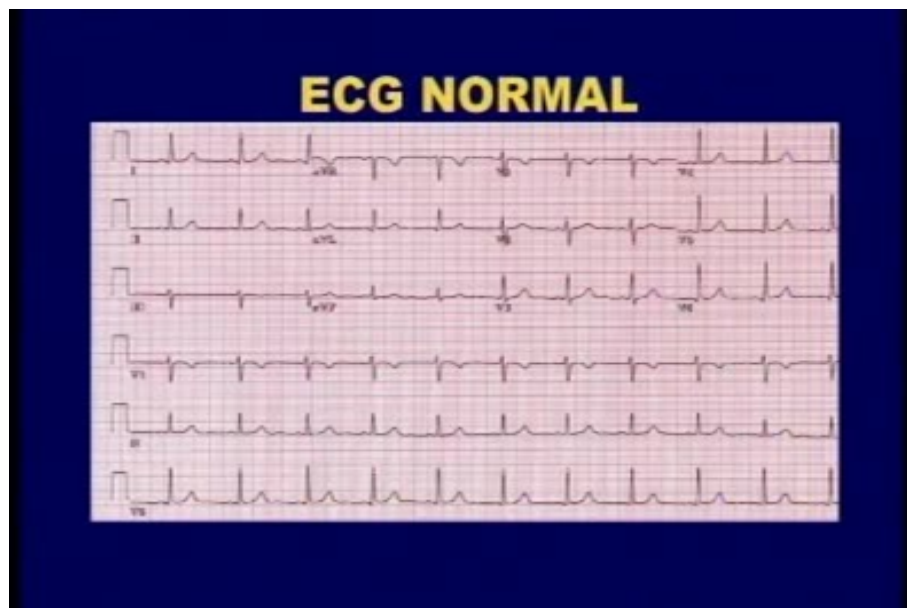


FIGURE 1.5 – Ecg Normal [6].

### 1.5.2 Les ondes du signal ECG

L'électrocardiogramme (ECG) représente l'activité électrique du cœur humain. Il est composé de 5 ondes : P, Q, R, S et l'onde T.

- L'onde P : marque la dépolarisation lors de la contraction des oreillettes, elle est monophasique, arrondie, et toujours positive . -Le complexe QRS : marque la dépolarisation et la contraction ventriculaire, et à leur contraction.
- L'onde Q : quand elle existe, est la première déflexion négative qui suit l'onde P. Souvent, il n'existe pas d'onde Q.
- L'onde R : est la première déflexion positive qui suit l'onde P.
- L'onde S : est la déflexion négative qui suit l'onde R
- l'onde T : est la dépolarisation des ventricules . Elle survient pendant la diastole, et ne témoigne d'aucun événement mécanique. C'est un phénomène purement électrique, elle est souvent positive et de forme asymétrique [7].

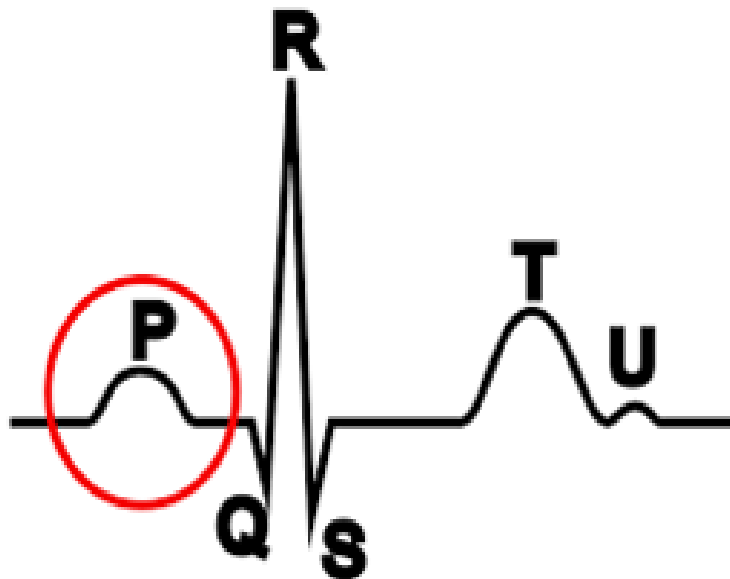


FIGURE 1.6 – Les ondes d'un ECG [7].

### 1.5.3 Les segments et intervalles du 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 [7][8].

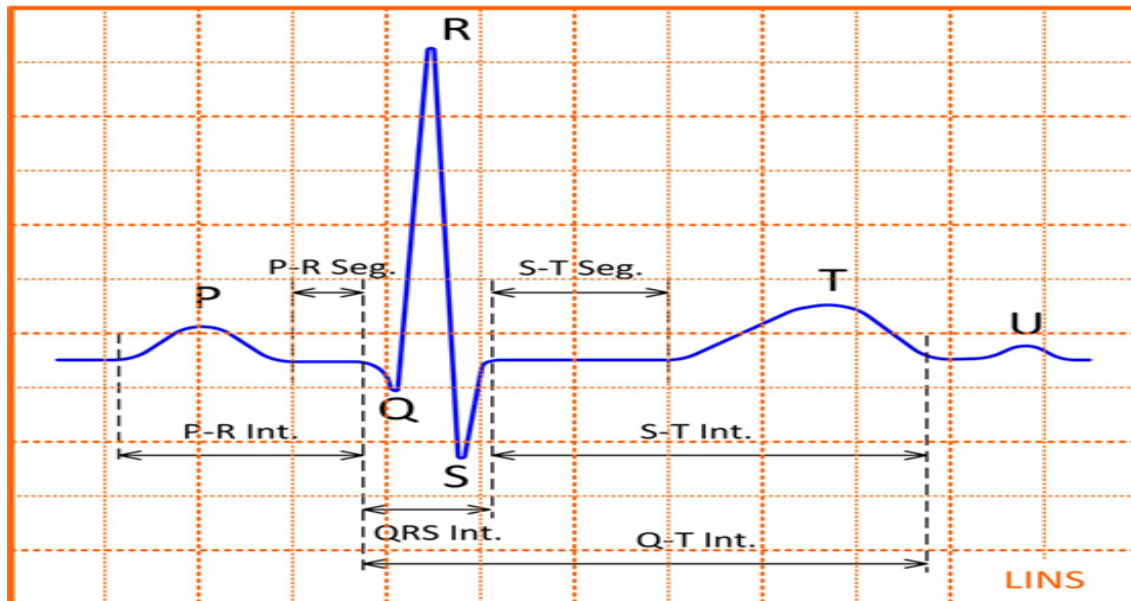


FIGURE 1.7 – Segment et intervalles de L'ECG [8].

- **L'intervalle RR** : c'est la distance entre deux ondes R successives, qui permet de calculer la fréquence cardiaque cet intervalle doit être constant.
- **L'intervalle PR** : Il représente la dépolarisation des deux oreillettes il se mesure depuis le début de l'onde P jusqu'au début de l'onde Q ou de l'onde R, et la mesure se fait du début de l'onde P au début du QRS. Valeur normale :  $0,12 \leq PR < 0,20$ .
- **L'intervalle QRS** : mesure le temps totale de dépolarisation ventriculaire (La durée de l'activation ventriculaire), depuis le début de l'onde Q ou de l'onde R jusqu'à la fin de l'onde S . Durée du QRS normale  $< 0,10s$  QRS modérément allongé  $\geq 0,10s$  QRS allongé  $\geq 0,12s$ . Si le QRS débute par une déflexion négative celle-ci est appelée onde Q.  
Le QS correspond à un complexe QRS qui présente seulement une ou des déflexions négatives.
- **L'intervalle QT** :
  - QTc chez la femme  $< 450$  ms et Des valeurs comprises entre 430 et 450 ms chez l'homme et 450–470 ms
  - QTc long si  $> 450$  ms chez l'homme et  $> 470$  chez la femme.
  - On parle d'un QT court si QTc  $\leq 390ms$  et du syndrome du QT court QTc  $< 320$  ms.

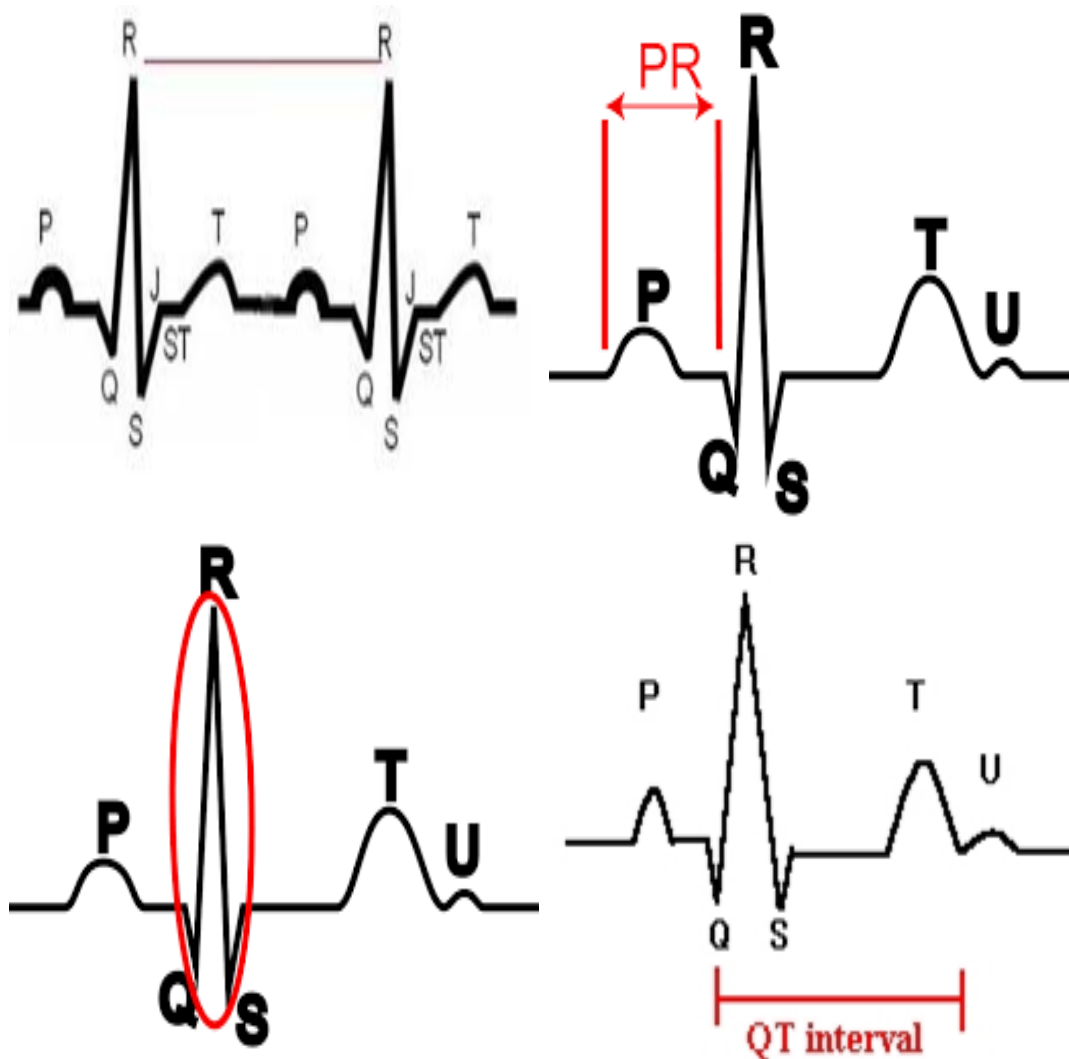


FIGURE 1.8 – Les intervalles RR,PR,QRS,QT respectivement.

### 1.5.3.1 Les dérivations d'un ECG

La machine ECG se compose de 12 dérivations, qui se divisent en deux groupes [9] :

- Les dérivations de membre : qui regardent le cœur dans un plan vertical, classées comme dérivations de membre bipolaires standard I, II et III, et les dérivations unipolaires augmentées aVL, aVF et aVR.
  - Dérivation I : enregistre les différences de potentiel électrique entre le poignet droit et le poignet gauche.
  - Dérivation II : enregistre les différences de potentiel électrique entre le poignet droit et la jambe gauche.
  - Dérivation III : enregistre les différences de potentiel électrique entre le poignet gauche et la jambe gauche.
  - Dérivation VF : paroi postérieure du cœur



-Dérivation aVL : paroi latérale haute du ventricule gauche

-Dérivation aVR : paroi latérale droite [8].

- Les m'ombre bipolaires : Les dérivations des membres sont codées par couleur pour éviter les erreurs de positionnement (bras rouge-droit, bras jaune-gauche, jambe verte-gauche et jambe noire-droite).

Les dérivations précordiales comprennent V1 à V6 enregistrent l'activité électrique du cœur dans le plan horizontal qui sont fixées à la surface de la poitrine. Pour l'emplacement correct, la méthode "Angle de Louis" est une option, et le placement exact est comme suivant [8] :

-V1 est placé à droite du bord sternal et V2 est situé à gauche du bord sternal.

- V4 est placé au niveau du cinquième espace intercostal dans la ligne mi-claviculaire.

- V4 doit être placé avant V3. V3 est placé entre V2 et V4.

-V5 est placé directement entre V4 et V6.

- V6 est placé au niveau du cinquième espace intercostal dans la ligne mi-axillaire.

- V4 à V6 doit s'aligner horizontalement avec le cinquième espace intercostal.

### 1.5.3.2 Interprétation de l'ECG

La lecture d'un ECG se base sur les étapes suivantes [10] 'abord il faut vérifier l'interopérabilité, Date le nom et l'heure, étalonnages de vitesse et en amplitude (10mm/mV,25mm/s), le filtre ECG qui est un filtre classique non numérique se situe entre 0.005 Hz et 150 Hz chez l'adulte l'adulte (250 Hz chez l'enfant). En cas de filtre digitalisé, le filtre pour les fréquences basses peut être situé pour les fréquences < 0,67 Hz et les signes d'inversion des électrodes. Après avoir vérifier que l'ECG est interprétable on vérifie la régularité du rythme et s'assurer de la fréquence cardiaque.[10]

Cas 01 : lorsque le rythme est régulier : (En assumant que l'étalonnage en vitesse est normal 25mm/s)

-) 300 / Nombre de grands carreaux.

-) 1500 / Nombre de petits carreaux.

Comme on peut mémoriser directement la fréquence cardiaque correspondant au nombre de grands carreaux : 300 pour 1 ; 150 pour 2 ; 100 pour 3 ; 75 pour 4 ; 60 pour 5 ; 50 pour 6 ; 43 pour 7 ; 37 pour 8 ; 33 pour 9.[10]

Cas 02 : lorsque le rythme est irrégulier : (En assumant que l'étalonnage en vitesse est normal 25mm/s) Délimiter 6 secondes dans le tracé ECG (30 grands carreaux) ; Calculer le nombre de QRS durant ces 6 secondes ;

Multiplier le chiffre obtenu par 10. Après avoir mesurer la fréquence cardiaque on s'intéresse a

l'onde p qui correspond aux forces électriques générés par l'activation des oreillettes. La première moitié de l'onde P correspond à la dépolarisation de l'oreillette droite ; la deuxième moitié correspond à la dépolarisation de l'oreillette gauche. et pour analyser l'onde P et déterminer ces valeurs on s'intéresse à la -L'Amplitude :

En DII de la ligne de base au sommet < 2,5mm En V1 de la ligne de base au sommet < 1,5mm, de la ligne de base au nadir < 1mm

-) Durée : Du début à la fin de l'onde P, < 0,12s (0,08 à 0,11 s)

-) Axe : 0 à 75°

-) Morphologie : L'onde P sinusale est positive en DI, DII, aVF, et positive ou biphasique en DIII, aVL, V1, V2.

Vérifier la morphologie de toutes les ondes P, et rechercher des ondes P cachées dans l'onde T ou segment ST.

Morphologies possibles : Onde P inversée, biphasique, flutter, fibrillation. NB : L'onde P peut être absente.

Après avoir analysé l'onde P on détermine l'origine du rythme et pour cela on doit s'intéresser à

La fréquence cardiaque ; La régularité du rythme ;

La morphologie de l'onde P ; Interval PR ;

La durée du QRS ;

La relation P/QRS

Et parmi les éléments qui caractérisent un rythme sinusal normal

-) FC entre 60 et 100 bpm -) Rythme régulier (variabilité PP < 0,16s ou < 10%)

-) Onde P sinusale

-) PR constant

-) QRS < 100ms (sauf trouble de la conduction intraventriculaire associé)

-) P/QRS=1 (chaque onde P est suivie d'un QRS, chaque QRS est précédé d'une onde P [10].

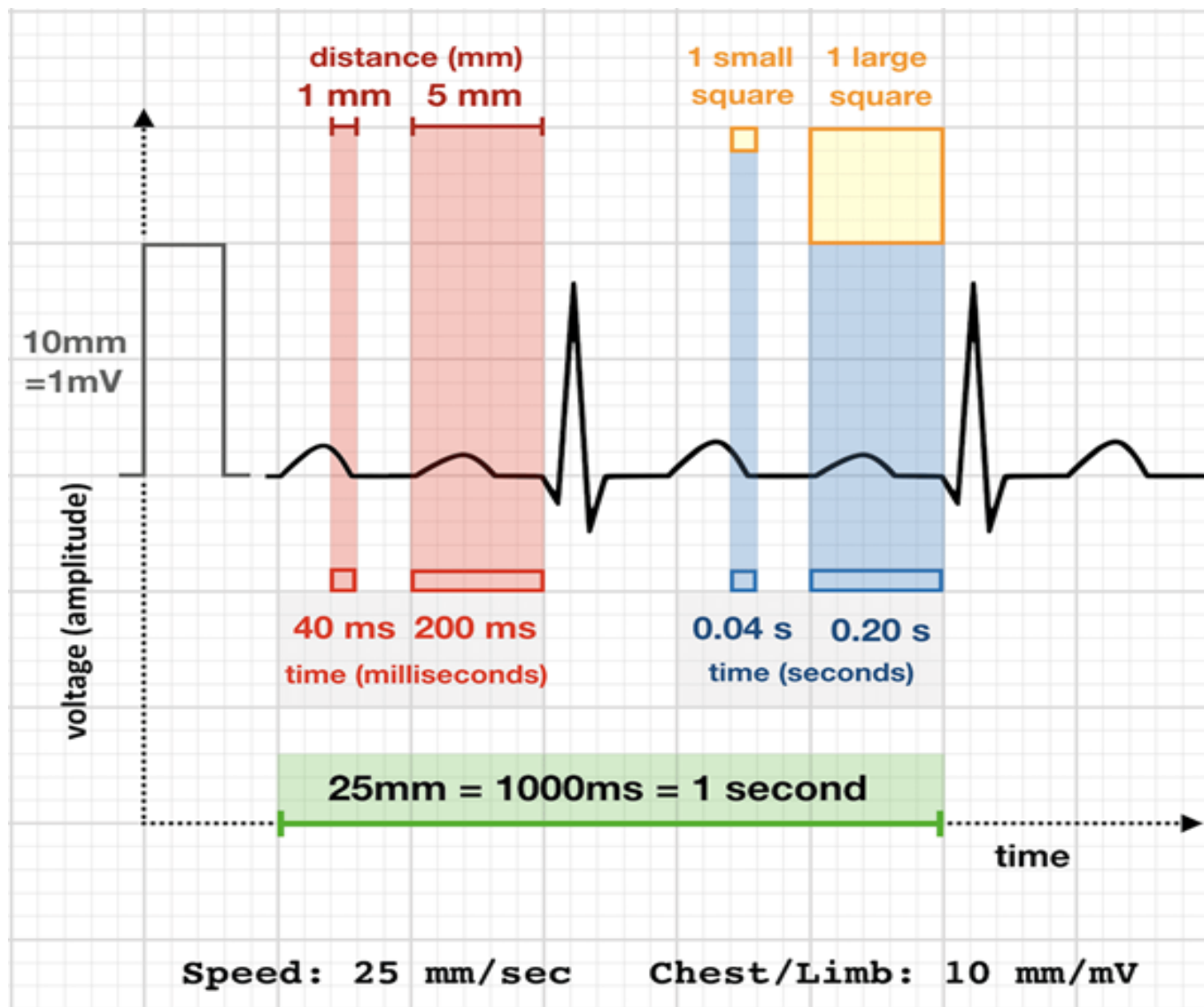


FIGURE 1.9 – Interprétation de L'ECG [12].

