REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

**UNIVERSITE YAHIA FARES DE MEDEA**

**F**aculté de **T**echnologie

**D**épartement du **G**énie **E**lectrique

# Mémoire de fin d'études de Master

Filière : Télécommunications

Spécialité : Système de Télécommunications

# Etude et implémentation d'un système connecté intelligent pour la surveillance de l'état de santé des patients souffrant d'arythmie cardiaque.

Présentés par :

**Berkani Mohamed Rafik Aymene**

**Achir Zahra**

Proposés et Dirigés par :

**Billel Bengherbia**

**Promotion 2022**

# Remerciements

Avant de commencer, louange et gratitude à **Allah le Tout Puissant**, de nous avoir donné la volonté, la santé, la patience et le courage d'atteindre notre objectif.

Il nous est agréable de remercier vivement et sincèrement notre encadreur **Mr. BENGHERBIA Billel** qui malgré ses lourdes taches, a bien voulu diriger et corriger minutieusement notre travail. Nous avons été particulièrement sensibles à sa présence constante durant toute l'année. Il a droit à toute notre reconnaissance, pour la grande patience dont il a fait preuve, pour les conseils, l'aide technique et matérielle, et l'encouragements, qu'il nous a prodigués.

Nos vifs remerciements aux :

**Mr. REBIAI Mohamed** pour ses enseignements dont nous avons tiré profit durant tout le cursus de spécialité et pour avoir bien voulu accepter d'honnorer de sa présence le jury et d'en assurer la présidence

**Mr. HANTABLI Hamza** pour avoir bien voulu examiner ce travail et accepter de participer au jury.

**Mr. CHOUCHANE Ammar**. Qu'il trouve ici l'expression de notre sincère gratitude et notre profond respect.

Nous ne saurions oublier d'adresser nos sincères remerciements et reconnaissances à toutes les personnes qui ont contribué de près ou de loin à la réalisation de ce travail.

Nos remerciements vont évidemment à tous les membres de nos deux familles en particulier nos très chers parents, à nos frères et sœurs et à nos amis(es) pour leur soutien moral et pour l'ambiance agréable qu'ils ont crée autour de nous.

# Dédicaces

Les jours passent et repassent

Et la science progresse.

Les chercheurs meurent et ne laissent

Que le fruit de leur labeur.

Selon ces derniers le vrai savoir : est de savoir que l'on sait ce que l'on sait et que l'on ne sait pas ce l'on ne sait pas.

C'est à ceux qui croient à l'expression de ces paroles que nous dédions notre travail

A mon très cher **père**.

A ma très chère **mère**.

Qui sans eux, je ne serais pas là où j'en suis maintenant, pour tous leurs sacrifices, leurs prières, leurs amours et leurs soutiens tout au long de mon parcours académique.

A mes très chères sœurs.

A mes très chers oncles et tantes des deux familles **Berkani** et **Zerrouki**.

A mes très chers **grands-parents** maternels**.**

A tous ceux qui ont participé de près ou de loin à l'élaboration de ce travail.

A tous mes amis sans exception avec qui j'ai passé d'agréables moments.

Sans oublier mes professeurs : primaire, moyen, secondaire et d'enseignement supérieur.

Ce travail est particulièrement dédié à mon ami et frère **Babaali Abdellah**

Cet ouvrage est également dédié à la mémoire de mes grands-parents paternels Berkani Mohamed décédé le 27/04/2013 et Lakmouchi FZ décédé le 20/01/2005.

**Berkani Mohamed Rafik Aymene.**

# Dédicace

Toutes les lettres ne sauraient trouver les mots qu'il faut...

Tous les mots ne sauraient exprimer la gratitude, l'amour, Le respect, la reconnaissance...

Je dédie cette Thèse...

A mes chers parents pour leur amour inconditionnel, leur encouragement, et leur soutien.

A ma mère, EL Mohri merci pour tes prières, tes encouragements, c'est par les quels que j'ai pu surmonter tous les obstacles, tu es unique.

A mon père Achir qui m'a beaucoup aidé, m'a tout fourni pour que je puisse accomplir ce travail, merci d'avoir être toujours à côté de moi.

Merci à mes deux sœurs et frère Selma, Nour et Abd Nour qui m'ont soutenu.

A la personne avec qui j'ai partagé les trajets et toute mes années qui m'a toujours soutenu mon oncle EL MOHRI Madani

A mes chers grands-parents que ce modeste travail, soit l'expression des vœux que vous n'avez cessé de formuler dans vos prières.

Que Dieu vous préserve santé et longue vie.

A Tout les membres de ma famille mes tantes et oncles,

Je ne pourrais jamais exprimer le respect que j'ai pour vous. Vos prières, vos encouragements et votre soutien m'ont toujours été un grand secours.

Puisse Dieu, le tout puissant vous préserver du mal, vous combler de santé et de bonheur.

Amis, chacun par son nom.

A tous ceux qui m'ont aidé, encouragé, et qui ont contribué de près ou de loin dans la réalisation de ce travail.

Merci d'être toujours là pour moi.

**Achir Zahra**

## ملخص

نقدم في هذا العمل معلومات عامة عن القلب بالإضافة لمختلف أمراض الشرايين مع أسباب حدوثها. لهذا نقترح نظاما ذكيا للكشف عنها و تصنيفها بسرعة وبصفة تلقائية، بالإعتماد على شبكة عصبية إلتفافية (CNN) التي بدورها تنتمي للتعلم العميق. أولا، نحاول إنشاء شبكة ذكية لجمع و تسجيل إشارات تخطيط القلب (ECG) من أجل إستعمالها في الإختبار النهائي، ثم نقوم بإدراج نموذج الـCNN على بطاقة Raspberry Pi 3B+ لإختبار الإشارات وعرض نتائج التنبؤ الخاصة بها على منصة Node-Red. علما أن هذا النظام الذكي يمكن ان يشخص المرض بسرعة فائقة ودقة عالية وهذا تفاديا للأخطاء الطبية في الحالات الإستعجالية خلال الكوارث الطبيعية.



**الكلمات المفتاحية:** إشارات تخطيط القلب ECG، القلب، الأمراض القلبية، IOT، CNN، Raspberry Pi، Node-Red.



## Résumé

Dans ce travail, nous présentons des informations et des généralités sur le cœur, les différentes maladies cardiovasculaires ainsi que leurs causes. Pour cela nous proposons un système intelligent pour la détection et classification des maladies cardiovasculaires systématiquement à l'aide d'un réseau neurones de convolution (CNN 1D) qui introduit à l'apprentissage en profondeur. D'abord nous essayons de créer un nœud d'acquisition de récolte des signaux ECG pour le test final, ensuite nous effectuons une implémentation du modèle CNN sur la carte Raspberry Pi 3B+ pour tester et afficher les différents résultats de prédictions sur la plateforme Node-Red. Le système intelligent proposé a prouvé et a validé son efficacité et sa fiabilité. Ce système intelligent a pour but de minimiser les erreurs médicales en cas d'urgences lors des catastrophes.