## 1.6 Conclusion

A la fin de ce chapitre on conclut que le cœur est un organe essentiel et l'ECG est la meilleure invention pour détecter les différentes maladies cardiaque Dans le deuxième chapitre nous aborderons les concepts de deep Learning et ces différentes approches.

# Chapitre 2

## Techniques de l'intelligence artificielle

### 2.1 Introduction

Dans un souci de développer des techniques et des méthodes pouvant imiter le comportement humain dans le but de les rendre si puissantes, des chercheurs en la matière se sont intéressés à ce qu'on appelle l'intelligence artificielle.

Dans ce sens nous allons aborder dans notre chapitre en première lieu la notion de l'intelligence artificielle est une discipline scientifique recherchant des méthodes de solution de problèmes à forte complexité logique ou algorithmique. En second plan, nous allons évoquer l'apprentissage automatique (machine Learning) qui est lui-même un champ d'étude de l'intelligence artificielle et l'apprentissage profond (deep learning) qui est un ensemble de méthodes d'apprentissage automatique tentant de modéliser avec un haut niveau d'abstraction des données grâce à des architectures articulées de différentes transformations non linéaires.

### 2.2 L'Intelligence artificielle

Si on pose la question à quelques dizaines d'ingénieurs dans le domaine de l'intelligence artificielle, on risque d'avoir une dizaine de définitions différentes. Effectivement, nos recherches nous ont permis de constater qu'il existe différentes définitions de l'IA : selon, Marvin Lee Minsky : " la construction de programmes informatiques qui s'adonnent à des tâches qui sont pour l'instant, accomplies de façon plus satisfaisante par des êtres humains car elles demandent des processus mentaux de haut niveau tels que : l'apprentissage perceptuel, l'organisation de la mémoire et le raisonnement critiquée" [12].

## 2.3 techniques d'intelligence artificielle

### 2.3.1 Le Machine Learning – Apprentissage automatique

Le machine Learning est l'un des champs d'étude de l'intelligence artificielle. C'est également une science qui concerne le développement et l'implémentation de procédures automatisables qui permettent à une machine d'apprendre et d'évoluer. L'être humain a tendance à améliorer sa situation en utilisant l'expérience obtenue durant des années.

Nous pouvons également appliquer ce procédé sur des machines. C'est ce que nous appelons machine Learning ou apprentissage automatique qui vise à entraîner un algorithme en se basant sur des exemples [13].

Cette technique permet à la machine d'apprendre sans avoir à modifier les algorithmes qui la constituent et donc, elle pourra exécuter des tâches pour lesquelles elle n'est pas programmée.

Ci-dessous, un schéma qui résume le processus d'apprentissage automatique :

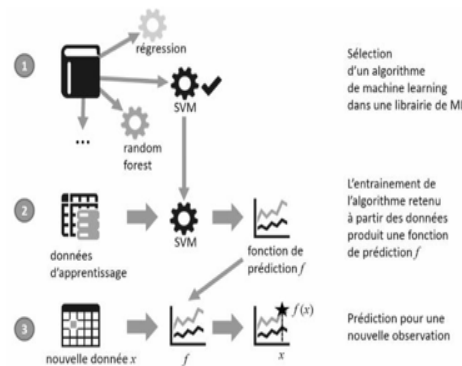


FIGURE 2.1 – Processus d'apprentissage automatique [13].

D'après le schéma ci-dessus, nous distinguons 3 grandes étapes du processus d'apprentissage automatique.

La première étape est la sélection d'un algorithme de ML pour résoudre un problème donné. La seconde étape consiste à entraîner cet algorithme afin qu'il puisse minimiser l'erreur et construire la fonction de prédiction la plus optimale.

La troisième et dernières étapes consiste à donner de nouvelles entrées à notre algorithme afin de prédire une valeur quantitative ou bien une valeur qualitative. Il faut noter qu'avant la première étape, un prétraitement des données est nécessaire pour espérer l'obtention d'un bon modèle.

#### 2.3.1.1 Quelques types de Machine Learning

Il existe plusieurs types de ML que nous allons définir succinctement.

- **L'apprentissage supervisé :** L'apprentissage supervisé est ce que l'on entend parler le plus quand nous évoquons le machine Learning. Pour mieux comprendre ce modèle, prenons par

exemple une fonction  $f$  qui a comme entrées, des données d'apprentissage (variables explicatives)  $X$  et en sorties une variable cible  $Y$  connues. Nous appelons cela l'étiquetage  $Y=f(X)$ . Ce modèle a donc pour objectif de construire une fonction de prédiction  $f$  grâce aux observations des variables  $X_1\dots X_n$  et  $Y_1\dots Y_n$ . Exemple d'algorithmes que nous pouvons appliquer : La régression linéaire et les machines à vecteurs supports [14].

- **L'apprentissage non supervisé** : Contrairement à l'apprentissage supervisé, on ne trouve pas d'étiquetage dans ce modèle. L'objectif est que l'algorithme lui-même utilise un jeu de données brutes et obtient un résultat en se fondant sur la détection de similarités entre certaines de ces données. L'algorithme des  $k$ -means ( $k$ -moyennes) appartient à cette catégorie [15].
- **L'apprentissage par renforcement** : Apprentissage par renforcement est un domaine de l'apprentissage machine concerne la façon dont les agents logiciels devraient prendre des mesures dans un environnement afin de maximiser la notion de récompense cumulative. C'est un type important d'apprentissage automatique dans lequel un agent apprend à se comporter dans un environnement en exécutant des actions et en voyant les résultats [16]. Par rapport à l'apprentissage non supervisé, l'apprentissage par renforcement est différent en termes d'objectifs. Alors que le but de l'apprentissage non supervisé est de trouver des similitudes et des différences entre les points de données, dans l'apprentissage par renforcement, l'objectif est de trouver un modèle d'action approprié qui maximiserait la récompense cumulative totale de l'agent.

### 2.3.1.2 Quelques algorithmes du Machine Learning

Il n'existe pas d'algorithme d'apprentissage parfait ou d'algorithme meilleur que l'autre. Chacun peut être meilleur selon le domaine d'étude. C'est là que le facteur humain intervient pour faire le bon choix en tenant compte des données utilisées, de l'objet de la problématique et surtout des indicateurs de performances. Nous allons voir quelques algorithmes populaires du ML.

- **La régression linéaire** :Ce modèle de ML supervisé est le plus simple de tous. Il est modélisé comme suit :  $f(x)=a_1x_1+\dots+a_nx_n+b$  Où  $x$  est la variable explicative (prédictive) et  $y(x)$ , la variable cible (à prédire),  $a$  et  $b$  sont des paramètres ou coefficient à calculer pour minimiser les erreurs de prédiction.

Nous pouvons utiliser la régression linéaire lorsqu'il s'agit d'étudier une tendance en économétrie et dans le marketing. Pour appliquer cet algorithme, il faut avoir de bonnes raisons de penser que la variable cible varie selon les variables explicatives choisies. L'inflation causé par une augmentation de prix par exemple. Un schéma pour résumer ce qu'a été dit : [17]

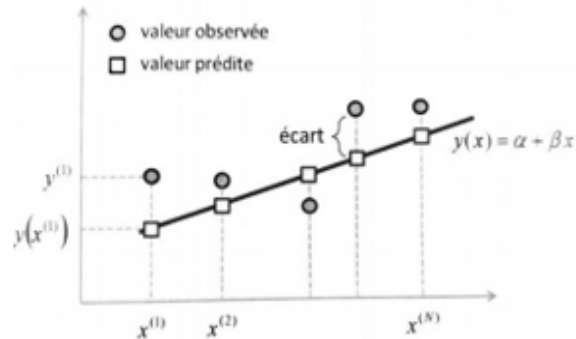


FIGURE 2.2 – Nuage de point des valeurs prédites et observées [17].

- **L'algorithme des k plus proches voisins** :KNN pour K Nearest Neighbors. Il appartient à la catégorie de classement supervisé. Son principe est simple, ce modèle suppose que les observations voisines sont similaires. Notre base d'apprentissage contient donc des exemples étiquetés. Le but est qu'à l'introduction d'un nouvel exemple non étiqueté, l'algorithme pourra affecter ce nouvel exemple à la bonne classe. Ceci en utilisant les k plus proche voisins étiquetés de la base d'apprentissage comme nous pouvons le voir dans l'image ci-dessous : [17]

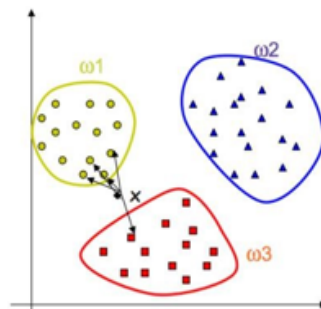


FIGURE 2.3 – Affectation d'un nouvel élément  $x$  à une classe par K-NN [17].

Nous constatons sur l'image ci-dessus, trois classe et le but c'est de trouver la classe de

l'exemple inconnu  $x$ . Nous choisissons notre paramètre  $k$  par exemple à 5 et nous calculons la distance (euclidienne, Manhattan, ...). Nous constatons qu'il y a 4 voisins de la classe  $w_1$  qui sont plus proche de  $x$ , donc elle sera affectée à la classe  $w_1$ .

**Remarque :** l'algorithme a besoin de connaître  $k$  pour savoir combien de voisins va-t-il prendre en considération. Pour un choix optimal du paramètre  $k$ , la validation croisée est souvent utilisée.

- **Arbre de décision :** Les arbres de décisions sont des modèles de ML supervisés et non paramétriques, nous pouvons les utiliser que ce soit pour la classification ou pour la régression. Nous pouvons donc les appeler arbre de classification ou arbre de régression selon le champ d'application. A noter que ce sont des méthodes qui permettent d'obtenir des modèles à la fois explicatifs et prédictifs. Ils fonctionnent sous le principe d'une hiérarchie de tests sur les valeurs des attributs, ceci dit, quand il s'agit de régression, l'algorithme va produire une valeur numérique. Par contre s'il s'agit d'une classification, l'algorithme prend un élément précis dans un ensemble de conclusion [17].

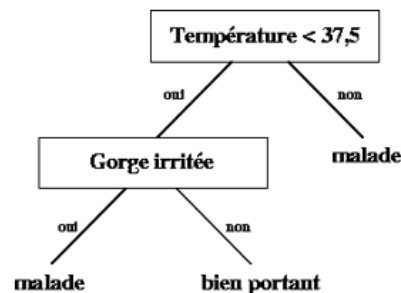


FIGURE 2.4 – Exemple d'arbre de décision [17]

- **Forêt aléatoire – Random Forest :** Dans la nature, une forêt est un ensemble d'arbres. En machine Learning, c'est tout à fait pareil. Le modèle de la forêt construit plusieurs arbres de décision pour ensuite les fusionner pour obtenir des résultats de prévisions plus précis et plus stable. Comme c'est une combinaison d'arbre de décision, les modèles de forêt peuvent être utilisés dans la régression et la classification également.

Le principe de ces modèles est de construire plusieurs arbres de décision, ensuite un système de vote majoritaire pour la prédiction ou la classification.

La notion d'aléas est ajoutée pour ce type de modèle car au lieu de rechercher la caractéristique la plus importante lors de la division d'un nœud, elle recherche la meilleure fonctionnalité parmi un sous ensemble aléatoire d'entités. La diversité engendrée aboutit généralement à un meilleur modèle.

L'existence de ce modèle vient éliminer les inconvénients (sur-apprentissage et la complexité



des opérations d'élagage) des arbres de décisions tout en conservant la plupart de leurs atouts. Ces modèles se sont fait connaître par Leo BREIMAN et Adèle CUTLER en 2001 [17].

### 2.3.2 Le Deep Learning – Apprentissage profond

Le Deep Learning est une approche d'intelligence artificielle, dérivée du machine Learning. Nous pouvons voir le Deep Learning comme une évolution du machine Learning. Il s'agit d'une méthode d'apprentissage particulière qui ouvre un champ de possibilité très large, [18] où la machine est capable d'apprendre par elle-même, contrairement à la programmation où elle se contente d'exécuter à la lettre des règles prédéterminées [19].

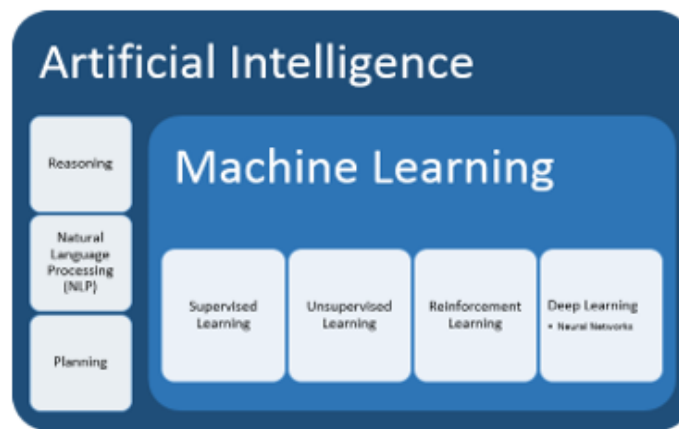


FIGURE 2.5 – La relation entre l'intelligence artificielle, le machine Learning et le deep Learning [19].

### 2.3.3 Les réseaux de neurones

Le deep learning repose sur des algorithmes qu'on appelle les réseaux de neurones dont le but est de parvenir à simuler l'activité du cerveau humain. C'est à dire les échanges d'informations entre les neurones à travers la synapse « donnée d'entrée – engendre – donnée de sortie ». En additionnant ces espaces inter synaptiques en parallèle, puis en superposant des couches. L'objectif est de générer un réseau de pondérations auto-réglés par l'algorithme à qui l'on indiquera simplement ce qui est attendu en sortie en fonction de ce qu'on y aura entré, laissant ainsi la machine faire les réglages entre les deux. La structure d'un réseau de neurones est inspirée des études menées sur le fonctionnement du cerveau. C'est une technique très puissante qui peut traiter des problèmes différents et dans plusieurs domaines.

Les algorithmes d'un réseau de neurones prennent donc plusieurs valeurs en entrée qui sont traitées par une multitude de fonctions pour avoir un résultat connu. Mais avant cela, ces fonctions passent par une phase d'apprentissage afin de calibrer les résultats en sortie. Une fois que le réseau est bien ajusté et configuré, nous pouvons lui donner des valeurs en entrée dont le résultat est inconnu. Un résultat sera retourné sans que nous sachions ce que le réseau a réellement fait (boîte noire) [20].

Les nœuds en vert représentent les données d'entrée du réseau de neurones. Elles peuvent être soit

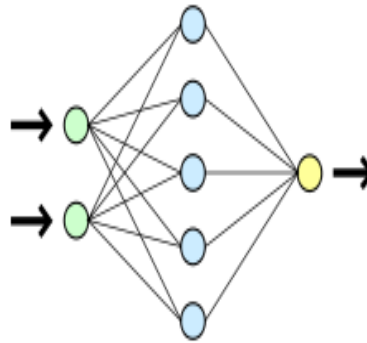


FIGURE 2.6 – Réseau de neurone simple [20].

un vecteur ou une matrice. De plus, on retrouve des fonctions qui définissent la propagation de l'entrée vers la sortie (nœud jaune) à travers la ou les couches cachées (nœuds bleus). Nous pouvons considérer un réseau de neurone comme un approximateur de fonction universelle. C'est-à-dire, il permet d'approcher n'importe quelle fonction tout en minimisant l'erreur.

On distingue deux grandes familles de réseaux de neurones : les réseaux bouclés et les réseaux non bouclés.

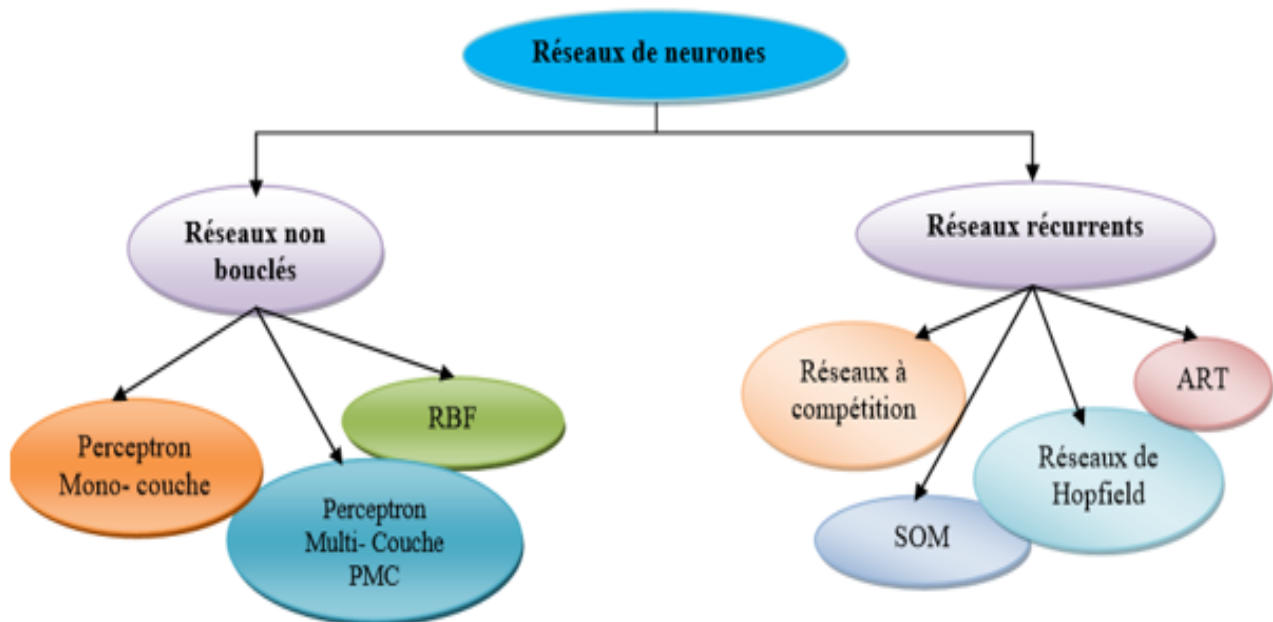


FIGURE 2.7 – Famille de réseau de neurones [27].

Les réseaux bouclés sont appelés aussi réseaux récurrents ou réseaux dynamiques, permettent des connexions arbitraires entre les neurones de toutes les couches avec retour en arrière.

Un réseau de neurone non bouclé (dit aussi statique) sont représenté graphiquement par un ensemble de neurones connectés entre eux, l'information circulant des entrées vers les sorties sans retour en arrière.

**Perceptron** :Le perceptron est l'un des réseaux de neurones les plus utilisés pour des problèmes d'approximation, de classification et de prédiction il a été introduit pour la première fois en 1957 par Frank Rosenblatt [27].

**Perceptron simple** :C'est un Réseau neuronal constitué d'un seul neurone artificiel ayant une valeur de sortie binaire pour calculer cette sortie, le neurone effectue une somme pondérée de ses entrées :  $Y = F(w_1 \times X_1 + W_2 \times X_2)$ , puis applique une fonction d'activation à seuil : si la somme pondérée dépasse une certaine valeur, la sortie du neurone est 1, sinon elle vaut 0. Il ne peut donc faire que de la classification, puis de la prédiction [26].

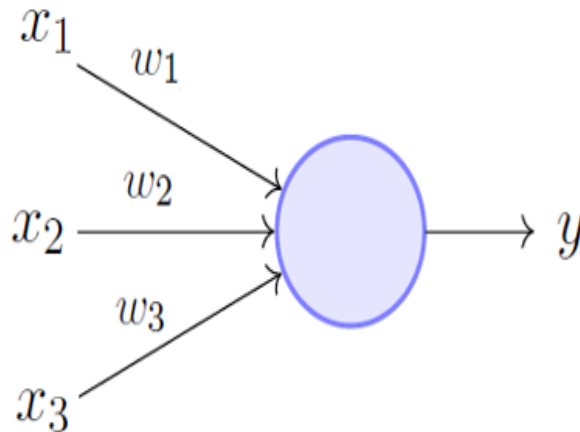


FIGURE 2.8 – Schéma du perceptron simple [26].

### Perceptron multicouches :

Le Perceptron multicouche est un Classifieur linéaire de type réseau neuronal formel organisé en plusieurs couches au sein desquelles une information circule de la couche d'entrée vers la couche de sortie uniquement ; il s'agit donc d'un réseau de type feedforward (en). Chaque couche est constituée d'un nombre variable de neurones, les neurones de la couche de sortie correspondant toujours aux sorties du système [28].

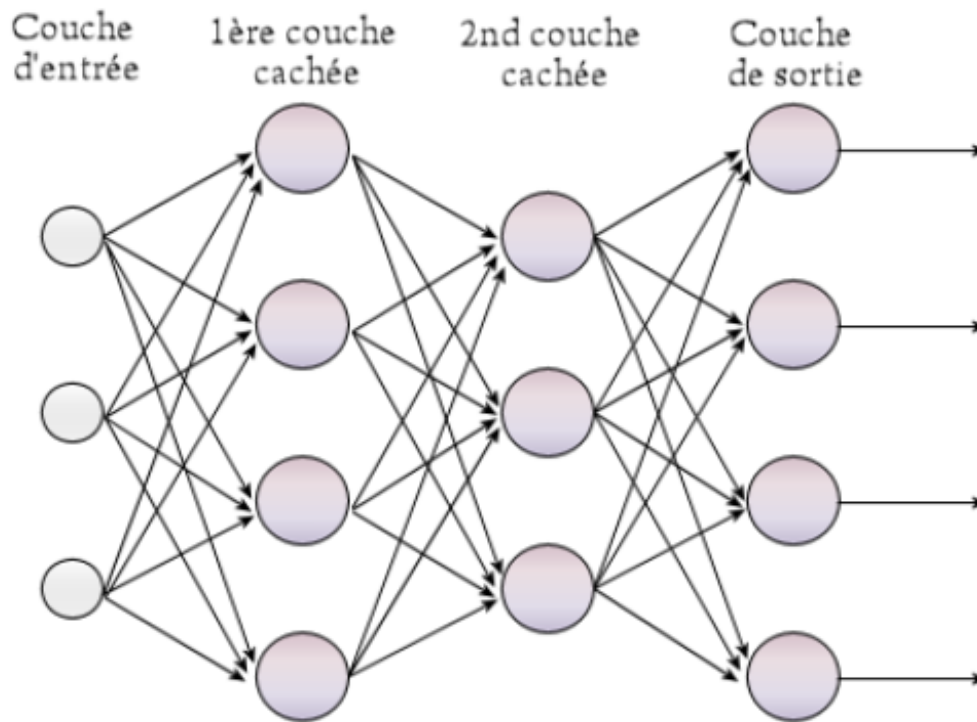


FIGURE 2.9 – Réseaux du perceptron multicouche [28].

**Les machines de Boltzmann restreintes (RBM) :** La machine restreinte de Boltzmann est le premier programme de deep learning à fonctionner sur une base de données non supervisée, elle est couramment utilisée pour avoir une estimation de la distribution probabiliste d'un jeu de données. Elle a initialement été inventée en 1986 par Paul Smolenski. Ce réseaux neurones fonctionne avec une couche d'entrée appelée visible et une couche cachée, les connexions entre les neurones sont restreintes car deux nœuds de la même couche ne partagent pas une connexion cela signifie que les neurones sont connectés uniquement aux neurones des autres couches. Dans la première couche, toutes les données ne sont pas forcément étiquetées. Les informations sont ensuite traitées par une couche cachée, qui va permettre d'avoir un étiquetage correct [29].

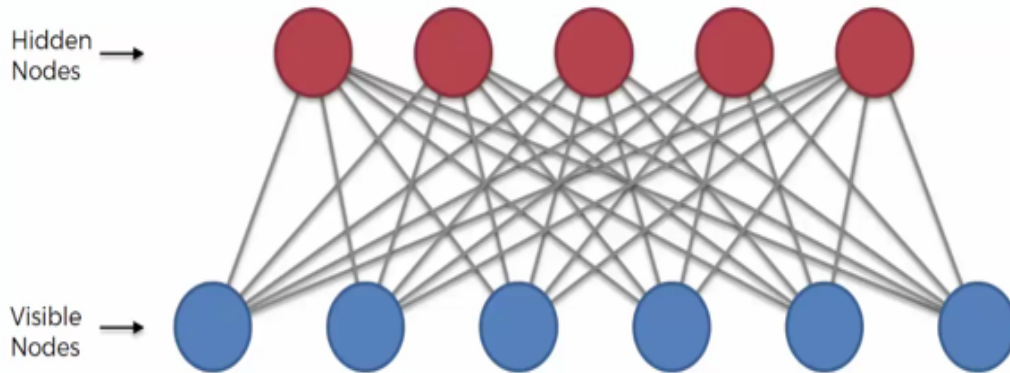


FIGURE 2.10 – Machine Boltzmann restreinte [22].

### 2.3.4 Domaine d'application du deep learning

Le deep Learning est utilisé dans divers domaines [21] :

- Reconnaissance d'image,
- Traduction automatique,
- Voiture autonome,
- Diagnostic médical,
- Recommandations personnalisées,
- Modération automatique des réseaux sociaux,
- Prédiction financière et trading automatisé,
- Identification de pièces défectueuses,
- Détection de malwares ou de fraudes,
- Chatbots (agents conversationnels),
- Exploration spatiale,
- Robots intelligents.

### 2.3.5 Différences entre Deep Learning et Machine Learning

En termes pratiques, l'apprentissage profond n'est qu'un sous-ensemble de l'apprentissage automatique. Ils fonctionnent d'une manière similaire mais leurs capacités sont différentes.

Bien que les modèles d'apprentissage automatique de base s'améliorent progressivement, quelle que soit leur fonction, ils ont encore besoin de conseils. Si un algorithme d'IA renvoie une prédiction inexacte, un ingénieur doit intervenir et effectuer des ajustements. Avec un modèle d'apprentissage en profondeur, un algorithme peut déterminer seul si une prédiction est précise ou non via son propre réseau neuronal [22].

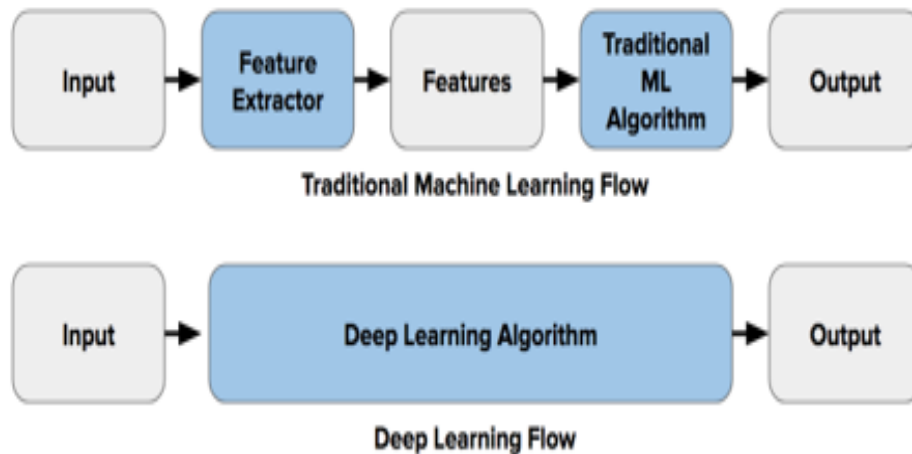


FIGURE 2.11 – Comparaison entre la machine Learning et le Deep Learning [22].

### 2.3.6 Avantages des réseaux de neurones

Les réseaux de neurones représentent des avantages, tels que [23] :

- Les réseaux de neurones sont souples et génériques. Ils peuvent résoudre différents types de problèmes dont le résultat peut être : une classification, analyse de données, etc.
- Ils traitent des problèmes non structurés sur lesquels aucune information n'est disponible à l'avance.
- Les réseaux neuronaux se comportent bien parce que même dans des domaines très complexes, ils fonctionnent mieux que les arbres de statistique ou de décision.
- Les réseaux de neurones fonctionnent sur des données incomplètes ou bruitées. Cette lacune d'information peut être complétée par l'ajout d'autres neurones à la couche cachée.

### 2.3.7 Inconvénients des réseaux de neurones

Les réseaux de neurones ont aussi des inconvénients, tels que [23] :

- La lenteur d'apprentissage.
- La difficulté de choisir des valeurs initiales des poids de connexion ainsi que l'adaptation du pas d'apprentissage.
- L'apprentissage au détriment de la généralisation.
- En cas d'erreur dans les résultats de sorties, l'utilisateur n'a aucune information sur le fonctionnement interne.

## 2.4 Médecine et IA

Au cours des dernières années, le domaine de la santé a connu une véritable révolution, s'appuyant sur les réseaux neuronaux et les techniques de deep learning, l'intelligence artificielle devient un atout considérable dans l'évolution et l'amélioration de la médecine. L'objectif de l'utilisation de l'intelligence artificielle n'est pas de remplacer le corps médical mais de devenir un atout pour perfectionner le diagnostic, améliorer la prise en charge et personnaliser les traitements en fonction des patients [24].

## 2.5 Les domaines les plus impactés par l'IA

- **Perfectionnement du diagnostic** : A différents niveaux, l'IA devient un atout pour détecter les pathologies et analyser des résultats médicaux, grâce à la considérable base de données sur laquelle les machines s'appuient afin de lire les examens et résultats médicaux. En effet, ces machines ont été préparées à l'élaboration de leur diagnostic par le traitement d'une multitude de résultats auparavant, rendant ainsi leur évaluation particulièrement rigoureuse. Plusieurs expériences ont déjà été menées sur la lecture d'IRM et de scanners, révélant l'efficacité de l'IA. Une étude sur la détection d'un cancer de la peau a notamment mis en exergue un taux de précision plus important chez l'IA que chez les spécialistes de la dermatologie [24].
- **La chirurgie** : L'intelligence Artificielle pourrait impacter diverses branches de la médecine entre autres, la chirurgie, cette révolution est déjà en partie en place avec le développement de la chirurgie assistée par ordinateur. En effet, plusieurs opérations sont déjà réalisées à l'aide de bras articulés dirigés par un chirurgien grâce à une vision 3D. Ce type de technologie offre une maniabilité et une précision encore plus importantes que le chirurgien lui-même. [24].
- **La radiologie** : L'intelligence Artificielle dispose d'un ensemble de technique qui nous offre une meilleure précision de l'interprétation des images via les échographies, les IRM ou les scanners, de plus les résultats de cette dernière dépasse parfois celle des experts eux-mêmes. A l'avenir, les résultats obtenus par cette nouvelle technologie deviendront des diagnostics fiables et précis sur lesquels s'appuyer [24].
- **Le domaine thérapeutique** : L'intelligence artificielle permet de traiter un nombre considérable de données, et ce d'une façon rapide et efficace. En outre l'IA peut représenter une aide incontournable dans l'élaboration de nouveaux médicaments. En effet, à partir de plusieurs sources d'informations (essais cliniques, dossiers médicaux et génétiques, allergies...), cette dernière permet de prédire les potentiels effets que les nouveaux médicaments pourraient avoir sur les patients. En ce sens, les outils d'intelligence artificielle permettent de personnaliser le traitement en fonction des profils médicaux [24].
- **Les systèmes experts** : Sont une des applications de l'IA. Ils imitent le raisonnement d'un professionnel spécialiste dans un domaine précis par exemple en matière de diagnostic de panne automobile ou de diagnostic médical. Le système pose des questions à l'utilisateur, ses

réponses orientent le système qui au fur et à mesure affine son diagnostic [25]. Un système expert se compose de 3 parties :

- Une base de faits,
- Une base de règles,
- Un moteur d'inférence.

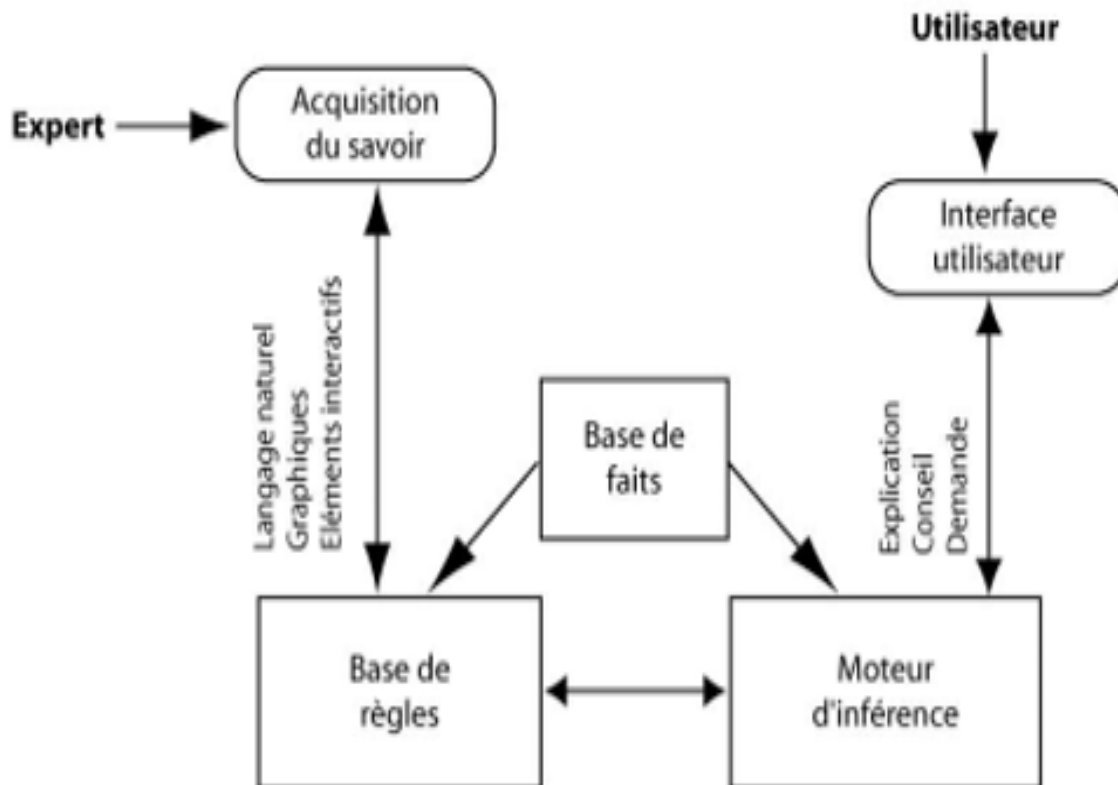


FIGURE 2.12 – Schéma de fonctionnement d'un système expert [25].

## 2.6 Conclusion

Nous avons consacré ce chapitre à la présentation des notions de base comme L'IA et son évolution, l'apprentissage automatique, ses types et les algorithmes les plus populaires et les plus utilisés, ensuite nous avons présenté l'apprentissage profond et son fonctionnement « les réseaux de neurones » et quelques domaines d'applications dont il est utilisé, nous nous sommes aussi intéressés à l'utilisation de l'IA dans le domaine de la médecine.



# Chapitre 3

## Deep learning pour la détection d'anomalies

### 3.1 Introduction

L'étude sur les maladies cardiovasculaires a toujours été un sujet médical populaire dans le monde. Pour toute l'humanité, l'étude de l'arythmie a une signification profonde et significative, le signal ECG est un indicateur continu du fonctionnement du cœur et sa surveillance est très utile pour détecter toute déviation fonctionnelle. Le machine learning via deep learning apporte de nombreuses solutions aux professionnels de santé qui peuvent être utilisées pour analyser et classer les données ECG.

Au cours de ce chapitre nous allons utiliser les méthode d'apprentissage en profondeur basée sur la LSTM (mémoire à long terme) et CNN (réseau neuronal convolutif) pour la détection des anomalies de rythme cardiaque. Et nous allons aussi cite quelques méthodes déjà appliquées.

### 3.2 Etat d'art

Pour la détection d'anomalie d'un ECG plusieurs méthodes ont été proposées parmi ces méthodes :

- Dans cette étude faite par Dr Ashok Kumar Dohare, Dr Vinod Kumar, Dr Ritesh Kumaront en 2017 consiste la détection de l'infarctus du myocarde a été réalisée à l'aide d'un système ECG à 12 dérivation avec classificateur SVM. En utilisant la base de données PTB standard ,L'analyse en composantes principales (ACP) est utilisée pour réduire le coût de calcul, selon la première expérience une fonction de biais radial de la fonction noyau (RBF) avec une validation croisée décuplée utilisé Dans leur travail, Pour l'analyse statistique le glyphplot (X) a été utilisé tel que chaque glyphplot représente une avance et chaque rayon en étoile est proportionnel à une variable et Chaque rayon d'une étoile représente une variable et la longueur de rayon est proportionnelle à la valeur de cette variable pour cette observation . Selon la deuxième expérience pour réduire la complexité du calcul, la réduction de la dimension des entités est importante. Par conséquent, la méthode proposée applique la technique de réduction de l'analyse en composantes principales (ACP), la fonction de biais radial (RBF) utilise

une validation croisée décuplée pour trouver le meilleur paramètre pour former l'ensemble des données d'entraînement. Le taux de réussite est de cette étude est 96,66%.

- Dans cette étude, Abhinav Vishwa, Mohit K. Lal, Sharad Dixit, Dr Pritish Vardwaj ont parlés principalement à la production de résultats de classification d'arythmie applicables dans les systèmes d'aide à la décision diagnostique. Le modèle de réseau neuronal avec algorithme de rétro-propagation est utilisé pour classer les cas d'arythmie en classes normales et anormales. Les modèles de réseaux sont formés et testés pour l'arythmie MIT-BIH. Ils ont utilisé la base de données d'arythmie MIT-BIH de physio et avec l'utilisation de deux canaux comme données source. La première étape du prétraitement des données ECG est la réduction du bruit de base. Ils ont supprimé la composante basse fréquence et appliqué la transformation de Fourier rapide directe pour supprimer les basses fréquences et restaurer l'ECG à l'aide de la FFT inverse Après la réduction du bruit de base, la segmentation du battement ECG a été effectué. Dans cette étape, les signaux ECG continus ont été transformés en battements ECG individuels. Parmi tous les réseaux de neurones, la propagation Feed Forward back a été Choisie des fins de test, nous utilisons un mélange d'arythmie MIT –BIT, de base de données QT et de base de données NSR. Le taux de réussite est 96,77% sur la base de données MIT-BIH et de 96,21% sur la base de données préparée.

### 3.2.1 Analyse et critique

Après analyse des articles nous avons constaté que la plupart des auteurs ont obtenus des résultats satisfaisant sur la détection d'anomalies de rythme cardiaque. Cependant, malgré les résultats encourageants, nous constatons certains manquements :

- Dans les travaux de (Ashok Kumar Dohare, Dr Vinod Kumar, Dr Ritesh Kumaront) En Appliquant cette méthode malgré le taux de réussite élevé mais elle est comparable aux autre chercheurs car la sensibilité est la même et la précision est à peine réduite. Les SVMs sont des systèmes d'apprentissage qui utilisent un espace d'hypothèses de fonctions linéaires on peut introduire artificiellement un déséquilibre des classes dans la construction des modèles individuels et déplacent le problème du sur-ajustement de l'optimisation des paramètres à la sélection du modèle. Malheureusement, les modèles de noyau peuvent être assez sensibles au sur-ajustement du critère de sélection du modèle.
- Dans les travaux de (Abhinav Vishwa, Mohit K. Lal, Sharad Dixit, Dr Pritish Vardwaj) Tout d'abord Les problèmes doivent être traduits en valeurs numériques avant d'être introduits dans ANN et Lorsque la méthode produit une solution, cela ne donne aucune idée du pourquoi et du comment et Cela réduit la confiance dans le réseau.