**Mots clé :** Signaux ECG, cœur, maladie cardiovasculaire, CNN, IOT, Raspberry Pi, Node-Red.



## Abstract

In this project, we present general information about human heart, without forgetting the various cardiovascular diseases as well as their causes. For this causes we propose an intelligent system, who can be detect and classifies different cardiovascular diseases systematically, using a convolutional neural network (CNN 1D) that introduces to deep learning. First we try to create an ECG signal harvest acquisition node for the final test, then we perform an implementation of the CNN model on the Raspberry Pi 3B+ (board) to test and display the different prediction results on the Node-Red platform. The proposed intelligent system has proven and validated its

effectiveness and reliability. This intelligent system aims to minimize medical errors in case of emergencies during disasters.

**Key words:** ECG signals, heart, cardiovascular disease, CNN, IOT, Raspberry Pi, Node-Red.

# Sommaire

Liste des Figures ..... X
Liste des Tableaux ..... XIII
Liste des Abréviation ..... XIV
Introduction Générale
Chapitre I : Généralité sur le cœur
I.1-Introduction ..... 2
I.2-Le Cœur ..... 2
I.2.1-Anatomie du Cœur ..... 2
I.2.2-Cycle Cardiaque ..... 4
I.3-Activité électrique du cœur ..... 5
I.4-L'origine de l'activité électrique ..... 6
I.5-Phonocardiogramme ..... 7
I.6-Eléctrocardiogramme ..... 7
I.7-Les Phases d'ECG ..... 8
I.8-Les Ondes et les intervalles d'ECG ..... 9
I.9-Description d'un ECG normal ..... 10
I.10-Aquisition et traitement d'un signal ECG ..... 11
I.11-Dérivation ..... 12
I.11.1-Trois dérivations bipolaires (ou dérivations standard) ..... 12
I.11.2-Trois dérivations unipolaires aVR, aVL, aVF ..... 13
I.11.3-Les Six dérivations précordiales ..... 14
I.12-Les Maladies Cardiovasculaires ..... 14
I.13-Les Causes des maladies cardiaques ..... 15
I.13.1-Cause des maladies Cardiovasculaires ..... 15
I.13.2-Les Causes de l'Arythmie Cardiaque ..... 15
I.14-Les Problèmes Cardiaques ..... 15
I.14.1-Rythme sinusal normal ..... 15
I.14.2-Fibrillation Atrial(Auriculaire) (AFIB) ..... 16
I.14.3-Contraction ventriculaire prématurée (extrasystole ventriculaire) (PVC) ..... 17
I.14.4-Left Bundle Branch Block Beat (Battement du Bloc de Branche Gauche) (LBBB) ..... 18
I.15-Conclusion ..... 19
Chapitre II : Intelligence Artificielle et Internet des Objets
II.1-Introduction ..... 20
II.2-Intelligence Artificielle ..... 20
II.3-Machine Learning ..... 22
II.3.1-Types d'apprentissage automatique ..... 23
II.3.1.1-Apprentissage Supervisé ..... 23

II.3.1.2-Apprentissage Non Supervisé ..... 23
II.3.1.3-Apprentissage par Renforcement ..... 23
II.4-Deep Learning ..... 24
II.4.1-Apprentissage supervisé ..... 25
II.4.2-Apprentissage Non Supervisé ..... 26
II.5-Les Réseaux Neurones ..... 26
II.5.1-Neurone Biologique ..... 26
II.5.2-Neurone Artificiel ..... 27
II.5.3-Architecture des réseaux neurones artificiels ..... 28
II.5.4-Le Perceptron ..... 29
II.5.5-Les Fonctions d'activation ..... 29
II.5.5.1-Fonction seuil (Echelon unitaire) ..... 29
II.5.5.2-La Fonction de Sigmoïde ..... 30
II.5.5.3-La Fonction Softmax ..... 30
II.5.5.4-La Fonction Tangente Hyperbolique « Tanh » ..... 30
II.5.5.5-La Fonction ReLu (Rectified Linear Unit) ..... 31
II.5.5.6-La Fonction Leaky ReLu ..... 31
II.5.5.7-La Fonction ELU (Unités Linéaires Exponentielles) ..... 31
II.5.5.8-La Fonction Argmax ..... 32
II.5.6-Types des réseaux neurones ..... 32
II.5.6.1-Réseau neurones de convolution CNN ..... 33
II.5.6.2-Architecture d'un réseau CNN ..... 33
II.6-Internet des Objets ..... 39
II.6.1-Architecture d'IoT ..... 39
II.6.2-Les Composantes d'IOT ..... 40
II.6.2.1-Les Capteurs ..... 40
II.6.2.2-Réseaux des capteurs ..... 41
II.6.2.3-Energie ..... 41
II.6.2.4-Actionneurs ..... 41
II.6.2.5-Connectivité ..... 41
II.6.3-Domaine d'application ..... 42
II.6.3.1-Villes intelligentes ..... 42
II.6.3.2-Domotique ..... 43
II.6.3.3-Agriculture ..... 44
II.6.3.4-La Santé (eHealth) ..... 44
II.7-Conclusion ..... 45

Chapitre III : Implémentation et test expérimental

III.1-Introduction ..... 46
III.2-Base de donnée (1000 Fragments ECG) ..... 46

III.2.1-Description de Base de données ..... 46
III.3-Les logiciels utilisés ..... 47
III.3.1-Matlab ..... 47
III.3.2-Python ..... 48
III.4-Les bibliothèques utilisées ..... 48
III.4.1-Tensorflow ..... 48
III.4.2-Keras ..... 48
III.4.3-Sklearn ..... 49
III.4.4-Matplotlib ..... 49
III.6-Prétraitement des Signaux ECG ..... 50
III.6.1-Segmentation des fichiers ..... 51
III.6.2-Décomposition ..... 52
III.6.3-Remplissage des signaux ..... 52
III.6.4-Enregistrement ..... 53
III.6.5-Normalisation ..... 54
III.6.6-Distribution de base de données ..... 55
III.7-Modèle CNN ..... 56
III.7.1-Architecture utilisée ..... 56
III.7.2-Méthode utilisée pour l'optimisation ..... 61
III.7.2.1-KFold Cross Validation ..... 61
III.7.2.2-Callbacks Model Check Point ..... 62
III.7.2.3-Compiler ..... 62
III.7.3-Les Résultats obtenus ..... 62
III.7.3.1-Graphe de précision et des pertes ..... 62
III.7.3.2-Rapport de classification ..... 63
III.7.3.3-Matrice de Confusion ..... 64
III.8-Comparaison ..... 65
III.9-Test expérimentale ..... 66
III.9.1-Matériels utilisé ..... 66
III.9.1.1-Carte Arduino Uno ..... 66
III.9.1.2-Capteur ECG AD8232 ..... 67
III.9.1.3-Les Electrodes ..... 67
III.9.2-Logiciels utilisés ..... 68
III.9.2.1-Arduino IDE ..... 68
III.9.3-Réalisation du nœud d'acquisition ..... 68
III.9.4-Résultats des tests ..... 70
III.9.4.1-Chargement du modèle ..... 71
III.9.4.2-Préparation des signaux ..... 71
III.9.4.3-Prédiction et classification des signaux ..... 71

III.10-Implémentation ..... 73
III.10.1-Raspberry Pi ..... 73
III.10.2-Raspberry Pi 3B+ ..... 74
III.10.3-Tensorflow Lite ..... 74
III.10.4-Préparation du modèle ..... 74
III.10.5-Résultat d'implémentation ..... 75
III.10.6-Node-Red ..... 76
III.10.6.1-Editeur Node-Red ..... 77
III.10.6.2-Les nœuds ..... 77
III.12-Conclusion ..... 80
Conclusion générale ..... 81
Bibliographies ..... 82

Annexe

Annexe A ..... i
Annexe B ..... ii
Annexe C ..... iii
Annexe D ..... iv

# Liste des Figures

**Figure I-1 :** La Cage thoracique----------02

**Figure I-2 :** Le cœur Humain----------03

**Figure I-3 :** Cycle Cardiaque----------04

**Figure I-4 :** Activité électrique du Cœur----------06

**Figure I-5 :** Enregistrement des potentiels d'action d'une cellule de Purkinje et cellule du nœud sinusal----------07

**Figure I-6 :** Phonocardiogramme----------07

**Figure I-7 :** Les Différentes Phases d'ECG----------09

**Figure I-8 :** Les Différentes Ondes P, Q, R, S, T----------10

**Figure I-9 :** Acquisition du signal ECG----------12

**Figure I-10 :** Signal ECG Normal----------12

**Figure I-11 :** Montage d'Einthoven pour l'enregistrement bipolaires des membres----------13

**Figure I-12 :** Dérivations Unipolaires----------13

**Figure I-13 :** Les Dérivations Thoraciques----------14

**Figure I-14 :** Rythme sinusal normal----------16

**Figure I-15 :** Fibrillation Atrial(Auriculaire)----------17

**Figure I-16 :** Contraction ventriculaire prématurée----------18

**Figure I-17:** Left bundle branch block beat----------19

**Figure II-1 :** Types de la Machine Learning----------24

**Figure II-2 :** Domaine d'intelligence artificiel----------25

**Figure II-3 :** Un ensemble de donnée étiqueté (classé) pour l'apprentissage supervisé et puis la classification d'une donnée inconnue----------25

**Figure II-4 :** Apprentissage Non Supervisé, l'entrainement et classement des données selon leurs caractéristique----------26

**Figure II-5 :** Neurone Biologique----------27

**Figure II-6 :** Structure d'un Réseau Neurone Artificiel ANN----------28

**Figure II-7 :** Architecture d'un Réseau Neurone Artificiel----------29

**Figure II-8 :** Principe de calcul d'un neurone artificiel----------29

**Figure II-9 :** Les Différentes Fonctions d'Activation----------32

**Figure II-10 :** Architecture de Base pour un réseau CNN----------33

**Figure II-11 :** Opération de Convolution----------34

**Figure II-12 :** Opération de Padding----------35

**Figure II-13 :** Sortie de Fonction d'activation----------35

**Figure II-14 :** Opération de Max Pooling----------36

**Figure II-15 :** Opération de Average Pooling----------37

**Figure II-16 :** Opération d'abandon (Dropout)----------38

**Figure II-17 :** Couche entièrement connectée (Fully Connected layer)----------38

**Figure II-18 :** Architecture d'IOT----------39

**Figure II-19 :** Les différentes Capteurs----------40

**Figure II-20 :** Infographie d'une ville intelligente----------43

**Figure II-21 :** Maison intelligente----------44

**Figure II-22 :** Agriculture intelligente----------44

**Figure II-23 :** L'application d'IOT dans le domaine médical----------45

**Figure III-1 :** Organigramme de Traitement des données----------50

**Figure III-2 :** Segmentation des fichiers----------51

**Figure III-3 :** Décomposition des fichiers----------52

**Figure III-4 :** Remplissage des signaux----------53

**Figure III-5 :** Enregistrement sous forme csv----------53

**Figure III-6 :** Normalisation du signal----------54

**Figure III-7 :** Distribution la Base de donnée----------55

**Figure III-8 :** Scatter plot du Base de donnée----------55

**Figure III-9 :** Diagramme schématique de procédure d'entrainement des réseaux CNN------56

**Figure III-10 :** L'architecture utilisée----------60

**Figure III-11:** K-Fold Cross Validation----------61

**Figure III-12 :** Graphe de précision et des pertes----------63

**Figure III-13 :** Rapport de Classification----------63

**Figure III-14 :** Matrice de confusion----------64

**Figure III-15 :** Signaux NSR et PVC----------65

**Figure III-16 :** Les Graphes d'Accuracy et Loss des différentes méthodes, A : Avec K-Fold, B : Sans K-Fold, C : Avec Normalisation et K-Fold----------66