## 3.3 CNN

Le Deep learning (DL) constitue une technique moderne de traitement d'images, avec un grand potentiel, appliqué avec succès dans divers domaines tels que la reconnaissance et la classification

d'images vidéos... etc. Cette technique est appelée les réseaux de neurones convolutifs désignés par l'acronyme CNN, en l'anglais Convolutional Neural Network, introduits pour la première fois par Fukushima en 1998 [30].

Un CNN utilise un ensemble de filtres et d'opérateur de convolution pour réduire le nombre de paramètres. Cet algorithme a déclenché les techniques de pointe pour la classification d'images. Essentiellement, la façon dont cela fonctionne pour 1D CNN est de prendre un filtre (noyau) de taille kernel-size commençant par le premier horodatage. L'opérateur de convolution prend le filtre et multiplie chaque élément par rapport aux premiers kernel-size pas de temps. Ces produits sont ensuite additionnés pour la première cellule de la couche suivante du réseau neuronal. Le filtre se déplace ensuite par stridepas de temps et se répète. La valeur stride par défaut dans Keras est 1, que nous utiliserons. Dans la classification d'images, nous utiliserons le remplissage par défaut qui est 0. La sortie de la convolution est ensuite multipliée par un ensemble de poids  $W$  et ajoutée à un biais  $b$  puis passé par une fonction d'activation non linéaire comme dans un réseau neuronal dense. Nous répétons cela avec des couches CNN supplémentaires, nous allons utiliser Dropout qui est une technique pour réduire le sur ajustement en supprimant aléatoirement certains nœuds. Une architecture CNN est formée par un empilement de couches de traitement indépendantes :

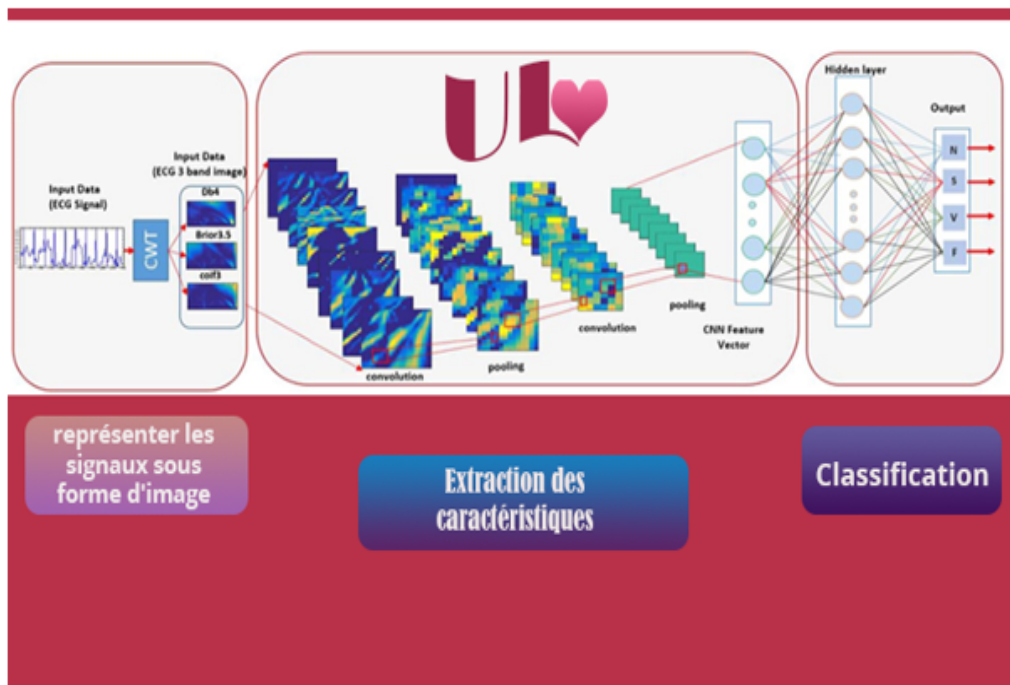


FIGURE 3.1 – Architecture de CNN

1. **La couche de convolution** Quand on lui présente une nouvelle image, le CNN ne sait pas exactement si les caractéristiques seront présentes dans l'image ou où elles pourraient être, il cherche donc à les trouver dans toute l'image et dans n'importe quelle position. En calculant

dans toute l'image si une caractéristique est présente, nous faisons un filtrage. Les mathématiques que nous utilisons pour réaliser cette opération sont appelés une convolution, de laquelle les réseaux de neurones à convolution tiennent leur nom. La couche de convolution est la composante clé des réseaux de neurones convolutifs, et constitue toujours au moins leur première couche. Son but est de repérer la présence d'un ensemble de features dans les images reçues en entrée.

Elle fonctionne comme un extracteur de caractéristiques des images. Une image est passée à travers une succession de filtres, ou noyaux de convolution. On obtient pour chaque paire (image, filtre) une carte d'activation, ou **feature map**, qui nous indique où se situent les features dans l'image : plus la valeur est élevée, plus l'endroit correspondant dans l'image ressemble à la feature [30].

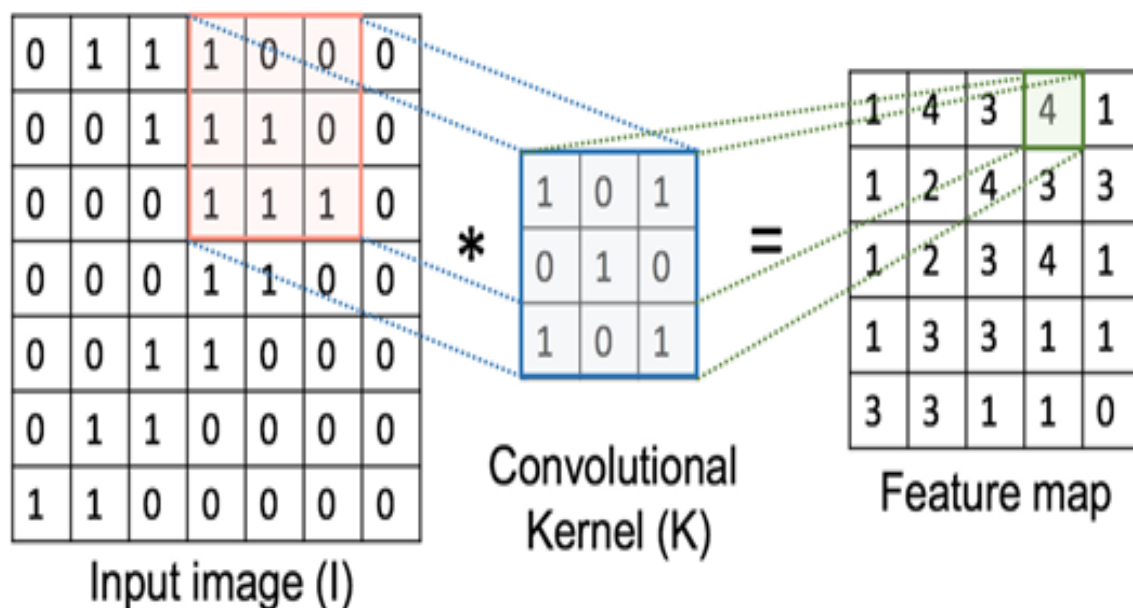


FIGURE 3.2 – Feature Map [30]

**Stride** : est le nombre de pixels décalés sur la matrice d'entrée. Lorsque la foulée est de 1, nous déplaçons les filtres à 1 pixel à la fois. Lorsque la foulée est de 2, nous déplaçons les filtres à 2 pixels à la fois et ainsi de suite [31].

**Padding** : Parfois, le filtre ne correspond pas parfaitement à l'image d'entrée. Nous avons deux options :

- Remplir l'image avec des zéros pour qu'elle s'adapte.
- Déposer la partie de l'image où le filtre ne correspondait pas. C'est ce qu'on appelle un remplissage valide qui ne conserve qu'une partie valide de l'image [31].

2. **Relu () Acronyme de Rectified Linear Unit (unité linéaire rectifiée)** : Pour les réseaux de neurone a convolution, une fois la convolution effectuée, une fonction d'activation

non-linéarité est appliquée sur toutes les valeurs de l'image filtrée. ReLU permet tout simplement de remplacer les résultats négatifs par zéro. La fonction ReLU est interprétée par la formule :  $f(z) = \max(0, z)$  [32].

**Remarque :** Il existe d'autres fonctions non linéaires telles que tanh ou sigmoid qui peuvent également être utilisées à la place de ReLU. La plupart des data scientiste utilisent ReLU car en termes de performances est meilleur que les deux autres [31].

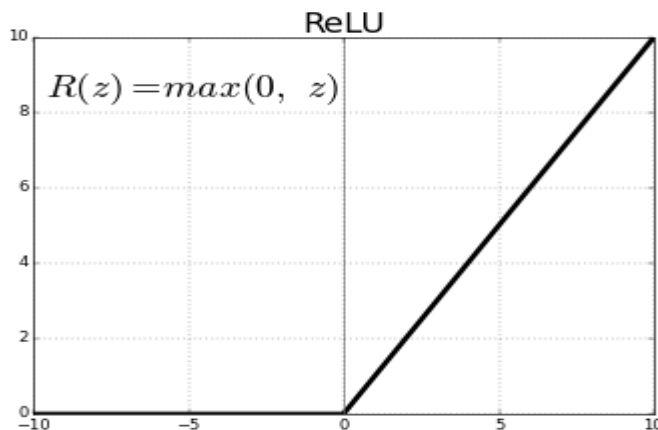


FIGURE 3.3 – Fonction Relu [31]

### 3. Couche d'entrée entièrement connectée (Pooling) :

La transformation ReLU est suivie d'une étape de pooling, dans laquelle le réseau de neurones convolutif sous-échantillonne la caractéristique convoluée (pour réduire le temps de traitement), diminuant ainsi le nombre de dimensions de la carte de caractéristiques tout en préservant les informations les plus critiques de la caractéristique. Un algorithme couramment utilisé pour ce processus est le pooling maximal [33]. Le pooling maximal fonctionne en déplacement sur la carte de caractéristiques et extraction des tuiles d'une taille spécifiée. Pour chaque tuile, la valeur maximale est transférée vers une nouvelle carte de caractéristiques et toutes les autres valeurs sont supprimées.

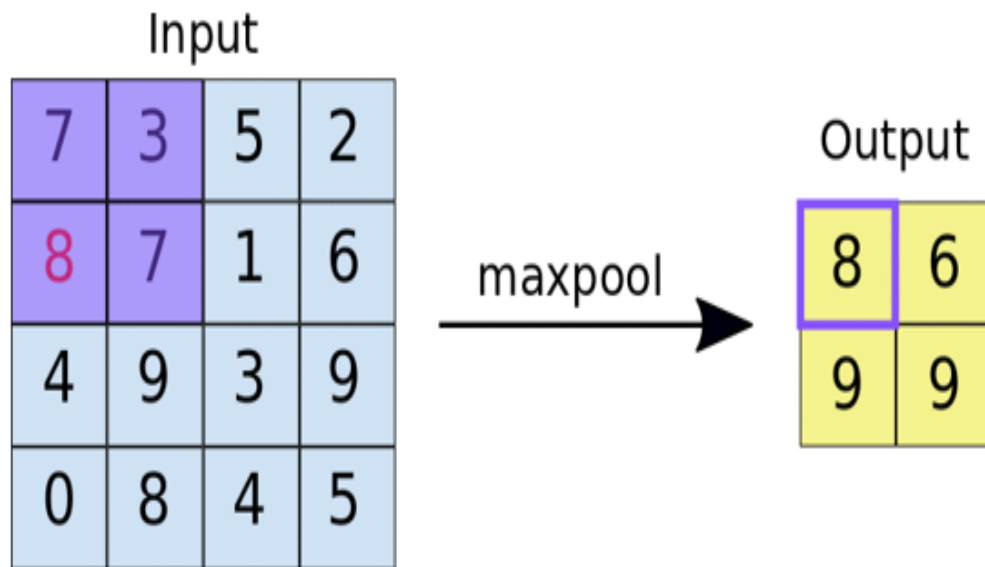


FIGURE 3.4 – Max pooling [33]

#### 4. Couche entièrement connectée (fully-connected) :

La couche entièrement connectée est un traditionnel perceptron multicouche (Multi Layer Perceptron) utilisant une fonction d'activation notamment appelée « softmax » dans la couche de sortie. Le terme « entièrement connecté » implique que chaque neurone dans la couche précédente est connecté à chaque neurone sur la couche suivante. La sortie des couches de convolution et de Pooling représente les fonctions de haut niveau de l'image d'entrée. Le but de la couche entièrement connectée est de pouvoir utiliser ces fonctions pour classer l'image d'entrée dans différentes classes en fonction de l'ensemble de données d'apprentissage [34].

#### Domaines d'application CNN

- CNNs sont à privilégier pour l'interprétation de données visuelles,
- incluent la détection des visages
- l'analyse médicale
- la découverte de médicaments et analyse images.

#### 3.3.1 L'entraînement d'un nouveau CNN

Créer un nouveau réseau de neurones convolutif est coûteux en termes d'expertise, de matériel et de quantité de données annotées nécessaires. Il s'agit d'abord de fixer l'architecture du réseau, c'est-à-dire le nombre de couches, leurs tailles et les opérations matricielles qui les connectent. L'entraînement consiste alors à optimiser les coefficients du réseau pour minimiser l'erreur de classification en sortie. Cet entraînement peut prendre plusieurs semaines pour les meilleurs CNN,

avec de nombreux GPU travaillant sur des centaines de milliers d'images annotées. Des équipes de recherche se spécialisent dans l'amélioration des CNN. Elles publient leurs innovations techniques, ainsi que le détail des réseaux entraînés sur des bases de données de références [35].

### **Quelques réseaux convolutifs célèbres**

- LeNet développée par Yann LeCun dans les années 1990.
- AlexNet développée par Alex Krizhevsky en 2012.
- Zfnet développée par Matthew Zeiler et Rob Fergus en 2013.
- GoogLeNet développée par Szegedy et al. De Google en 2014.
- ResNet développé par Kaiming He et al en 2015.

### **Avantages de CNNs**

Un avantage majeur des réseaux convolutifs est l'utilisation d'un poids unique associé aux signaux entrant dans tous les neurones d'un même noyau de convolution. Cette méthode réduit l'empreinte mémoire, améliore les performances et permet une invariance du traitement par translation. C'est le principal avantage du CNN par rapport au MLP, qui lui considère chaque neurone indépendant et donc affecte un poids différent à chaque signal entrant. Lorsque le volume d'entrée varie dans le temps (vidéo ou son), il devient intéressant de rajouter un paramètre de temporisation (delay) dans le paramétrage des neurones. On parlera dans ce cas de réseau neuronal à retard temporel (TDNN).

Comparés à d'autres algorithmes de classification de l'image, les réseaux de neurones convolutifs utilisent relativement peu de pré-traitement. Cela signifie que le réseau est responsable de faire évoluer tout seul ses propres filtres (apprentissage sans supervision), ce qui n'est pas le cas d'autres algorithmes plus traditionnels. L'absence de paramétrage initial et d'intervention humaine est un atout majeur des CNN [36].

```

from tensorflow.python.keras.layers import MaxPooling2D #maxpool
from tensorflow.python.keras.layers import Flatten #aplatir dans un vecteur vertical
from tensorflow.python.keras.layers import Dense #Ajouter des couches complètement connecté

#Initialiser le rx neuronne a conv
classfier = Sequential()

#Ajouter la couche de conv
classfier.add(Conv2D(32, (3, 3), input_shape = (150, 150, 3), activation = 'relu'))

# Step 2 - Pooling
classfier.add(MaxPooling2D(pool_size = (2, 2)))
#ajouter une nouvelle couche de conv
classfier.add(Conv2D(32, (3, 3), activation = 'relu'))
classfier.add(MaxPooling2D(pool_size = (2, 2)))

#ajouter une deuxieme couche de conv
classfier.add(Conv2D(64, (3, 3), activation = 'relu'))
classfier.add(MaxPooling2D(pool_size = (2, 2)))

#ajouter une 3eme couche de conv
classfier.add(Conv2D(64, (3, 3), activation = 'relu'))
classfier.add(MaxPooling2D(pool_size = (2, 2)))

# Step 3 - Flattening
classfier.add(Flatten())

# Step 4 - Full connection
classfier.add(Dense(units = 128, activation = 'relu'))
classfier.add(Dense(units = 128, activation = 'relu'))
classfier.add(Dense(units = 128, activation = 'relu'))
classfier.add(Dropout(0.3))#pour regler le probleme du sur entrainement
classfier.add(Dense(units = 1, activation = 'sigmoid'))#sigmoid car cest une classification et un seul rx neuronne

# Compiling the CNN
classfier.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = ['accuracy'])

# Part 2 - Fitting the CNN to the images

from tensorflow.python.keras.preprocessing.image import ImageDataGenerator #augmentation d'images pour eviter le surEntrainement
#Augmenter notre jeu d'entrainement

train_datagen = ImageDataGenerator(rescale = 1./255,
                                   shear_range = 0.2,#transvection
                                   zoom_range = 0.2,
                                   horizontal_flip = True)

test_datagen = ImageDataGenerator(rescale = 1./255)

```

FIGURE 3.5 – Implémentation du code de CNN

Le code ci-dessus (Figure 3.7), se constitue de 5 méthodes :

- Conv2D : Cette méthode crée un noyau de convolution qui est convolué avec l'entrée de la couche pour produire un tenseur de sorties, elle prend deux paramètres le premier est le nombre de filtre et le second est la taille de filtre. Par exemple, dans la première couche de convolution, nous créons 32 filtres de taille 3x3. Nous utilisons aussi la non-linéarité relu comme activation.
- MaxPooling2D : cette méthode crée une couche maxpool, le seul paramètre est la taille de la fenêtre. Nous utilisons une fenêtre 2x2 car c'est la plus courante. Par défaut, la longueur de foulée est égale à la taille de la fenêtre, qui est de 2 dans notre cas, donc nous ne changeons



pas cela.

- Flatten : aplatit les sorties générées par les couches précédentes pour les transformer en un seul vecteur qui peut être utilisé comme entrée pour la couche suivante.
- Dense (128) : est une couche avec 128 unités cachées, et on a utilisée Relu comme activation.
- ImageDataGenerator : C'est une classe de la bibliothèque d'apprentissage en profondeur Keras, elle est utilisée pour augmenter artificiellement la taille d'un ensemble de données d'entraînement en créant des versions modifiées des images dans l'ensemble de données. Cette stratégie est utilisée pour étendre le jeu de données d'entraînement afin d'améliorer les performances et la capacité du modèle à se généraliser.

## 3.4 LSTM

### 3.4.1 RNN (recurrent neural networks)

Les réseaux neuronaux multicouches simples sont des classificateurs qui, lorsqu'ils sont donnés une certaine entrée, marquent l'entrée comme appartenant à l'une des nombreuses classes. Ils sont formés à l'aide des algorithmes de rétro propagation existants. Ces réseaux sont excellents dans ce qu'ils font, mais ils ne sont pas capables de gérer les entrées qui viennent dans une séquence. Par exemple, pour qu'un réseau de neurones identifie les noms dans une phrase, avoir uniquement le mot comme entrée n'est pas du tout utile. Beaucoup d'informations sont présentes dans le contexte du mot qui ne peuvent être déterminées qu'en regardant les mots près du mot donné. La séquence entière doit être étudiée pour déterminer la sortie. C'est là que les réseaux neuronaux récurrents (RNN) trouvent leur utilisation. Au fur et à mesure que le RNN traverse la séquence d'entrée, la sortie de chaque entrée devient également une partie de l'entrée de l'élément suivant de la séquence. Les RNN sont très adaptés aux problèmes de classification de séquence et la raison pour laquelle ils sont si bons dans ce domaine est qu'ils sont capables de conserver des données importantes des entrées précédentes et d'utiliser ces informations pour modifier la sortie actuelle.