**Figure III-17 :** Arduino Uno----------67

**Figure III-18 :** Capteur ECG AD8232----------67

**Figure III-19 :** Les électrodes utilisé----------68

**Figure III-20 :** Réalisation du nœud d'acquisition----------69

**Figure III-21 :** Placement des électrodes----------70

**Figure III-22 :** Signal récolté----------70

**Figure III-23 :** Organigramme de test expérimental----------71

**Figure III-24 :** Classement des signaux----------71

**Figure III-25 :** Matrice de confusion----------72

**Figure III-26 :** Organigramme des étapes d'implémentation----------73

**Figure III-27 :** Raspberry Pi 3B+----------74

**Figure III-28 :** Préparation des données----------75

**Figure III-29 :** Résultats de la prédiction d'un seul signal----------75

**Figure III-30 :** Résultats de la prédiction pour 16 personnes au même temps----------76

**Figure III-31 :** Editeur Node-Red----------77

**Figure III-32 :** Schéma des nœuds pour visualiser et sauvegarder les signaux----------78

**Figure III-33 :** Schéma des nœuds pour l'affichage des résultats----------79

**Figure III-34 :** Les résultats de prédictions----------80

# Liste des Tableaux

**Tableau III-1 :** Description des données collectées ----------47

**Tableau III-2 :** Les meilleurs Architectures élaborées----------57

**Tableau III-3 :** Meilleur Résultat du réseau----------65

**Tableau III-4 :** Les meilleurs Résultats de cette architecture----------65

**Tableau III-5 :** L'interconnexion entre ECG AD8232 et Arduino Uno----------69

**Tableau III-6 :** Placements des électrodes----------69

**Tableau III-7 :** Les résultats de test expérimental----------72

**Tableau III-8 :** Résultats des implémentations----------76

# Liste des Abréviation

**ADC :** Analog to Digital Conversion.

**AFIB :** Fibrillation Atrial (auriculaire).

**AI :** Artificial Intelligence.

**ANN :** Artificial Neural Network.

**API :** Application Programming Interface.

**AV :** Auriculo Ventriculaire.

**AVC :** Accident Vasculaire Cérébral.

**aVR, aVL, aVF :** Dérivations unipolaires.

**BN :** BatchNormalization.

**CNN :** Convolutional Neural Network.

**CSV :** comma-separated values.

**DL:** Deep Learning.

**DI, DII, DIII :** Dérivations Standard (d'EINTHOVEN) de l'ECG.

**ECG :** Electrocardiogramme.

**ELU :** Exponential Linear Units.

**FNN :** Feed Forward Neural Network.

**GPU :** Graphics processing unit.

**GUI :** Graphical User Interface.

**GND :** Ground.

**GPIO :** General-Purpose Input/Output.

**LBBB :** Left Bundle Branch Block.

**LReLu :** Leaky Rectified Linear Unit.

**IBM :** International Business Machines.

**IOT :** Internet of things.

**Matlab :** MaTrix Laboratory.

**MCV :** Maladies CardioVasculaires.

**ML :** Machine Learning.

**MNN :** Modular Neural Network.

**NSR :** Normal Sinus Rhythm.

**OMS :** Organisation Mondiale de la Santé.

**OS :** Operating System.

**PC :** Personal Computer.

**PVC :** Premature Ventricular Contraction.

**POE :** Power over Ethernet.

**RBF :** Radial Basis Function Neural Network.

**ReLu :** Rectified Linear Unit.

**RNN :** Recurrent Neural Network.

**SOM :** Self Organization Map Neural Network.

**SND :** Shut Down.

**Sklearn :** Scikit-learn.

**SVM :** Machines à Vecteur de Support.

**Tanh :** Tangente Hyperbolique.

**TPU :** Thermoplastic Polyurethane.

**USB :** Universal Serial Bus.

**UIT :** Union Internationale des Télécommunications.

**Wi-Fi :** Wireless Fidelity.

# Introduction Générale

# Introduction général

Selon l'Organisation mondiale de la santé (OMS), les maladies cardiovasculaires (MCV) sont la cause principale des décès dans le monde dont plus de 17,7 millions personnes ont été victimes des MCV et ce en 2017 [1].

Le diagnostique de ces maladies fatales semble une tâche vitale dans les services de cardiologie au niveau des hôpitaux. Le signal électrocardiogramme (ECG) reste l'un des outils prédominants pour le diagnostique et l'analyse des arythmies cardiaques.

En réalité, le diagnostique (l'analyse) d'ECG est un outil non invasif effectué par le médecin en vue d'explorer le fonctionnement du cœur par l'emploi des électrodes externes mises en contact avec la peau. Il s'agit d'un signal qui reflète les variations de l'activité électrique du cœur en fonction du temps.

Du fait des erreurs médicales commises par certains médecins généralistes non spécialisés lors du diagnostic d'ECG, on nous a proposé notre projet de fin d'études qui consiste à créer un système intelligent de détection des maladies cardiovasculaires par analyse intelligent des signaux ECG.

Le premier chapitre explicite d'une part l'anatomie du cœur humain, les cycles cardiaques, l'activité électrique, les différentes ondes et intervalles d'un signal ECG, l'acquisition et le traitement d'un signal ECG avec les différentes dérivations. D'autre part, les maladies cardiovasculaires et leurs causes.

Le second chapitre consiste à exposer le domaine d'intelligence artificielle et ses sous-domaines tels que : apprentissage automatique (ML), l'apprentissage en profondeur (DL) avec une description des différents types de DL. Pour cela, on a adopté le modèle des réseaux neurones artificiels (ANN) et à convolutions (CNN), tout en détaillant les différentes couches de l'architecture des réseaux CNN 1D et leurs différents paramètres. Sans oublier de décrire l'internet des objets et ses domaines d'application.

Le dernier chapitre, est consacré à la partie prétraitement des ECG et la création de la base de données, et l'élaboration d'un modèle CNN capable de détecter et classer les différentes maladies cardiovasculaires d'une façon automatique avec une grande précision. Cependant, ce modèle doit être testé par des signaux récoltés à l'aide d'un capteur d'ECG AD8232, afin de prouver et valider son efficacité et sa fiabilité. Ce modèle intelligent sera enfin implémenté et testé sur la carte Raspberry Pi 3B+.

# Chapitre I : Généralité sur le cœur

## I.1-Introduction

Ce chapitre consiste à découvrir : l'anatomie du cœur humain, son fonctionnement, les cycles cardiaques, l'activité électrique du cœur, les différentes ondes et intervalles d'un signal ECG, l'acquisition et le traitement d'un signal ECG avec les différentes dérivations. D'autre part, on veut avoir une idée générale sur les maladies cardiovasculaires, les causes de ces maladies. Ensuite on va discuter et donner une brève description sur quelques problèmes et maladies cardiaques et ses causes. A la fin de ce chapitre, une idée sur l'électrocardiographie comme méthode de visionnement et diagnostique des différentes maladies cardiaques est aussi donnée.

## I.2-Le Cœur

Le cœur est un organe musculaire de $[250-350]g$, situé au milieu de la cage thoracique dans un espace appelé médiastin antérieur, il se trouve entre les deux poumons, en arrière du sternum, en avant de la colonne vertébrale et au-dessus du diaphragme. Il a un diamètre de $[14-16]cm$ et un volume de $[50-60]cm^3$. La cage thoracique et ses détails sur la figure I-1.

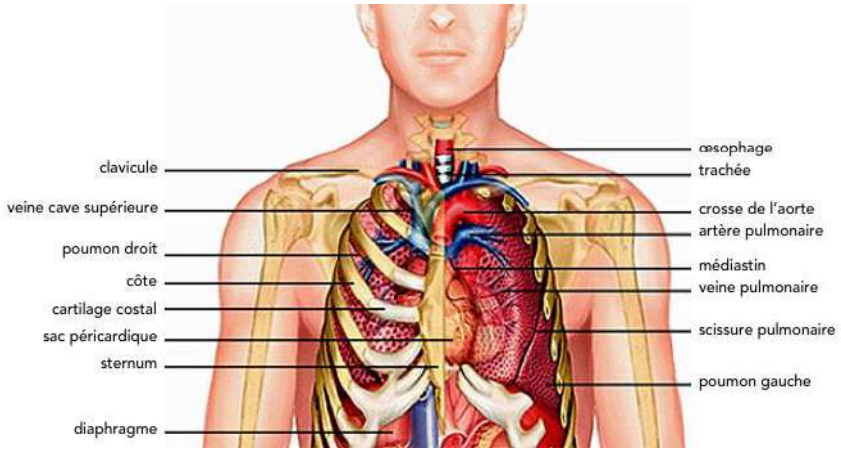



**Figure I-1 :** La Cage thoracique

### I.2.1-Anatomie du Cœur

Le cœur est composé de quatre chambres appelées cavités cardiaques :

- Deux oreillettes en haut.
- Deux ventricules en bas.

Ils sont entourés par une paroi musculaire parcourue de terminaisons nerveuses. Les parois des ventricules sont plus épaisses que celles des oreillettes qui sont capables de pousser le sang avec une très grande force.

Le septum est une paroi musculaire épaisse qui sépare le cœur gauche (l'oreillette et le ventricule gauches) du cœur droit (l'oreillette et du ventricule droits) afin d'empêcher le passage du sang entre les deux parties du cœur.

Il existe des valves entre les oreillettes et les ventricules qui assurent le passage unidirectionnel du sang de l'oreillette vers le ventricule.

La figure I-2 représente le cœur humain et ses constituants.

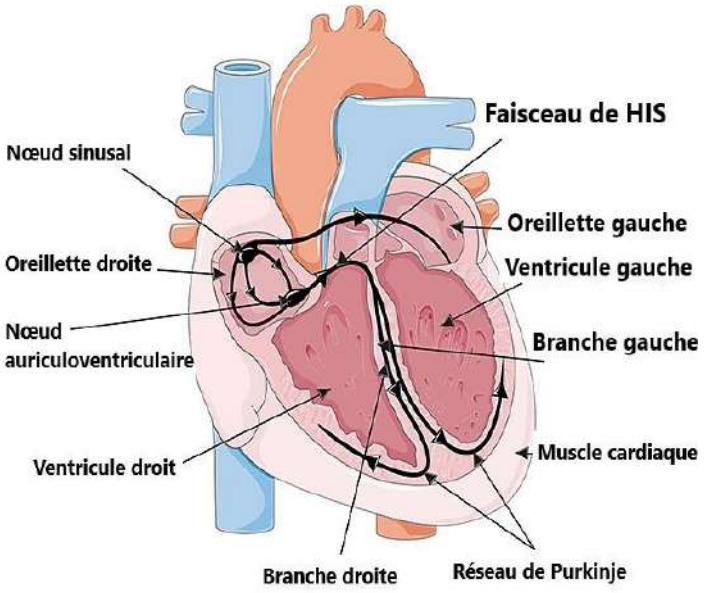



**Figure I-2 :** Le Cœur Humain

La paroi du cœur est constituée par un muscle qui ne se fatigue pas. Elle est constituée par trois différentes couches :

- L'épicarde (epicardium) : se compose par un épithélium pavimenteux simple (mésothélium) doublé d'une fine couche sous-mésothéliale. Il est séparé du myocarde par un tissu conjonctif.
- Le myocarde (myocardium) : c'est un muscle cardiaque, entre les fibres myocardiques s'étendent des interstices allongés contenant un tissu conjonctif dense, des capillaires artériels, veineux et lymphatique.
- L'endocarde : se trouve à l'intérieur. Un endothélium mince, doublé d'une fine couche sous-endothéliale. Une couche conjonctif-élastique, une couche sous-endocardique de tissu conjonctif lâche, ou l'on trouve des éléments de tissu nodal [2].

### I.2.2-Cycle Cardiaque

Le cœur possède une fréquence de battements de $[60 - 70]$ battements par seconds et un débit de $[4.5 - 5]$ litres de sang par minutes. Chaque battement du cœur entraine une série d'évènements connus sous le nom de révolution cardiaque. Il est constitué de trois phases principals :

- **Systole auriculaire :** les oreillettes se contractent et passent le sang vers les ventricules. Une fois le sang expulsé des oreillettes, les valves, entre les oreillettes et les ventricules : la valve tricuspide et la valve mitrale, se ferment. Cette fermeture évite un retour du sang vers les oreillettes. Ainsi cette fermeture des valves produit le son familier des battements du cœur.
- **Systole ventriculaire :** implique la contraction des ventricules, qui pousse le sang vers le système circulatoire. Une fois le sang expulsé la valve pulmonaire et la valve aortique se ferment. Par conséquent le sang ne retourne pas vers les ventricules. La fermeture de ces valves produit un deuxième son cardiaque plus aigu que le premier. Au cours de cette systole, les oreillettes se remplissent de sang.
- **Diastole :** C'est la relaxation de toutes les parties du cœur, permettant le remplissage (passif) des ventricules et l'arrivée de nouveau sang [3].