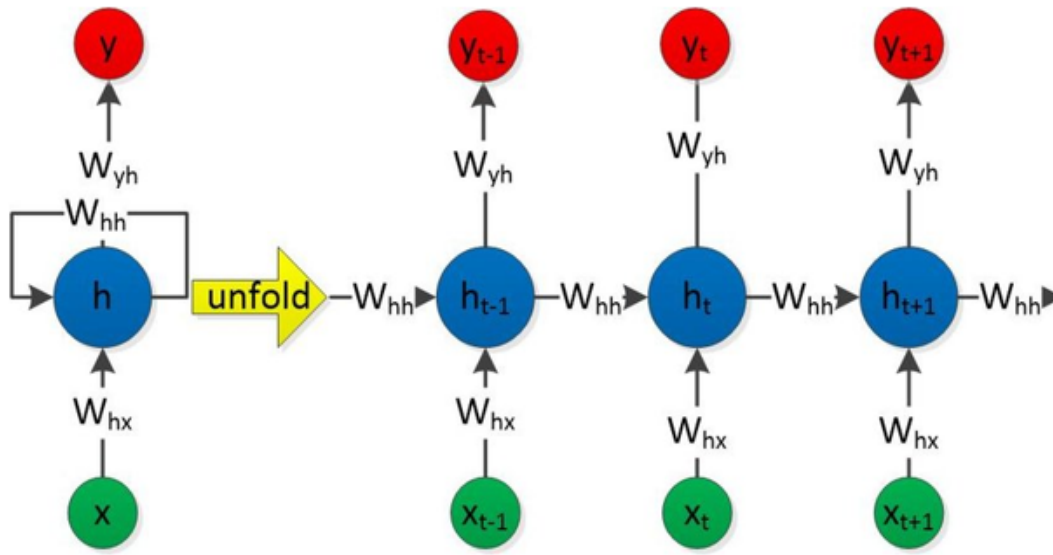


FIGURE 3.6 – Architecture du RNN [36]

### Domaines d'application RNN

Les RNN ont connu un grand succès dans de nombreuses tâches de traitement du langage naturel. Les plus grands exploits des RNN ont été accomplis par l'architecture LSTM car ils sont bien meilleurs pour capturer des dépendances à long terme.

- la traduction
- l'extraction d'entités
- l'intelligence conversationnelle
- l'analyse des sentiments et l'analyse de la parole

### Avantages de RNN

- Possibilité de traiter une entrée de n'importe quelle longueur
- La taille du modèle n'augmente pas avec la taille de l'entrée
- Le calcul prend en compte les informations historiques
- Les poids sont partagés dans le temps

#### 3.4.1.1 Le problème La descente de gradient

La descente de gradient : est un algorithme d'optimisation souvent utilisé pour trouver les poids ou les coefficients des algorithmes d'apprentissage automatique, tels que les réseaux de neurones artificiels et la régression logistique.

Cela fonctionne en permettant au modèle de faire des prédictions sur les données d'apprentissage et en utilisant l'erreur sur les prédictions pour mettre à jour le modèle de manière à réduire l'erreur.

Le but de l'algorithme est de trouver des paramètres de modèle (par exemple, des coefficients ou des poids) qui minimisent l'erreur du modèle sur le jeu de données d'apprentissage.

Pour ce faire, il modifie le modèle en le déplaçant le long d'une pente d'erreur vers une valeur d'erreur minimale. Cela donne à l'algorithme le nom de « descente de gradient » [37].

Il existe trois variantes de cette méthode :

- Batch gradient descent, de gradient stochastique,
- Mini-batch gradient descent.

**Problème :** 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. C'est problématique parce que la signification d'une phrase est souvent déterminée par des mots qui ne sont pas très proches. Dans ce cas précis si la fonction d'activation est une tanh ou bien une sigmoïde alors nous aurons le problème du vanishing gradient.

Le problème de la descente de gradient a été découvert pour la première fois par Sepp (Joseph) Hochreiter en 1991 [38].

**Solutions [39] :**

- Matériel plus rapide : le temps de compilation plus rapide lors du passage des processeurs aux GPU a rendu possible la méthode de rétropropagation standard quand le coût du modèle est très inférieur.
- Autre fonction d'activation : Relu souffrent moins du problème de la descente de gradient, car ils ne saturent que dans une direction.
- Réseaux résiduels : les réseaux de neurones résiduels ou les resnets évitent le problème en construisant des ensembles de nombreux réseaux courts.
- LSTM (La mémoire à long terme).

### La mémoire à long court terme (LSTM)

La mémoire à long terme a été conceptualisé par Sepp Hochreiter et Jürgen Schmidhuber en 1997.[40] C'est est une architecture de réseau neuronal récurrent artificiel (RNN) largement utilisée dans l'apprentissage en profondeur. Ces réseaux classent, traitent et font des prédictions basées sur des ensembles de données chronologiques qui pourraient contenir des informations cachées entre des événements importants.

Contrairement aux réseaux de neurones à réaction standard, les LSTM ont des connexions de rétroaction. En plus de traiter des points de données uniques, comme des images, LSTM peut également traiter des séquences entières de données, telles que la parole et la vidéo. Ils peuvent également être appliqués à des tâches telles que la reconnaissance d'écriture manuscrite connectée et non segmentée. [41] Les LSTM résolvent le problème du gradient en introduisant quelques portes supplémentaires qui contrôlent l'accès à l'état de la cellule.

**Principe :** Les LSTMs ont des mécanismes internes appelés portes (Gates), ces portes servent à régulariser le flux d'information en apprenant quelles informations sont utiles à garder ou à oublier

dans le temps. Ce faisant, le réseau pourra transmettre des données pertinentes, et ceci, grâce à l'état de la cellule, agissant comme une route de transport qui transfère les informations relatives tout au long de la chaîne de séquence.

Au fur et à mesure que l'état se poursuit, des informations sont ajoutées ou supprimées grâce à trois portes principales. [42] La cellule est responsable de se souvenir valeurs sur des intervalles de temps arbitraires, et chacune des trois portes peut être considérée comme un neurone artificiel conventionnel, calculant une activation (en utilisant une fonction d'activation) d'une somme pondérée des données actuelles.

- Porte d'entrée : La porte d'entrée (Input Gate), Contrôle les nouvelles informations pour mettre à jour état de la cellule à partir de l'entrée actuelle. La fonction d'activation est sigmoïde, c'est-à-dire qu'elles produisent une valeur entre 0 ou 1 et dans la plupart des cas, elle est égale à 0 ou 1. Une coordonnée proche de 0 signifie que la coordonnée en position n'est pas importante. A l'inverse, une coordonnée proche de 1 sera jugée importante. Ce résultat est ensuite multiplié par état actuel.
- Porte d'oubli :  
La porte d'oubli (Forget Gate) décide quelles informations doivent être jetées ou conservées. Les informations de l'état précédent sont passés à la fonction Tanh donnant des valeurs entre -1 et 1. Ensuite, afin de sélectionner les caractéristiques importantes, une couche sigmoïde va décider quelle valeur va être mise à jour [43].
- Porte de sortie :  
Enfin, nous avons la porte de sortie (Output Gate), doit décider de quel sera le prochain état caché, qui contient des informations sur les entrées précédentes du réseau et sert aux prédictions dans une fonction sigmoïde [43].

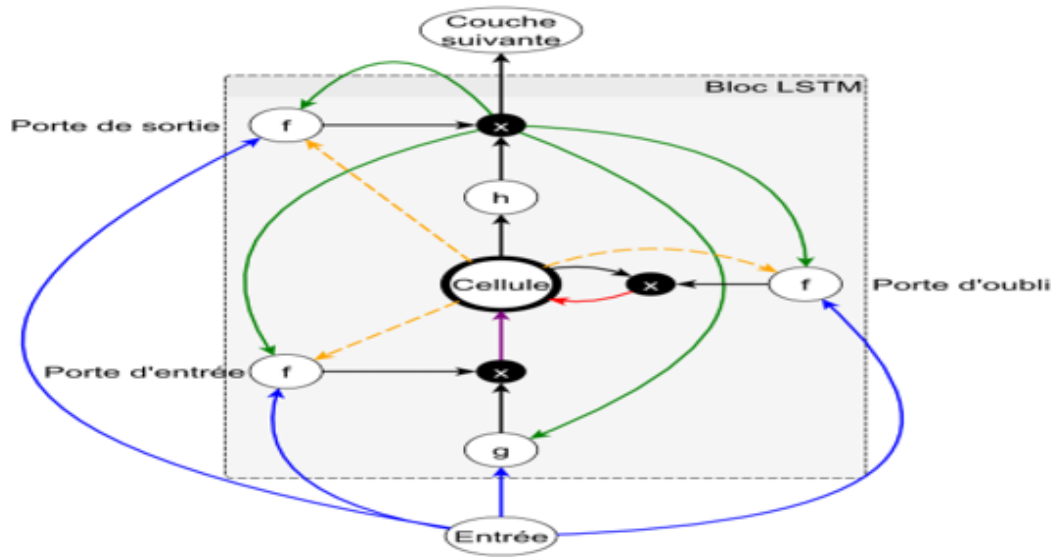


FIGURE 3.7 – Schéma d'un neurone LSTM comprenant une mémoire interne contrôlée par les trois portes « Schéma issu de la thèse de Bruno Stuner (2018) » [43]

### Normalisation des lots :

L'entraînement des LSTM est compliqué par le fait que la distribution statistique des entrées de chaque couche change pendant l'entraînement, à mesure que les paramètres des couches précédentes changent. Cela ralentit la formation en exigeant des taux d'apprentissage plus faibles et une initialisation minutieuse des paramètres, et rend extrêmement difficile la formation de modèles avec des non-linéarités saturantes. La normalisation des lots a récemment été introduite pour surmonter ce problème en normalisant le  $X_t$  et  $h_{t-1}$  activations entrant dans chaque couche en appliquant un décalage de covariable. Cela renforce les moyens et les écarts de  $X_t$  et  $h_{t-1}$  être invariant aux changements dans les distributions de paramètres des couches sous-jacentes et Découple efficacement les paramètres de chaque couche de ceux des autres couches, conduisant à un problème d'optimisation mieux conditionné [42].

```

from tensorflow.python.keras.models import Sequential
from tensorflow.python.keras.layers import Dense
from tensorflow.python.keras.layers import LSTM
from tensorflow.python.keras.layers import Dropout

regressor = Sequential()
#Couche LSTM & Dropout
regressor.add(LSTM(units=50,return_sequences=True,input_shape=(x_train.shape[1],1)))
regressor.add(Dropout(0.2))

regressor.add(LSTM(units=50,return_sequences=True))
regressor.add(Dropout(0.2))

regressor.add(LSTM(units=50,return_sequences=True))
regressor.add(Dropout(0.2))

regressor.add(LSTM(units=50,return_sequences=False))
regressor.add(Dropout(0.2))

#Couche de sortie
regressor.add(Dense(units=1))

#
# Compilation
#
regressor.compile(optimizer="adam",loss="mean_squared_error")

#
# Entraînement du rx
#

regressor.fit(x_train,y_train,epochs=100,batch_size=32)#32 observation on va calculer la fct de cout puis on
#
#

```

FIGURE 3.8 – Exemple d’une implémentation de LSTM

La figure ci-dessus nous montre l’étape de formation du modèle, les données sont transmises au réseau neuronal et entraînées pour la prédiction attribuant des biais et des poids aléatoires. Notre modèle LSTM est composé d’une couche d’entrée séquentielle suivie de 3 couches LSTM et d’une couche dense avec activation et enfin d’une couche de sortie dense avec fonction d’activation linéaire. Le type d’optimiseur utilisé peut grandement affecter la vitesse à laquelle l’algorithme converge vers la valeur minimale. Aussi, il est important qu’il y ait une certaine notion d’aléatoire pour éviter de rester coincé dans un minimum local et ne pas atteindre le minimum global. Après la phase de formation de modèle, vient l’étape de génération de sortie. Dans cette couche, la valeur de sortie générée par la couche de sortie du RNN est comparée à la valeur cible. L’erreur ou la différence entre la cible et la valeur de sortie obtenue est minimisée en utilisant un algorithme de rétro-propagation qui ajuste les poids et les biais du réseau.

- **Dropout** : est une méthode de régularisation la plus populaire pour les réseaux de neurones profonds utilisé pour éviter le sur-apprentissage, consiste à sélectionner de façon aléatoire les nœuds pendant l’entraînement c’est-à-dire certains nœuds sont sélectionnés au hasard et leurs sorties sont mises à zéro pour désactiver les nœuds, la figure suivante explique comment cela fonctionne. Elle permet également d’améliorer considérablement les résultats et la performance des réseaux de neurones.

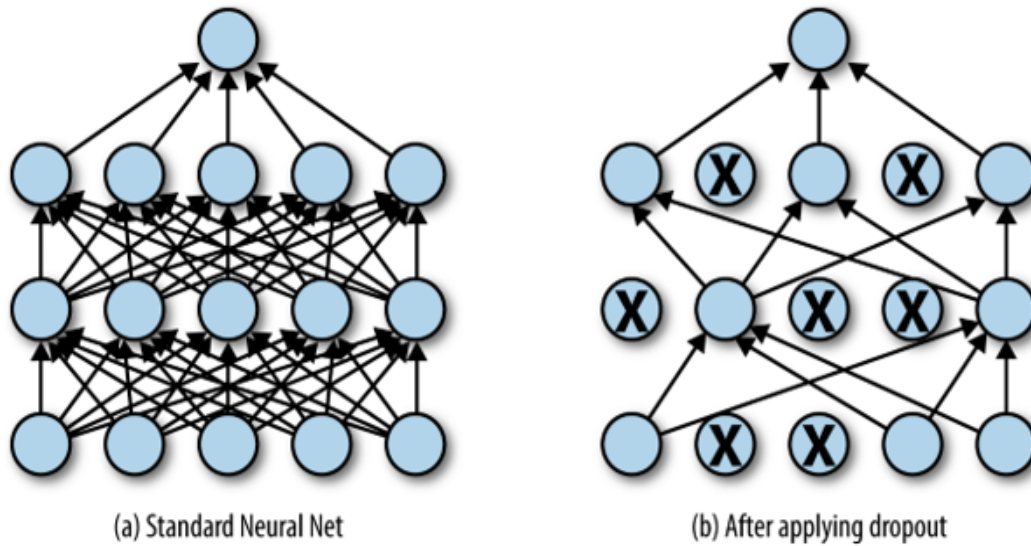


FIGURE 3.9 – Régularisation Dropout dans le deep learning [43]

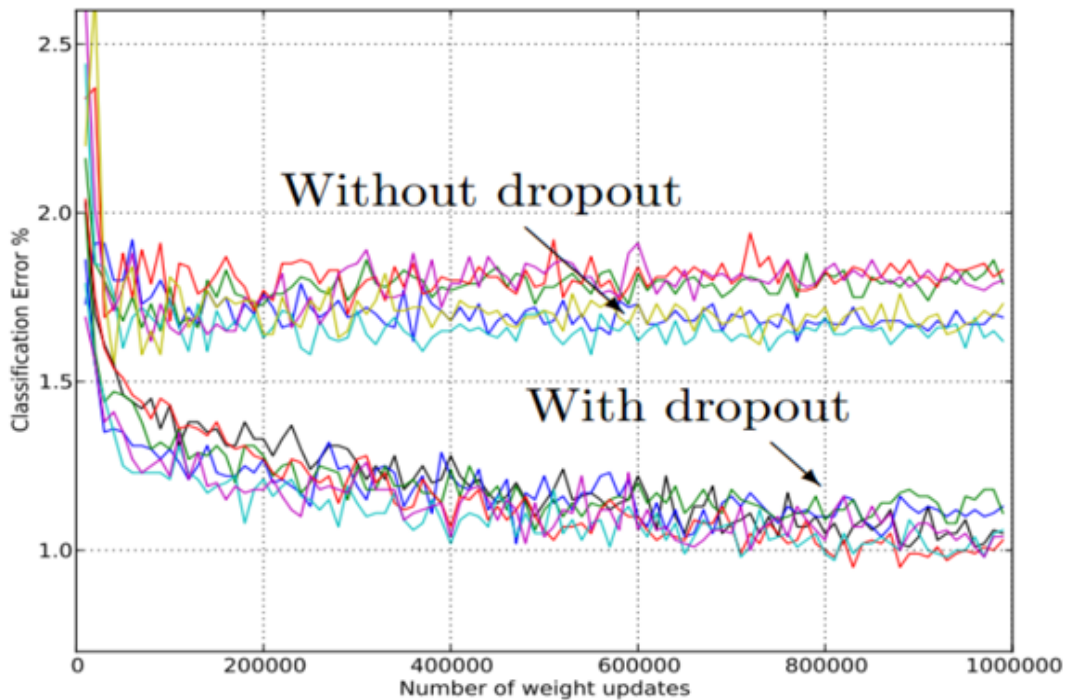


FIGURE 3.10 – Classification des erreurs avec ou sans Dropout [43]

— Le sur-apprentissage (over-fitting) :

Il se produit lorsque vous réalisez un bon ajustement de votre modèle sur les données d'entraînement, alors qu'il ne se généralise pas bien sur les nouvelles données invisibles. En d'autres

termes, le modèle a appris des modèles spécifiques aux données d'apprentissage, qui ne sont pas pertinents dans d'autres données [43].

### 3.5 CNN pour la détection d'anomalies

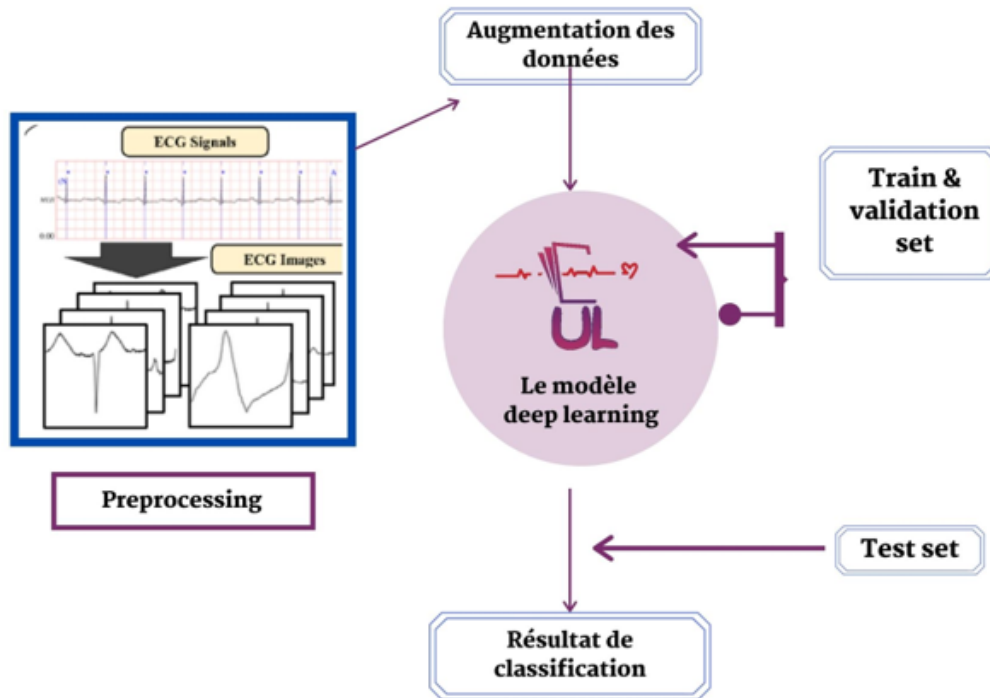


FIGURE 3.11 – Architecture du modèle en utilisant le CNN

Étant donné que le modèle CNN gère une 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 ECG obtenues, la classification de sept types d'ECG est effectuée à l'étape du classificateur CNN. Les sept classes sont :

Contraction auriculaire prématurée, normale, bloc de branche gauche, battement stimulé, contraction ventriculaire prématurée, bloc de branche droit et battement d'échappement ventriculaire.