Ces phases sont représentées sur la figure I-3

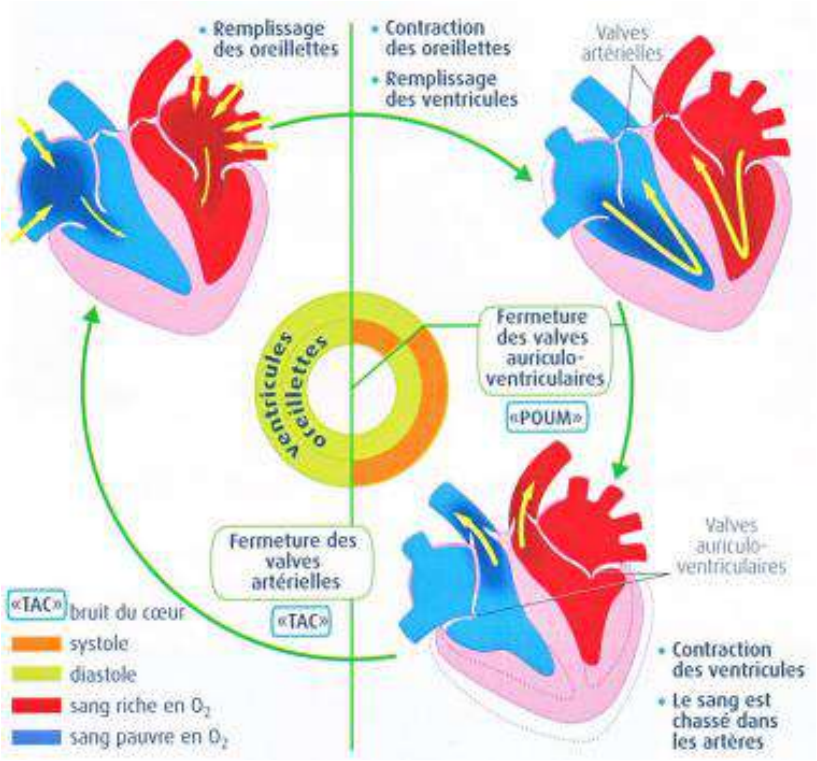



**Figure I-3 :** Cycle Cardiaque

## I.3-Activité électrique du cœur

La contraction cardiaque (voir figure I-4) est due au raccourcissement des cardiomyocytes lors de l'excitation. Pour que cette excitation se produise correctement, le cœur doit développer :

- **Un centre de contrôle ou « pace maker » :** qui émet automatiquement des ondes d'excitation à un intervalle régulier.
- **Un système spécialisé :** le tissu nodal (ganglionnaire), qui est un tissu conducteur chargé de propager des ondes d'excitation à grande vitesse dans certaines zones cardiaques.
- **Système régulateur extracardiaque :** c'est un système nerveux sympathique qui augmente le rythme cardiaque et le système nerveux parasympathique qui ralentit le rythme cardiaque (pneumogastrique).
- **Le « pace maker » normal :** c'est le nœud sinusal (sino-auriculaire) ou nœud de Keith et Flack situé dans la paroi de l'oreillette droite au voisinage de l'orifice de la veine cave supérieure.

  La fréquence du nœud sinusal est très élevée, elle est de 100 battements par minute, mais la fréquence cardiaque réelle est plus faible en raison de l'action constante du nerf vague (pneumogastrique), qui effectue un « frein vagal » afin de régler le rythme sinusal de [70-75] battements par minute.

  La dépolarisation commence au sein du nœud sinusal (sino-auriculaire) et se propage à l'oreillette droite, puis à l'oreillette gauche. Dans les oreillettes, il n'y a pas de voies de conduction spécifique.
- **Le tissu nodal :** comprend :
  - **Le nœud d'Aschoff-Tawara :** est situé en arrière et au-dessous du septum interauriculaire, où elle collecte les ondes d'activation à son pôle supérieur et les conduit vers le faisceau **His.**
  - **Le faisceau His :** c'est un pont musculaire reliant les oreillettes et les ventricules, il court sur le côté droit du septum et se divise en deux branches droite et gauche :

    1-La branche droite : est longue.

    2-La branche gauche : se divise en deux branches : antérieure et postérieure.
  - **Réseau Purkinje :** les deux branches du faisceau **His** se résolvent en un réseau couvrant l'endocarde des ventricules [4].

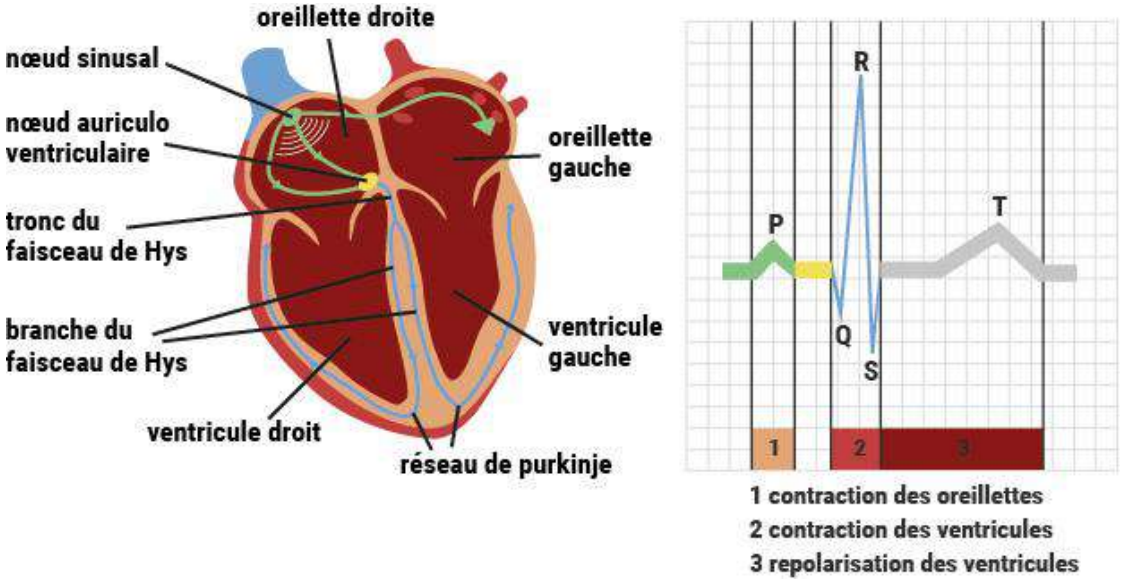



**Figure I-4 : Activité électrique du Cœur**

## I.4-L'origine de l'activité électrique

L'intérieur de la cellule cardiaque au repos est électriquement négatif par rapport à l'extérieur, la sortie active d'ions sodium maintenant une différence de potentiel de membrane d'environ (moins) $-90\ mV$. La diminution de cette différence de potentiel jusqu' à un seuil de $60mV$ déclenche un potentiel d'action auto-entretenu.

Dans la cellule de Purkinje, ceci se produit en réponse à une stimulation électrique qui réduit la différence de potentiel jusqu'au seuil par augmentation de la perméabilité de membrane aux ions sodium. Le potentiel d'action passe par cinq phases (comme le montre la figure I-5) :

- **La Phase 0 :** Dépolarisation rapide causée par une entrée rapide d'ions sodium.
- **La Phase 1 :** Repolarisation précoce causée par la sortie d'ions sodium.
- **La Phase 2 :** Plateau pendant lequel la dépolarisation est temporairement interrompue par l'entrée lente d'ions calcium.
- **La Phase 3 :** Repolarisation rapide due à la poursuite de la sortie des cations.
- **La Phase 4 :** Diastole au cours de laquelle la différence de potentiel transmembranaire se maintient à $-90mV$ jusqu'à ce que la cellule soit à nouveau stimulée (potentiel de repos) [5].

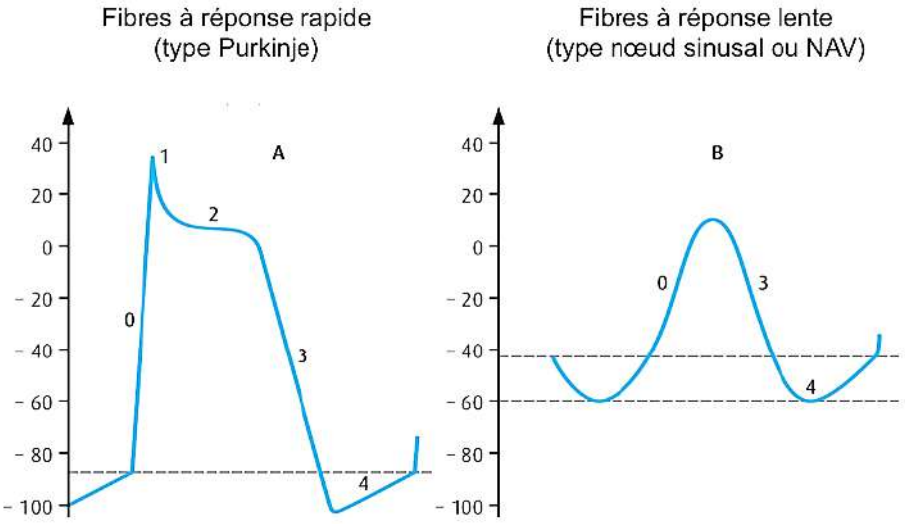



**Figure I-5 : Enregistrement des potentiels d'action d'une cellule de Purkinje et cellule du nœud sinusal**

## I.5-Phonocardiogramme

Le phonocardiogramme est un enregistrement des sons cardiaques ou le signal sonore est transformé en impulsions électriques de façon à pouvoir être enregistré sur un tracé. Cet examen permet de préciser ce que le cardiologue détecte à l'oreille avec son stéthoscope. Il n'est plus effectué car il a été remplacé avec avantage par l'écho-doppler [6].



**Figure I-6 : Phonocardiogramme**

## I.6-Eléctrocardiogramme

L'électrocardiogramme (ECG) est un tracé qu'on peut obtenir par l'enregistrement et la transcription des courants électriques qui parcourent le cœur pendant chaque contraction cardiaque. Il donne des informations sur l'activité du nœud sinusal à travers les oreillettes aux ventricules.

L'électrocardiogramme permet de détecter si un infarctus (une crise cardiaque) a eu lieu ultérieurement ou si le cœur est mal irrigué. Elle est collectée par des électrodes à la surface de la peau. L'ECG est le seul outil qui peut déceler un trouble du rythme cardiaque [7].

Les électrocardiographes enregistrent l'activité électrique du cœur, ils interceptent également l'activité d'autres muscles comme les muscles squelettiques. Les électrocardiographes sont pour filtrer cette activité électrique dans la mesure du possible. Durant l'enregistrement de l'ECG, le patient doit être en état de relâchement musculaire pour obtenir une trace non parasite.

Par convention on attribue aux ondes principales de l'ECG les lettres P, Q, R, S, T et U. Chaque onde représente la dépolarisation (décharge électrique) ou la repolarisation (recharge électrique) d'une certaine région du cœur.

Les changements de voltage détectent par l'électrocardiographe sont minimes (de l'ordre du millivolt). La taille de chaque onde correspond à l'amplitude du voltage génère par l'évènement qui lui a donné naissance [8].

## I.7-Les Phases d'ECG

L'électrocardiogramme est composé d'une phase d'activité appelée la systole alternée par une phase de repos appelée diastole. Le rythme de ces deux phases est d'environ 70 Fois par minute (70 $bpm$).

Il est à préciser que la phase diastole est d'une durée supérieure à celle de la systole pour un fonctionnement cardiaque normal. Par contre, en cas de pathologie cardiaque comme la tachycardie, la durée de la diastole est très fortement raccourcie, alors que celle de la systole varie peu.

- **L'onde P :** La naissance de l'onde P commence avec le début de l'impulsion produite au nœud Sinusal. La fin de l'onde P correspond au remplissage total des oreillettes.
- **L'onde Q :** L'onde Q correspond au début du passage du sang des oreillettes vers les ventricules.
- **Le complexe QRS :** Le complexe QRS se traduit par le remplissage total des ventricules avec un pic R qui correspond à l'ouverture totale des valves pour laisser passer le sang.
- **L'onde T :** L'onde T correspond à la contraction du myocarde pour impulser le sang stocké dans les ventricules vers le corps [9].

La figure I-7 représente les ondes cardiaque d'un ECG.