Après l'étape de transformation des signaux ECG en images 1D, nous allons finaliser la phase de preprocessing en augmentant nos données, nous effectuons les phases de training et validation de notre dataset puis l'étape de Test qui nous permet d'avoir les résultats de la classification.

### 3.6 LSTM pour la détection d'anomalies

La chaîne typique de traitement des signaux, en particulier les ECG, est composée des phases suivantes :



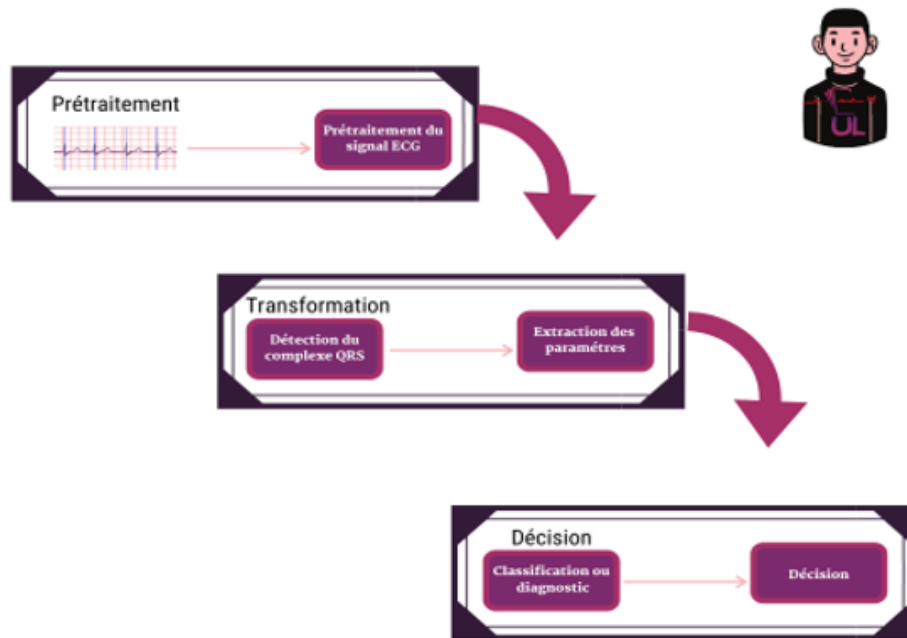


FIGURE 3.12 – Architecture du modèle en utilisant LSTM

- Prétraitement ou conditionnement des données.
- Transformation, c'est la phase la plus délicate de la chaîne puisque le signal subit une série de transformation afin de le préparer à l'étape de prise de décision.
- Décision (classification).

La figure 2 montre les étapes de traitement pour l'analyse d'un signal ECG. Le graphe contient le filtrage, la détection du complexe QRS et l'extraction des caractéristiques, et la classification.

Nous avons baptisé notre système « UL » qui signifie le cœur.

### 3.7 Conclusion

Dans ce chapitre, nous avons décrit la démarche suivie dans le but d'obtenir le meilleur modèle de détection d'anomalie de rythme cardiaque. Pour réaliser notre architecture nous avons commencé en premier lieu par définir notre première méthode le modèle CNN simple qui apporte des solutions à plusieurs problèmes de reconnaissance d'image, et il est activement utilisé dans le domaine médical en raison de son efficacité, et on a expliqué ses différentes couches et quelques méthodes ensuite nous avons cité ses domaines d'application, ses avantages et quelques réseaux convolutifs célèbres. En deuxième lieu nous avons défini notre seconde méthode à savoir le modèle

LSTM dont on parle de ses différentes portes, mais avant nous avons parlé sur RNN ses avantages et de la descende de gradient.

# Chapitre 4

## Expérimentation

### 4.1 Introduction

La classification de l'arythmie cardiaque est une étape importante dans les dispositifs de surveillance médicale. En effet, le signal ECG est un indicateur continu du fonctionnement du cœur et sa surveillance est très utile pour détecter toute déviation fonctionnelle du cœur. Au fur et à mesure que les ECG passaient de l'analogique au numérique, l'analyse automatisée par ordinateur des électrocardiogrammes standard à 12 dérivations a gagné en importance dans le processus de diagnostic médical. Cependant, les performances limitées des algorithmes classiques excluent leur utilisation en tant qu'outil de diagnostic autonome et les relèguent à un rôle auxiliaire[44].

Les réseaux de neurones profonds ont récemment obtenu un succès frappant dans des tâches telles que la classification d'images et la reconnaissance vocale, et les attentes sont grandes quant à la manière dont cette technologie peut améliorer les soins de santé et la pratique clinique. Jusqu'à présent, les applications les plus réussies utilisaient une configuration d'apprentissage supervisé pour automatiser le diagnostic des examens.

Dans ce chapitre, nous nous focalisons sur la mise en place d'un réseau neuronal convolutif CNN et un autre récurrent RNN. Le cas de l'ECG diffère des applications où les réseaux convolutifs sont habituellement utilisés, à savoir : la reconnaissance d'image. Dans ce dernier cas, les données sont toujours présentées sous forme de données 2D avec certains canaux de couleur, contrairement aux séries temporelles où les données sont 1D, nous verrons en détails comment créer un modèle qui prédit les arythmies de chaque battement de cœur en utilisant le deep learning. Enfin, nous présentons les résultats obtenus sur la base de données publique MIT-BIH Arrhythmia.

## 4.2 Présentation des outils

### 4.2.1 Outils logiciels

#### 4.2.1.1 Environnement

- **Anaconda** : est un gestionnaire de packages, une distribution gratuite et open-source des langages de programmation Python et R pour le calcul scientifique qui vise à simplifier le package gestion et déploiement [45].
- **Spyder** : est un environnement de développement pour Python. Libre et multiplateforme, il intègre de nombreuses bibliothèques d'usage scientifique : Matplotlib, NumPy, SciPy et IPython [45].
- **Jupyter-notebook** : Le bloc-notes Jupyter est un environnement informatique interactif qui permet aux utilisateurs de créer des documents de bloc-notes comprenant : du code en direct, des widgets interactifs, des tracés, du texte narratif, des équations, des images, de la vidéo [45].

#### 4.2.1.2 Framework

- **Tensorflow** : Il fut créé par l'équipe Google Brain pour mener des recherches sur le ML et le Deep Learning. Il est considéré comme une version moderne de Theano [45].
- **Keras** : Le Framework le plus haut niveau, le plus convivial de la liste. Il permet aux utilisateurs de choisir si les modèles qu'ils construisent sont exécutés sur Theano ou TensorFlow. Il est écrit et entretenu par Francis Chollet, un autre membre de l'équipe Google Brain [45].

#### 4.2.1.3 Bibliothèques utilisées

- **Numpy** : Est une bibliothèque permettant d'effectuer des calculs numériques avec Python. Elle introduit une gestion facilitée des tableaux de nombres, des fonctions sophistiquées (diffusion), on peut aussi l'intégrer le code C / C ++ et Fortran [48].
- **Pandas** : est une bibliothèque open source sous licence BSD fournissant des structures de données hautes performances et faciles à utiliser, ainsi que des outils d'analyse des données pour le langage de programmation Python [47].
- **Matplotlib** : est une bibliothèque de traçage pour le langage de programmation Python et son extension mathématique numérique NumPy. Il fournit une API orientée objet permettant d'incorporer des graphiques dans les applications[47].
- **Scikit-learn** : est une bibliothèque développée en Python, un langage de programmation de haut niveau. Elle est dédiée à l'apprentissage statistique (machine learning) et peut être utilisée comme middleware, notamment pour des tâches de prédiction[48].
- **WFDB** : Package natif de la base de données de formes d'ondes Python (WFDB). Une bibliothèque d'outils pour lire, écrire et traiter les signaux et annotations WFDB. Les principaux

composants de ce package sont basés sur les spécifications WFDB d'origine[46]. Ce package ne contient pas exactement les mêmes fonctionnalités que le package WFDB d'origine. Il vise à implémenter autant de ses fonctionnalités principales que possible, avec des API conviviales. Des outils supplémentaires utiles de traitement du signal physiologique sont ajoutés au fil du temps.

- **Seaborn** :est une bibliothèque de visualisation de données Python basée sur matplotlib. Il fournit une interface de haut niveau pour dessiner des graphiques statistiques attrayants et informatifs. Pour une brève introduction aux idées derrière la bibliothèque, vous pouvez lire les notes d'introduction.
- **Warning** :ne sont pas des erreurs mais des soupçons d'erreurs. Le programme peut continuer mais il est possible qu'il s'arrête un peu plus tard et la cause pourrait être un warning déclenché un peu plus tôt[49].

#### 4.2.1.4 Langages de programmation

- **Python** :permet dans l'industrie de piloter des systèmes complexes, mais il se répand aussi dans le monde de l'éducation, du lycée à l'université, par sa simplicité et ses bibliothèques logicielles, mathématiques notamment. Ce livre présente l'essentiel du langage puis, à-travers 130 exercices et problèmes, la seconde partie propose un entraînement au codage[48]. Partant de la base algorithmique classique, il aborde ensuite des secteurs moins enseignés comme le graphisme, l'analyse lexicale et syntaxique, les moteurs physiques, l'intelligence artificielle (génétique, neurones), et la musique électronique.

## 4.3 Classification avec Réseau Convolutif

Un réseau neuronal convolutif peut être considéré comme est un perceptron multicouche conçu spécifiquement pour reconnaître des formes avec un degré élevé d'invariance à la translation, à la mise à l'échelle, au biais et à d'autres formes de distorsion. Cette tâche difficile est apprise d'une manière supervisée au moyen d'un réseau dont la structure comprend les fonctions suivantes[47].

**Extraction de caractéristiques** : Chaque neurone tire ses entrées synaptiques d'un champ réceptif local de la couche précédente, l'obligeant ainsi à extraire des caractéristiques locales. Une fois qu'une caractéristique a été extraite, son emplacement exact devient moins important, tant que sa position par rapport à d'autres caractéristiques est approximativement préservée[47].

**Mappage des caractéristiques** : Chaque couche de convolution est composée de plusieurs vecteurs de caractéristiques, chaque vecteur de caractéristiques se présentant sous la forme d'un plan dans lequel les neurones sont contraints de partager le même.

Les expérimentations ont tous été effectuées sur une machine qui offre des performances acceptables dont voici les caractéristiques :

CPU	Intel® Core™ i7-8565U CPU 1.80GHz 1.99GHz
RAM	8 GO
GPU	NVIDIA Geforce MX150 4GB

## 4.4 La base de données MIT-BIH Arrhythmia

La base de données des arythmies MIT-BIH contient 48 extraits d'une demi-heure d'enregistrements d'ECG ambulatoires à deux canaux, obtenus de 47 sujets étudiés par le laboratoire d'arythmie BIH entre 1975 et 1979. Vingt-trois enregistrements ont été choisis au hasard parmi 4000 enregistrements ECG ambulatoires par heure collectés auprès d'une population hétérogène de patients hospitalisés (environ 60%) et ambulatoires (environ 40%) à l'hôpital Beth Israël de Boston. Les 25 enregistrements restants ont été sélectionnés dans le même ensemble pour inclure des arythmies moins courantes mais cliniquement significatives qui ne généralement pas bien représentées dans des petits échantillons.

Les enregistrements ont été numérisés à 360 échantillons par seconde et par canal avec une résolution de 11 bits sur une plage de 10 mV. Deux cardiologues au moins annotés indépendamment chaque enregistrement ; les désaccords ont été résolus pour obtenir les annotations de référence lisibles par ordinateur pour chaque battement (environ 110 000 annotations au total) incluses dans la base de données[50].

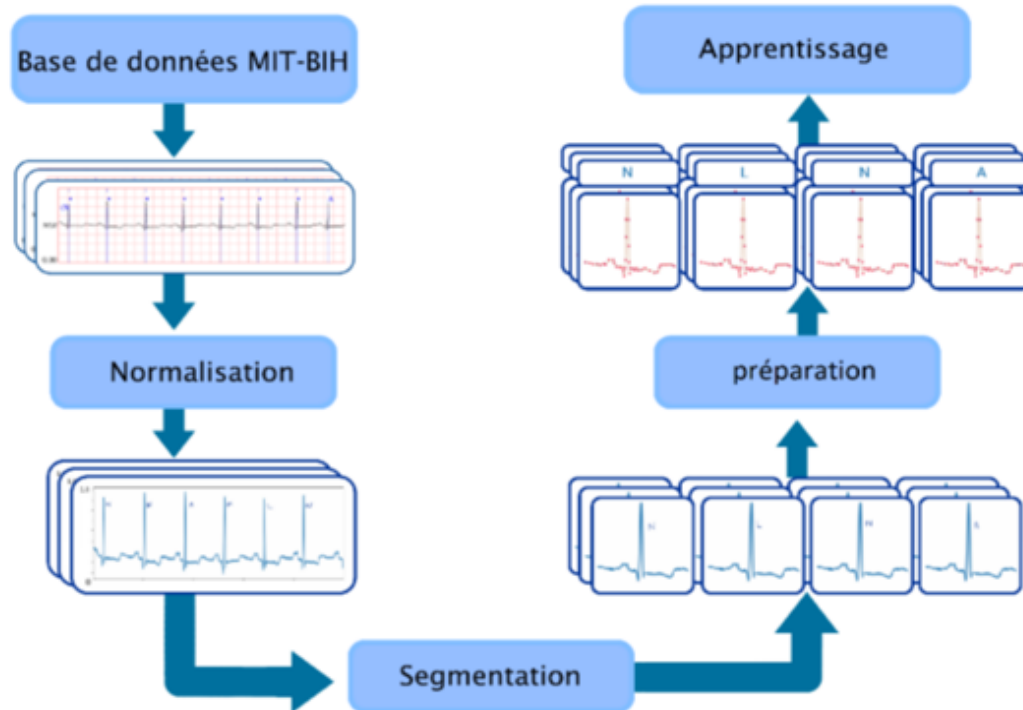


FIGURE 4.1 – Processus de traitement et de préparation des enregistrements.

La figure 1 montre le diagramme de prétraitement qui a été utilisé sur les données avant d’entraîner le réseau de neurones.

## 4.5 Techniques d’échenillage

Les techniques de sampling sont utilisées pour ajuster la distribution des classes d’un corpus. On retrouve deux catégories principales : le sur-échantillonnage et sous-échantillonnage (l’oversampling et l’undersampling). Cependant il existe aussi des hybridations entre le sur-échantillonnage et le sous-échantillonnage. Elles consistent à établir d’une manière simultanée l’augmentation des samples de la classe minoritaire par les approches de sur-échantillonnage et la diminution des samples de la classe majoritaire par les approches de sous-échantillonnage pour équilibrer la répartition des classes.

## 4.6 Fractionnement de données

La base de données MIT-BIH pourrait être téléchargée dans différents formats, adaptés à différents langages de programmation. Il comprend 48 enregistrements, chacun provenant d’un patient. Chaque enregistrement est constitué de deux fichiers, l’un pour le signal et l’autre pour les annotations de chaque battement cardiaque. Les annotations sont symbolisées par des lettres et leur signification est indiquée dans le Tableau 1. Lorsqu’il est extrait et traité, le signal avec des annotations ressemble à l’échantillon de la Figure 1. Avec chaque enregistrement dure plus d’une demi-heure, cette base de données est plus que suffisante pour l’apprentissage d’un réseau de neurones. Il est à noter qu’il y a des arythmies qui ne se sont pas produites assez souvent dans tous les enregistrements, ces cas rares ont été ignorés.

A	Contractions auriculaires prématurées	2694 fois
L	Bloc de branche gauche	8072 fois
N	Normale	46258 fois
/	Battement cardiaque irrégulier	7024 fois
V	Contractions ventriculaires prématurées	7130 fois
R	Bloc de branche droite	7256 fois

TABLE 4.1 – Types et occurrences des battements cardiaques dans le dataset MIT-BIH.

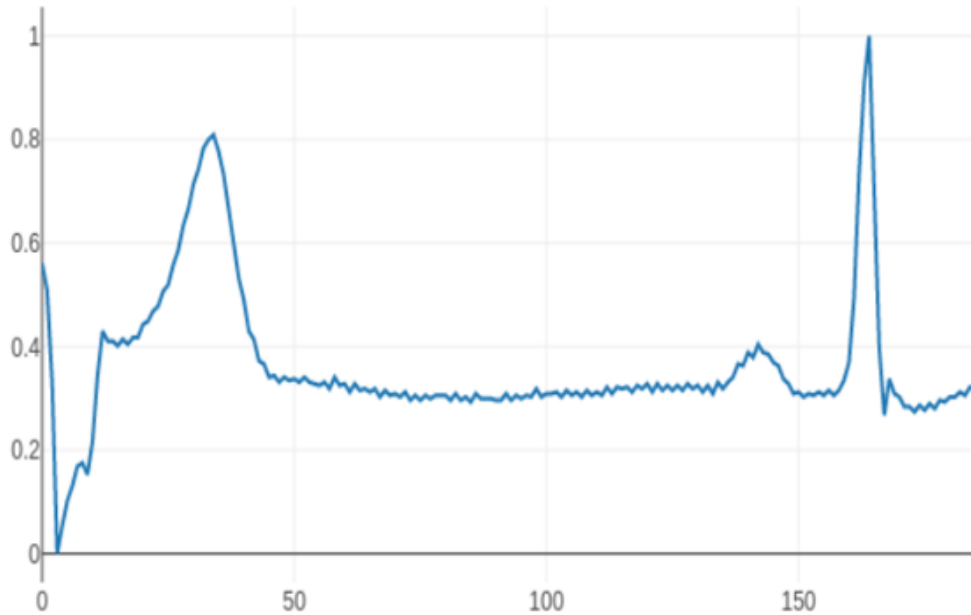


FIGURE 4.2 – Échantillon tracé à partir de la base de données MIT-BIH[51].

## 4.7 Modèle CNN

Nous allons répartir notre dataset selon la figure ci-dessous :

```
print("Train data")
print("Type\tCount")
print((mit_train_data[187]).value_counts())
print("-----")
print("Test data")
print("Type\tCount")
print((mit_test_data[187]).value_counts())
```

```
Train data
Type      Count
0.0      72471
4.0       6431
2.0       5788
1.0       2223
3.0         641
Name: 187, dtype: int64
-----
Test data
Type      Count
0.0     18118
4.0      1608
2.0      1448
1.0       556
3.0       162
Name: 187, dtype: int64
```

FIGURE 4.3 – Fractionnement de données de train et test.



CNN est un type spécial d’algorithme d’apprentissage en profondeur qui utilise un ensemble de filtres et l’opérateur de convolution pour réduire le nombre de paramètres. Cet algorithme a déclenché les techniques de pointe pour la classification d’images.

Essentiellement, la façon dont cela fonctionne pour 1D CNN est de prendre un filtre (noyau) de taille kernel-size en commençant par le premier horodatage. L’opérateur de convolution prend le filtre et multiplie chaque élément par les premiers pas de temps kernel-size. Ces produits sont ensuite additionnés pour la première cellule de la couche suivante du réseau neuronal. Le filtre se déplace ensuite par pas de temps et se répète. Stride par défaut dans Keras est 1, que nous utiliserons. Dans la classification des images, la plupart des gens utilisent le remplissage qui vous permet de sélectionner certaines fonctionnalités sur les bords de l’image en ajoutant des cellules “supplémentaires”, nous utiliserons le padding par défaut qui est 0. La sortie de la convolution est ensuite multipliée par un ensemble de poids  $W$  et ajouté à un biais  $b$ , puis passé par une fonction d’activation non linéaire comme dans un réseau neuronal dense.

La figure 3 nous montre le signal autour de l’un des battements anormaux que nous sommes parvenus à tracer en utilisant la bibliothèque matplotlib.

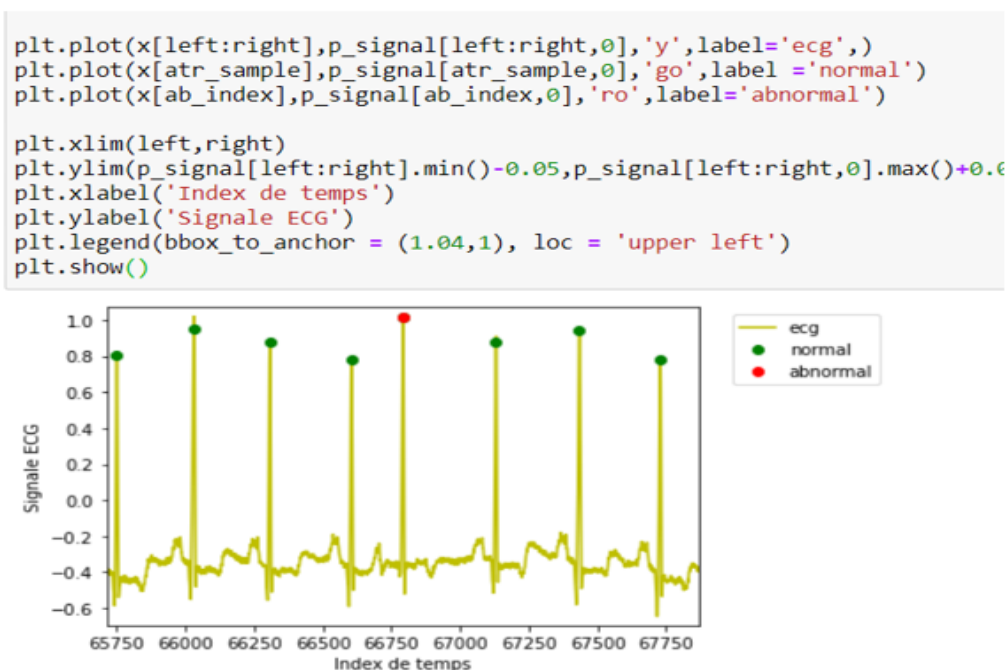


FIGURE 4.4 – Signale autour de l’un des battements anormaux.

### 4.7.1 Phase d’entraînements

La formation du modèle a été effectuée de manière similaire à la méthode de formation appliquée dans le modèle CNN.

Keras et TensorFlow nous proposent trois méthodes pour implémenter nos propres architectures de réseau neuronal :

- API séquentielle

- API fonctionnelle
- Sous-classification du modèle

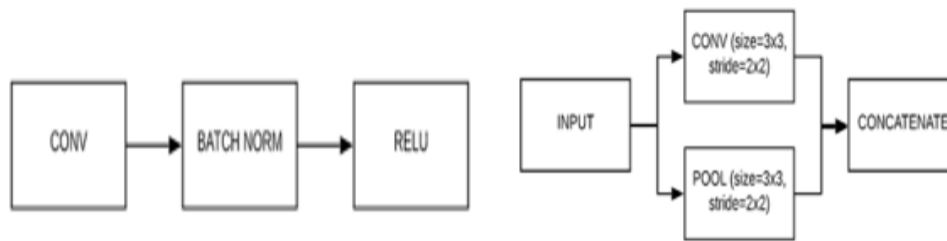


FIGURE 4.5 – API séquentielle et API fonctionnelle.

Afin de réaliser notre modèle, nous avons opté pour l'API Séquentielle, nous avons utilisé 100 EPOCHS, comme nous avons opté pour ADAM comme optimizer. L'exécution du modèle sur l'ensemble de test a permis d'obtenir une précision de 97,61%.

Ces résultats sont assez surprenants, car l'entraînement n'a pris que quelques minutes. Nous allons essayer de lui attribuer certains paramètres. Dans les réseaux de neurones, les paramètres sont utilisés pour former le modèle et faire des prédictions.

```

from keras.models import Sequential
from keras.layers import Dense, Activation

model = Sequential()

model.add(Dense(50, activation='relu', input_shape=(187,)))
model.add(Dense(50, activation='relu'))
model.add(Dense(5, activation='softmax'))

model.compile(optimizer='adam',
              loss='categorical_crossentropy',
              metrics=['accuracy'])

history = model.fit(X, y, validation_data=(testX, testy), epochs=100)

print("Evaluation: ")
mse, acc = model.evaluate(testX, testy)
print('mean_squared_error :', mse)
print('Précision:', acc)

```

Ac  
Ac

FIGURE 4.6 – Entraînement du modèle.

Il existe 2 types de paramètres :

- Paramètres : est interne au réseau de neurones. Il va évoluer durant l'ensemble du processus d'entraînement, lors de la back propagation .

- Hyperparamètres : est à l'inverse d'un paramètre, externe au processus d'entraînement, il en définit des propriétés. Il reste statique durant l'entraînement (ex : batch size, epoch, learning rate...).

La figure ci-dessus montre la première étape pour la réalisation de notre modèle qui consiste à l'entraînement du modèle, nous avons utilisé categorical-crossentropy comme perte car la sortie du modèle est catégorique. Pour la première et la deuxième couche nous avons implémenté 50 neurones, et pour la valeur d'input-shape qui représente le nombre de colonnes, nous avons mis 187.

Enfin, pour la dernière couche, nous avons 5 catégories à classifier, par conséquent nous avons implémenter 5 neurones pour la dernière couche.

```

Epoch 1/100
87554/87554 [=====] - 7s 85us/step - loss: 0.2576 - accuracy: 0.9292
Epoch 2/100
87554/87554 [=====] - 3s 33us/step - loss: 0.0400 - accuracy: 0.9871
Epoch 50/100
Epoch 100/100
87554/87554 [=====] - 3s 29us/step - loss: 0.0255 - accuracy: 0.9910

```

FIGURE 4.7 – Précision et perte tout au long de l'exécution du modèle.

Si nous faisons une comparaison de la (Figure6) nous constatons que lors du début d'apprentissage la perte vaut 0.2576 et la précision vaut 0.9292, lors de EPOCH 50 la perte vaut 0.04 tandis que la précision a atteint 0.98 et pour la dernière itération, nous avons pu atteindre une précision 0.991 et une perte de 0.02 , cela signifie que notre modèle a été bien entraîné et réponds d'ailleurs à la définition du réseau de neurones qui disait que plus le réseau de neurones est profond ,meilleur sont ses performances.

## 4.7.2 Phase de test

La figure 4.8 nous montre la phase de test, utiliser pour visualiser la précision et la perte du modèle. Nous avons aussi d'autres mesures qu'on peut calculer pour avoir une idée générale sur notre modèle CNN.

```

Entrée [11]: fig,ax = plt.subplots(1,2,figsize=(10,3))
              for i,met in enumerate(['accuracy','loss']):
                  ax[i].plot(history.history[met])
                  ax[i].plot(history.history['val_'+met])
                  ax[i].set_title('Modele {}'.format(met))
                  ax[i].set_xlabel('epochs')
                  ax[i].set_ylabel(met)
                  ax[i].legend(['train','val'])

              plt.show()

```

FIGURE 4.8 – Implémentation de la phase de Test

### 4.7.3 Critère d'évaluation

**Les faux positifs (FP) :** La valeur fausse positive ou false positive (FP), indique que la méthode détecte un QRS inexistant.

**Les faux négatifs (FN) :** La valeur fausse négative ou false négative (FN), indique que la méthode ne détecte pas les positions des complexes QRS.

— Accuracy =  $(tp+tn)/\text{toutes prédictions}$ .

Un rappel élevé, signifie que le ML a pu prédire correctement un bon nombre de documents pour chaque classe

— Précision =  $tp/(tp+fp)$ .

La précision est le nombre de cycles pertinents retrouvés rapporté au nombre de cycles total. Toutes les informations retournées superflus ou non pertinents constituent du bruit.

La précision s'oppose à ce bruit. Si elle est élevée, cela signifie que peu d'informations inutiles sont proposées par le classifieur et que ce dernier peut être considéré comme "précis".

— Recall || TruePositiveRate(TPR) =  $tp / (tp+fn)$ .

— specificity || TNR =  $TN / (Tn+Fp)$ .

— F1 Score =  $2 * (\text{precision} * \text{TPR}) / (\text{precision} + \text{TPR})$ .

Le F-score est la moyenne harmonique de la précision P.

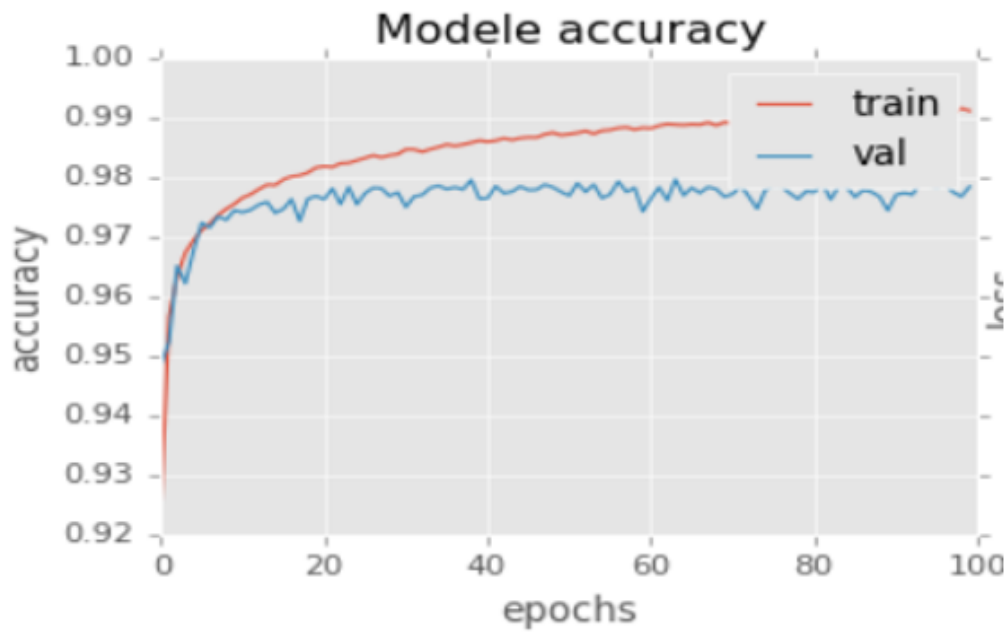


FIGURE 4.9 – Précision du modèle.

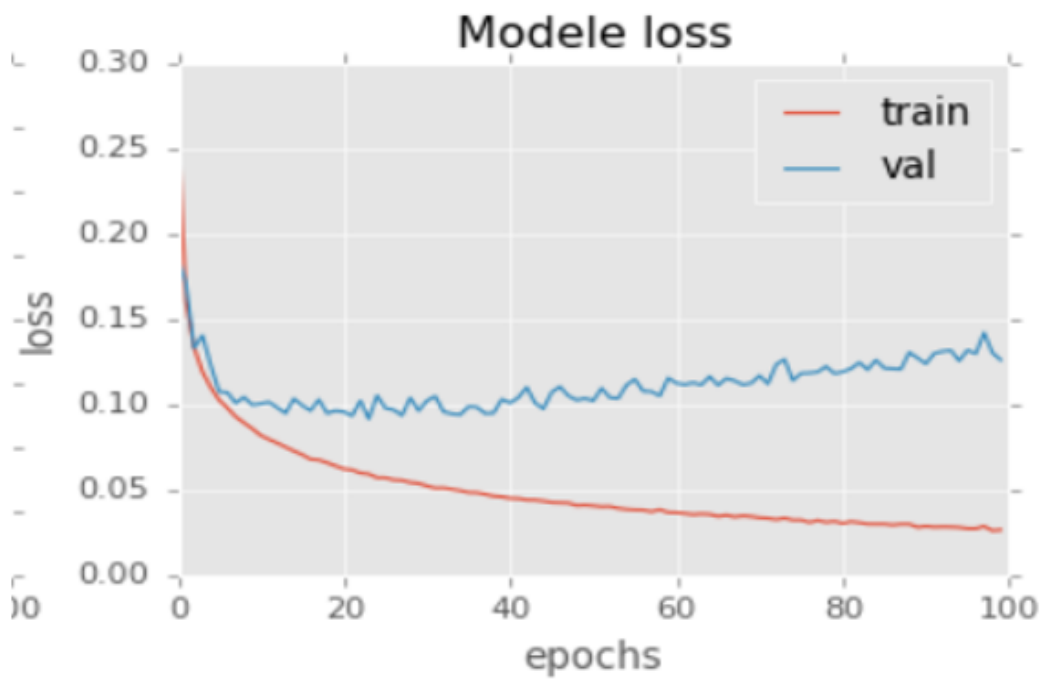


FIGURE 4.10 – Modèle de perte.

La modification de la précision et de l'entropie croisée catégorielle (appelée la perte) peut être

observée dans la figure 9, car le modèle est formé à travers les différentes époques. Les précisions de l'entraînement et de la validation augmentent progressivement au fil des EPOCH de la figure 9 jusqu'à ce qu'elles atteignent un certain seuil, indiquant qu'aucune autre amélioration n'est apportée et entraînant la fin de l'entraînement. De même, dans la figure 8 la perte de validation et de formation diminue au fur et à mesure que le modèle apprend, jusqu'à ce qu'il se termine finalement lorsque le modèle commence à se sur-ajuster.

## 4.8 Modèle LSTM sous Tensorflow

```

LSTM

Entrée [ ]:
Entrée [45]: from keras.layers import Bidirectional, LSTM
Entrée [46]: model = Sequential()
              model.add(Bidirectional(LSTM(64, input_shape=(X_train_cnn.shape[1], X_train_cnn.shape[2])))
              model.add(Dropout(rate = 0.25))
              model.add(Dense(1, activation = 'sigmoid'))
              model.compile(
                  loss = 'binary_crossentropy',
                  optimizer = 'adam',
                  metrics = ['accuracy'])

```

FIGURE 4.11 – Formation du modèle avec LSTM.

À cette étape, les données sont transmises au réseau neuronal et entraînées pour la prédiction attribuant des biais et des poids aléatoires. Notre modèle LSTM est composé d'une couche d'entrée séquentielle suivie de 3 couches LSTM et d'une couche dense avec activation et enfin d'une couche de sortie dense avec fonction d'activation linéaire. Les LSTM bidirectionnels sont une extension des LSTM traditionnels qui peuvent améliorer les performances du modèle sur les problèmes de classification de séquence.

Dans les problèmes où tous les pas de temps de la séquence d'entrée sont disponibles, les LSTM bidirectionnels entraînent deux LSTM au lieu d'un sur la séquence d'entrée. Le premier sur la séquence d'entrée tel quel et le second sur une copie inversée de la séquence d'entrée. Cela peut fournir un contexte supplémentaire au réseau et entraîner un apprentissage plus rapide et encore plus complet du problème.

Le type d'optimiseur utilisé peut grandement affecter la vitesse à laquelle l'algorithme converge vers la valeur minimale. Aussi, il est important qu'il y ait une certaine notion d'aléatoire pour éviter de rester coincé dans un minimum local et ne pas atteindre le minimum global. Il existe

quelques excellents algorithmes, mais nous avons choisi d'utiliser l'optimiseur Adam. L'optimiseur Adam combine les avantages de deux autres optimiseurs : ADAGRAD et RMSprop.

Un autre aspect important de la formation du modèle est de s'assurer que les pondérations ne deviennent pas trop importantes et de commencer à se concentrer sur un point de données, d'où le surajustement. Nous devrions donc toujours inclure une pénalité pour les poids importants (la définition de grand dépendrait du type de régulariseur utilisé).

### 4.8.1 Phase d'entraînement avec le modèle LSTM

Lors de la phase d'entraînements, l'erreur ou la différence entre la cible et la valeur de sortie obtenue est minimisée en utilisant un algorithme de rétro-propagation qui ajuste les poids et les biais du réseau.

L'étape de prétraitement implique :

- La discrétisation des données : partie de la réduction des données mais avec une importance particulière, en particulier pour les données numériques.
- Transformation des données : normalisation.
- Nettoyage des données : Cette étape consiste à remplir les valeurs manquantes.
- Intégration de données : intégration de fichiers de données.

Une fois l'ensemble de données transformé en un ensemble de données propre, l'ensemble de données est divisé en ensembles d'apprentissage et de test afin d'être évalué.

```
from keras.layers import Bidirectional, LSTM

# remodeler les entrées [samples, time steps, features = 1]
X_train_cnn = np.reshape(X_train, (X_train.shape[0], X_train.shape[1], 1))
X_valid_cnn = np.reshape(X_valid, (X_valid.shape[0], X_valid.shape[1], 1))

print(X_train_cnn.shape)
print(X_valid_cnn.shape)

(73096, 2160, 1)
(36003, 2160, 1)
```

FIGURE 4.12 – Préparation des entrées pour l'entraînement du modèle.

Comme les données ECG de chaque sujet sont divisées en séquences plus petites, nous devons d'abord prédire une étiquette pour chaque séquence. Dès que nous avons ces prédictions, nous pouvons les regrouper pour chaque patient et faire des prédictions au niveau du patient.

```
model.fit(X_train_cnn[:20000], y_train[:20000], batch_size = 32, epochs = 5, verbose = 1)
```

FIGURE 4.13 – **Entraînement du modèle.**

Les RNN ont des boucles dans le réseau pour garder un souvenir de ce qui s'est passé dans le passé. Cela permet au réseau de transmettre des informations des premières étapes de temps à des étapes de temps ultérieures qui seraient généralement perdues dans d'autres types de réseaux. Il y a essentiellement un terme supplémentaire pour cet état mémoire dans le calcul avant de passer par une fonction d'activation non linéaire. Ici, nous utilisons les informations bidirectionnelles afin que les informations puissent être passées dans les deux sens (de gauche à droite et de droite à gauche). Cela nous aidera à recueillir des informations sur les battements cardiaques normaux à gauche et à droite du battement cardiaque central.

Cela a pris du temps à s'entraîner. Nous avons réduit l'ensemble de formation à 20 000 échantillons. Le programme s'est entraîné sur 5 EPOCH avec un batch size qui vaut 32.

```
Epoch 1/5
73096/73096 [=====] - 6s 80us/step - loss: 0.2518 - accuracy: 0.9065
Epoch 2/5
73096/73096 [=====] - 4s 54us/step - loss: 0.1670 - accuracy: 0.9467
Epoch 3/5
73096/73096 [=====] - 4s 57us/step - loss: 0.1483 - accuracy: 0.9537
Epoch 4/5
73096/73096 [=====] - 4s 55us/step - loss: 0.1335 - accuracy: 0.9596
Epoch 5/5
73096/73096 [=====] - 4s 53us/step - loss: 0.1262 - accuracy: 0.9612
```

FIGURE 4.14 – **Précision et perte tout au long de l'exécution du modèle LSTM.**



```
print('Train');
print_report(y_train, y_train_preds_cnn, thresh)
print('Valid');
print_report(y_valid, y_valid_preds_cnn, thresh);
```

```
Train
AUC:0.993
accuracy:0.966
recall:0.967
precision:0.923
specificity:0.966
prevalence:0.299
```

```
Valid
AUC:0.907
accuracy:0.821
recall:0.809
precision:0.723
specificity:0.827
prevalence:0.358
```

FIGURE 4.15 – évaluation du modèle.

## 4.9 Limites

- La non collecte de données supplémentaires comme suggéré avec la courbe d'apprentissage.
- La non exploration la littérature sur la prévalence de l'arythmie pour voir si cet ensemble de données est représentatif de la population générale.
- L'utilisation de LSTM est plus simple en utilisant le framework pytorch que tensorflow

## 4.10 Tensorflow et pytorch

Ils sont tête de la liste des frameworks les plus populaires en apprentissage profond. PyTorch a été publié en 2016 par le laboratoire de recherche sur l'IA de Facebook. Comme son nom l'indique, il est principalement destiné à être utilisé en Python, mais il possède également une interface C ++. Les passionnés de Python l'adorent pour son style de programmation impératif, qu'ils considèrent comme plus «pythonique» que celui d'autres frameworks plus déclaratifs. Compte tenu de sa nature pythonique, PyTorch s'intègre parfaitement dans l'écosystème d'apprentissage automatique Python[52].

TensorFlow, d'autre part, a des interfaces dans de nombreux langages de programmation. Mais l'API Keras de haut niveau pour TensorFlow en Python a fait ses preuves auprès des praticiens du deep learning que la dernière version de TensorFlow l'intègre par défaut.

## 4.11 Comparaison entre les 2 modèles

Les réseaux de neurones convolutifs et récurrents sont à la base de nombreuses applications d'intelligence artificielle.

Un CNN est constitué de plusieurs couches de perceptrons, et les filtres qui constituent un réseau « comprennent » de mieux en mieux l'image à chaque passage. Les CNN sont à privilégier pour l'interprétation de données visuelles, des données éparpillées ou des données qui ne viennent pas en séquence. Tandis que les RNN s'appuient sur les données passées pour prédire l'état futur.

Nous avons utilisé le modèle CNN qui fonctionne pour 1D CNN qui va prendre un filtre (noyau) de taille kernel-size commençant par le premier horodatage. L'opérateur de convolution prend le filtre et multiplie chaque élément par rapport aux premiers kernel-size, nous avons utilisé 100 EPOCHS comme nous avons utilisé un réseau neuronal récurrent LSTM avec 5 EPOCHS, cependant nous avons de meilleurs résultats avec le réseau neuronal convolutif avec plus de 99% de précision tandis que LSTM nous avons obtenu une précision de 98%, malgré plusieurs essais d'améliorations de l'algorithme en modifiant les hyper paramètres que ça soit le nombre d'EPOCHS ou le nombre de batch size.

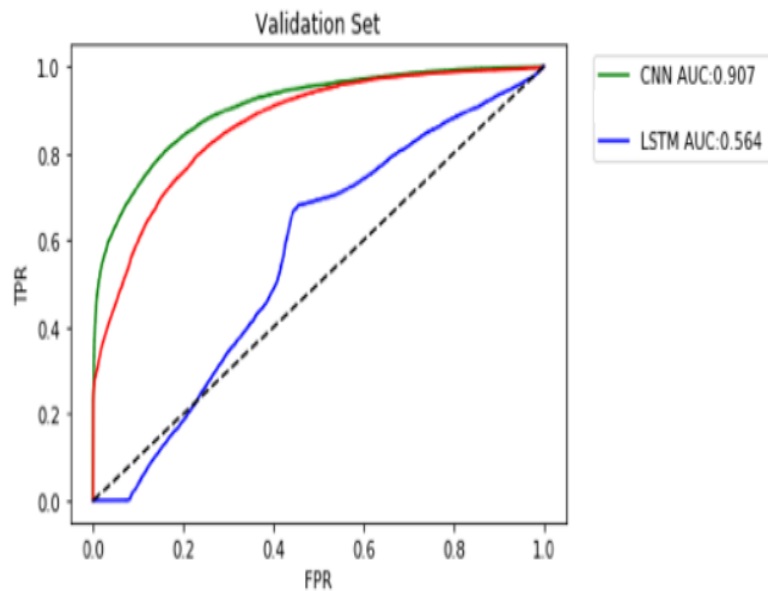


FIGURE 4.16 – Courbe ROC

Avec plus de temps, il serait bon d'essayer d'optimiser les hyperparamètres et d'utiliser Pytorch

avec LSTM.

CNN	LSTM
EPOCH : 100	EPOCH :5
Données spatiales (Images 2D et 1D).	Données temporelles, séquentielles.
Précision ACC : 99%	Précision ACC :96%
Loss :2%	Loss :12%

TABLE 4.2 – Comparaison entre les résultats des deux modèles.

Bien que les résultats montrent que CNN est meilleur que LSTM, cependant il faut savoir que LSTM peut être amélioré en modifiant quelque paramètre de son architecture ou en combinant les 2 réseaux de deep learning.

## 4.12 Conclusion

Dans ce chapitre, nous avons utilisé deux (02) modèles de classification basés sur les réseaux de neurones convolutifs et récurrents, pour la classification des battements cardiaques comme normaux et anormaux à partir des paramètres caractérisant chaque signal ECG.

Pour l'évaluation de ces modèles, nous avons effectué des tests sur la base de données MIT-BIH. Les 2 modèles ont donné des résultats différents, concluant que, pour la classification de l'ECG, un modèle neuronal à convolution avec plus de filtres par couche fonctionne mieux qu'un modèle récurrent en utilisant la technique LSTM.

Nous pouvons constater que l'utilisation des réseaux de neurones convolutifs avec apprentissage approfondi pour la détection automatique des arythmies ECG a permis d'avoir des résultats très satisfaisants et prouvent ainsi la capacité du réseau de classer correctement les battements cardiaques dans un court temps.

# Conclusion générale

Chaque personne est exposée aux nombreuses maladies. Certaines maladies nécessitent un diagnostic immédiat par le médecin, l'Electrocardiographie reste un examen indispensable en cardiologie depuis de longues années, L'objectif principal de notre projet est la détection d'anomalie de l'activité électrique du cœur (ECG) qui sont localisées dans ces différentes ondes et intervalles et qui reflète l'état de santé de l'appareil cardiovasculaire et l'une des principales causes c'est Le taux élevé de mortalité dans le monde dû aux problèmes liés au dysfonctionnement de l'appareil cardiaque qui a poussé les chercheurs à développer des techniques de classification automatique des maladies cardiovasculaires pour un bon diagnostic . Diverses techniques ont été utilisées pour classer les arythmies, nous sommes en mesure de mettre en évidence tous les aspects de l'ECG et de ses éléments les plus importants, nous avons également abordé un aperçu sur l'ECG et ses caractéristiques, ensuite une explication simplifiée de Deep learning qui est une structure similaire à celle du cerveau humain. Ce mémoire représente l'ensemble des tâches que nous avons effectué durant l'année scolaire pour aboutir à des objectifs fixés d'avance, lors de la réalisation de ce projet, On a eu l'occasion de mettre en pratique les cours qui nous ont été prodiguées au sein de l'université Targa Ouzemour ainsi que d'autre cours qu'on a suivi en tel que la programmation en langage Python. On porte un intérêt pour la data science et ce mémoire nous a permis d'apprendre beaucoup de choses que ce soit en théorie ou en pratique. Notre travail ouvre des perspectives scientifiques. Nous soulignons dans ce qui suit les perspectives qui nous semblent pertinentes pour l'évolution des systèmes développés dans ce projet.

- Tester le modèle **LSTM** avec le framework pytorch.
- Visualiser les résultats de nos modèles sur des cas Réels.
- Combiner les 2 architectures LSTM et CNN et essayer d'avoir de meilleurs résultats.

Le plaisir de réaliser un projet est que nous apprenons et découvrons pleins de connaissances que nous ignorons. Il y a également un grand plaisir de faire relire le mémoire par des amis qui sont dans des domaines non informatique et qu'ils trouvent le sujet intéressant. Ils trouvent que c'est important de comprendre ce qu'est l'intelligence artificielle puisque tout le monde en parle.

# Bibliographie

- [1] « Définition de l'électrocardiogramme », <https://www.passeportsante.net/>, 12 août 2015, <https://www.passeportsante.net/fr/Maux/examens-medicaux-operations/Fiche.aspx?doc=examen-electrocardiogramme>.
- [2] Yasar Sattar ; Lovely Chhabra., electrocadigram ,Last Update : December 3, 2019.
- [3] Lolita P. Pharm D, Ph, « Cardiologie : Généralités sur le coeur, Rythme cardiaque, Structure Anatomique », 2020, <https://www.sante-sur-le-net.com/maladies/cardiologie/generalites-coeur/>.
- [4] « Comment fonctionne le cœur | Fondation des maladies du cœur et de l'AVC | Fondation des maladies du cœur et de l'AVC », consulté le 17 octobre 2020, <https://www.coeuretavc.ca/maladies-du-coeur/qu-est-ce-que-les-maladies-du-coeur/comment-fonctionne-le-coeur>.
- [5] « Définition | Électrocardiogramme - ECG | Futura Santé », consulté le 18 juillet 2020, <https://www.futura-sciences.com/sante/definitions/medecine-electrocardiogramme-3353/>.
- [6] Laurence Piquard, « Comprendre l'ECG (électrocardiogramme), un défi osé? », Actusoins - infirmière, infirmier libéral actualité de la profession, 8 avril 2016, <https://www.actusoins.com/266783/comprendre-lecg-electrocardiogramme-un-defi-ose-2.html>.
- [7] « Intervalles et segments de l'électrocardiogramme », 1 octobre 2020, <https://fr.my-ekg.com/generalites-ecg/intervalles-segments-ecg.html>.
- [8] « 3. L'ELECTROCARDIOGRAMME NORMAL », consulté le 25 juillet 2020, <http://archive.cme.mcgill.ca/html/videos/hrt-org-ekg/bro981107R3.html>.
- [9] Ziad Obermeyer, M.D., M.Phil. and Ezekiel J. Emanuel, M.D., Ph.DHHS Author Manuscripts,19 oct 2016, Predicting the Future — Big Data, Machine Learning, and Clinical Medicine.
- [10] Rebecca R. Chamley et al., « ECG Interpretation Interpretation of the ECG in Young, Fit, Asymptomatic Individuals Undertaking High-Hazard Occupations Is the Topic of the Fourth Article in the Occupational Cardiology Series », European Heart Journal 40, no 32 (21 août 2019) : 2663-66, <https://doi.org/10.1093/eurheartj/ehz559>.
- [11] Abhinav Vishwa, Mohit K. Lal, Sharad Dixit, Dr. Pritish Vardwa, International Journal of Artificial Intelligence and Interactive.

- [12] e-monsite, « Définition de l'Intelligence Artificielle ». <http://tpe-intelligence-artificielle-2013.e-monsite.com/pages/definition-de-l-intelligence-artificielle.html> (consulté le aout.01, 2020).
- [13] Antoine Cornuéjols – Laurent Miclet (2014). Apprentissage Artificiel : concepts et algorithmes. Edition Eyrolles
- [14] Zakariyaa ISMAILI, « Apprentissage Supervisé Vs. Non Supervisé ». <https://le-datascientist.fr/apprentissage-supervise-vs-non-supervise>
- [15] MENJ - MESRI - MC, « Commission d'enrichissement de la langue française ». <https://www.education.gouv.fr/bo/19/Hebdo6/CTNR1832601K.htm> (consulté le aout.03, 2020).
- [16] R. Munos, « Apprentissage par renforcement, étude du cas continu », PhD Thesis, Doctorat Sciences cognitives option intelligence artificielle, Ecole des . . . , 1997.
- [17] Aurélien Géron, « Machine-Learning-with-Scikit-2E2019 ». 2019-04-22
- [18] L +Bastien, « Deep Learning ou apprentissage profond ». <https://www.lebigdata.fr/deep-learning-definition> (consulté le aout.04, 2020).
- [19] +Bastien L, « Deep Learning ou apprentissage profond : définition, concept », LeBigData.fr, juill. 16, 2018. <https://www.lebigdata.fr/deep-learning-definition> (consulté le aout.05, 2020).
- [20] O. Kharkovyna, « An Intro to Deep Learning for Face Recognition », Medium, juin 26, 2019. <https://towardsdatascience.com/an-intro-to-deep-learning-for-face-recognition-aa8dfbbc51fb> (consulté le aout. 09, 2020).
- [21] « Définition | Deep Learning - Apprentissage profond | Futura Tech ». <https://www.futurasciences.com/tech/definitions/intelligence-artificielle-deep-learning-17262/> (consulté le aout. 09, 2020).
- [22] Brett Grossfeld, « Deep learning vs machine learning ». <https://www.zendesk.fr/blog/machine-learning-and-deep-learning/> (consulté le aout. 10, 2020).
- [23] LABIAD ALI, « COMME EXIGENCE PARTIELLE DE LA MAÎTRISE EN MATHÉMATIQUES ET INFORMATIQUE APPLIQUÉES ». JUIN 2017
- [24] LUCAS, « Comment l'intelligence artificielle va-t-elle révolutionner la santé? ». <https://ia-data-analytics.fr/intelligence-artificielle/sante/> (consulté le aout. 10, 2020).
- [25] Benoît RIVIERE, « L'intelligence artificielle et les systèmes experts | Audit Systèmes d'Information ». <https://www.auditsi.eu/?page-id=2273> (consulté le aout. 11, 2020).
- [26] N. P. Rougier, "Perceptron simple Perceptron multi-couches.". Cours en ligne pour les Master 2 - Sciences cognitives, université de Bordeaux.
- [27] johan-ellau, « Perceptron. Teori perceptron dan simple. . . | by johan-ellau | Medium ». Jun 20 · 7 <https://medium.com/@ellaujohan/perceptron-de822b401b58> (consulté le aout. 11, 2020).
- [28] AICHA Majda, « le Perceptron multicouche ». <https://msirtadgrp2.files.wordpress.com/2015/02/percept-multicouche.pdf> (consulté le aout. 11, 2020).

- [29] Manish Nayak, « An Intuitive Introduction Of Restricted Boltzmann Machine (RBM) ». Apr 17, 2019 . <https://medium.com/datadriveninvestor/an-intuitive-introduction-of-restricted-boltzmann-machine-rbm-14f4382a0dbb> (consulté le aout. 11, 2020).
- [30] Pascal Monasse, et al., « Découvrez les différentes couches d'un CNN ». <https://openclassrooms.com/fr/courses/4470531-classez-et-segmentez-des-donnees-visuelles/5083336-decouvrez-les-differentes-couches-dun-cnn> (consulté le aout. 16, 2020).
- [31] Prabhu, « Understanding of Convolutional Neural Network (CNN) — Deep Learning | by Prabhu | Medium ». Mar 4, 2018 <https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148> (consulté le aout. 16, 2020).
- [32] « Travaux pratiques sur le machine learning ». <https://developers.google.com/machine-learning/practica/image-classification/convolutional-neural-networks> (consulté le aout. 17, 2020).
- [33] Van den Oord, A., Dieleman, S., Schrauwen, B. (2013). Deep content-based music recommendation. In *Advances in neural information processing systems* (pp. 2643-2651).
- [34] Julien Krywyk, Pierre-Alain Jachiet, « Classification d'images : les réseaux de neurones convolutifs en toute simplicité | OCTO Talks! » 25/10/2016 <https://blog.octo.com/classification-dimages-les-reseaux-de-neurones-convolutifs-en-toute-simplicite/> (consulté le aout. 17, 2020).
- [35] « Boughaba-Boukhris.pdf ». <https://dspace.univ-ouargla.dz/jspui/bitstream/123456789/17195/1/Boughaba-Boukhris.pdf>.
- [36] Dauphin, Y. N., Pascanu, R., Gulcehre, C., Cho, K., Ganguli, S., Bengio, Y. (2014). Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *Advances in neural information processing systems* (pp. 2933-2941).
- [37] l'équipe SuperDataScience, « Recurrent Neural Networks (RNN) - The Vanishing Gradient Problem - Blogs SuperDataScience - Big Data | Analytics Careers | Mentors | Success ». Thursday Aug 23, 2018 <https://www.superdatascience.com/blogs/recurrent-neural-networks-rnn-the-vanishing-gradient-problem> (consulté le aout. 18, 2020).
- [38] Anirudh Sharma, « (1) What is the vanishing gradient problem? - Quora ». August 29, 2018 <https://www.quora.com/What-is-the-vanishing-gradient-problem>
- [39] Youcef Messaoud, « LSTM, Intelligence artificielle sur des données chronologiques | by Youcef Messaoud | Smile Innovation | Medium » Nov22,2018 <https://medium.com/smileinnovation/lstm-intelligence-artificielle-9d302c723eda> (consulté le aout. 15, 2020).
- [40] Singh Jaskaran, « A Simpler and Effective Implementation of LSTM Demonstrated with a Sales Prediction Model ». Apr 14·7 <https://sparks.mediaagility.com/a-simpler-and-effective-implementation-of-lstm-demonstrated-with-a-sales-prediction-model-6c430ec64860>
- [41] Sihem Nouas, « Analyse sémantique des publications dans les réseaux sociaux par apprentissage profond ». 17/07/2019

- 
- [42] C. Laurent, G. Pereyra, P. Brakel, et coll., «Réseaux de neurones récurrents normalisés par lots», dans IEEE ICASSP, Shanghai, mars. 2016.
- [43] S. Lawrence et C. L. Giles, « Overfitting and neural networks : conjugate gradient and back-propagation », in Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing : New Challenges and Perspectives for the New Millennium, 2000, vol. 1, p. 114–119.
- [44] A. H. Ribeiro et al., « Automatic diagnosis of the 12-lead ECG using a deep neural network », Nature Communications, vol. 11, no 1, Art. no 1, avr. 2020, doi : 10.1038/s41467-020-15432-4.
- [45] C. Barès, « Python–Informatique », 2020.
- [46] « wfdb - documentation wfdb 3.1.1 ». <https://wfdb.readthedocs.io/en/latest/> (consulté le oct. 04, 2020).
- [47] « Plot With Pandas : Python Data Visualization for Beginners – Real Python ». <https://realpython.com/pandas-plot-python/> (consulté le oct. 04, 2020).
- [48] R. Python, « Python Tutorials – Real Python ». <https://realpython.com/> (consulté le oct. 04, 2020).
- [49] « Warning — Programmation avec le langage Python ». <http://www.xavierdupre.fr/app/teachpyx/helpsphinx/c-exception/warning.html> (consulté le oct. 11, 2020).
- [50] S. S. Haykin, Neural networks and learning machines/Simon Haykin. New York : Prentice Hall, 2009.
- [51] G. B. Moody et R. G. Mark, « The impact of the MIT-BIH arrhythmia database », IEEE Engineering in Medicine and Biology Magazine, vol. 20, no 3, p. 45–50, 2001.
- [52] I. Vasilev, D. Slater, G. Spacagna, P. Roelants, et V. Zocca, Python Deep Learning : Exploring deep learning techniques and neural network architectures with Pytorch, Keras, and TensorFlow. Packt Publishing Ltd, 2019.



## RÉSUMÉ

L'électrocardiogramme (ECG) est un test qui mesure le rendement électrique du cœur par enregistrement des variations des ondes électriques de l'activité de polarisation et de dépolarisation ventriculaire et auriculaires du cœur. Il est l'un des tests les plus importants dans le domaine de la médecine car il est utilisé pour détecter tous les problèmes du cœur.

Ce travail s'inscrit dans cette problématique et présente un algorithme de détection des anomalies de rythme cardiaque par l'application des algorithmes de deep Learning(DL) tel que CNN et LSTM. Les résultats ont été validés par des signaux ECG de différents patients de la base de données « MIT-BIH Arrhythmia ».

**Mots clés :** CNN,deep Learning(DL),Electrocardiogram(ECG), LSTM, MIT-BIH.

## ABSTRACT

The electrocardiogram (ECG) is a test that measures the electrical output of the heart by recording changes in the electrical waves of the heart's ventricular and atrial polarization and depolarization activity. It is one of the most important tests in the field of medicine because it is used to detect all problems of the heart.

This work is part of this problem and presents an algorithm for the detection of cardiac rhythm abnormalities through the application of deep Learning(DL) algorithms such as CNN and LSTM. The results were validated by ECG signals from different patients in the « MIT-BIH Arrhythmia » database.

**Key words :** CNN,deep Learning(DL),Electrocardiogram(ECG), LSTM, MIT-BIH.