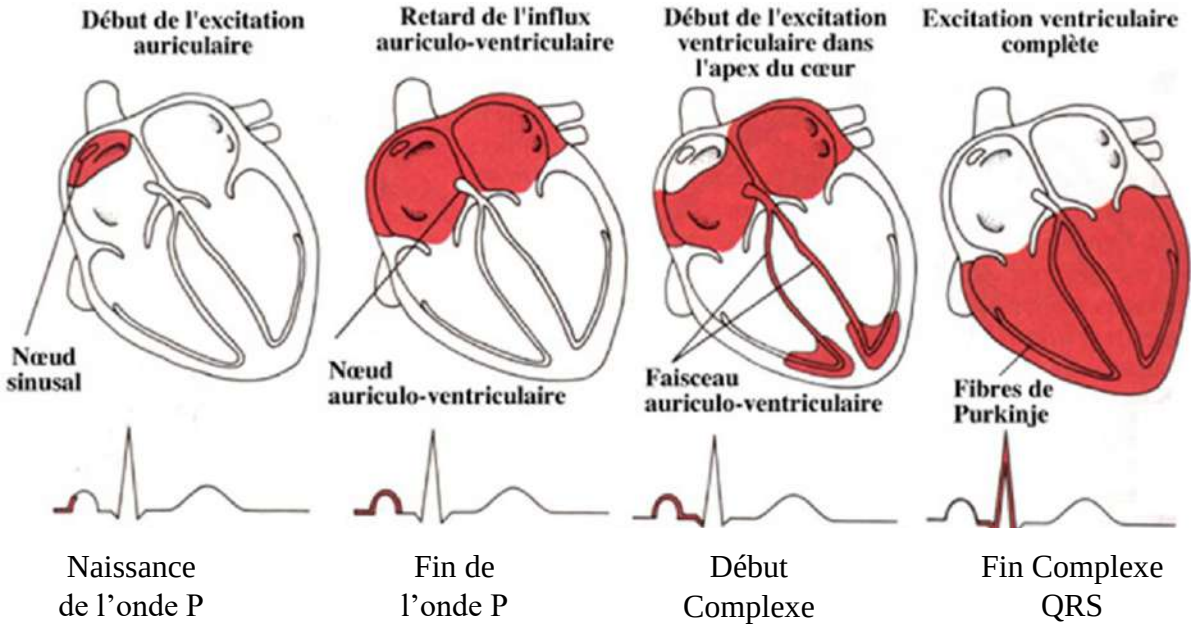



**Figure I-7 :** Les Différentes Phases d'ECG

## I.8-Les Ondes et les intervalles d'ECG

Le tracé relevé, nommé électrocardiogramme (ECG), présente un aspect bien défini, ses déformations sont le signe d'anomalies dans le fonctionnement cardiaque cependant toute interprétation de l'électrocardiogramme se réfère aux caractéristiques de forme et de largeur d'onde du signal ECG normal.

- **Onde P :** Elle correspond à la dépolarisation des oreillettes depuis le nœud sinusal vers le nœud atriau-ventriculaire. Elle peut être (de signes) positive ou négative avec une durée de l'ordre de $90ms$. C'est l'onde qui précède le complexe QRS, généralement il est difficile de l'observer sur l'ECG surtout dans des conditions bruitées.
- **Complexe QRS :** Il correspond à la dépolarisation ventriculaire précédant l'effet mécanique de contraction. Il a une plus grande amplitude de signal ECG, il est composé de trois ondes accolées qui suivent l'onde P : une onde appelée Q en premier d'amplitude négative, une onde appelée R d'amplitude positive dans un ECG normal et finalement l'onde S qui est (avec un amplitude) négative. La durée normale de complexe QRS est comprise entre $85ms$ et $95ms$.
- **Onde T :** Elle correspond à la repolarisation des ventricules, elle peut être (peuvent êtres de signes) positive, négative ou biphasique. C'est l'onde qui succède le complexe QRS après retour à la ligne isoélectrique (sauf pathologie particulière), son amplitude est plus faible que celle du complexe QRS.

- **Intervalle PR :** L'intervalle PR qui a une durée (de $[0.14 - 0.2]s$) entre $0.14s$ à $0.2s$ mesurée entre le début de l'onde P et le début de complexe QRS, c'est-à-dire la durée entre la dépolarisation des oreillettes et du nœud AV. Ce temps est la durée nécessaire pour qu'une impulsion électrique passe du nœud sinusal aux ventricules. Il est utilisé pour la détection d'arythmie et la variabilité de fréquence cardiaque.
- **Segment ST :** Ou encore durée de stimulation complète des ventricules. Elle correspond à l'intervalle durant lequel les ventricules restent dans un état de dépolarisation actif. C'est le temps entre la fin de l'onde S et le début de l'onde T.
- **Intervalle QT :** Cet intervalle correspond au temps de systole ventriculaire, qui va du début de l'excitation des ventricules jusqu'à la fin de leur relaxation. C'est le temps entre le début du complexe QRS et la fin de l'onde T [10].(comme le montre la figure I-8 ).

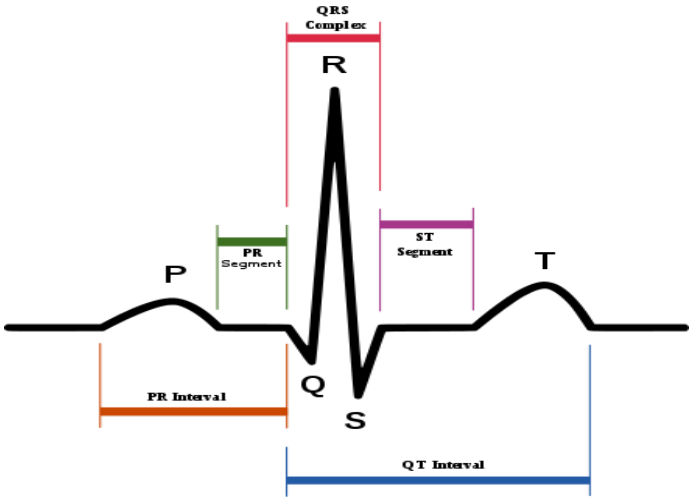



**Figure I-8 :** Les Différentes Ondes P, Q, R, S, T

## I.9-Description d'un ECG normal

- **Rythme cardiaque :** L'ensemble de l'activation cardiaque est sous la dépendance du nœud de Keith et Flack. Le rythme auriculaire est régulier, les intervalles P-P ne variant pas de plus de $0{,}15s$ au repos.

  La fréquence varie d'un sujet à l'autre, mais elle est considérée comme normale chez l'adulte, pour des valeurs comprises entre 60 et 100 battements par minute. Les ondes P ont un axe et une morphologie normale, ainsi que les complexes QRS qui sont normaux et séparés par un intervalle P-R supérieur à $0{,}12s$.
- **Durée :** La durée de l'onde P est inférieure à $0{,}12s$ avec des valeurs moyennes augmentant légèrement avec l'âge.

- **Amplitude :** L'amplitude de l'onde P est inférieure à $0{,}2mV$. Elle est maximale en dérivations $D_2$ et $V_1$.
- **Repolarisation :** La repolarisation auriculaire est en principe non visible, masquée par le complexe QRS, sauf en cas de bloc auriculoventriculaire. La durée de la repolarisation auriculaire varie de $0{,}15s$ à $0{,}45s$ et son amplitude est faible $0{,}08mV$. En cas de dilatation auriculaire, la repolarisation atriale peut modifier le segment ST.
- **Intervalle PR :** L'intervalle PR correspond au temps de conduction intra-auriculaire, auriculoventriculaire nodal et à la conduction dans le tissu His-Purkinje.
  - Chez l'adulte et l'adolescent, la durée de l'intervalle PR varie entre $[0{,}11 - 0{,}21]s$.
  - Chez les sujets jeunes et vagotoniques à rythme cardiaque inférieur à $60(bpm)$ par minute, il est possible d'observer des intervalles PR allant jusqu'à 0,24s.
  - Chez les enfants de moins de 10 ans, l'intervalle PR est plus court (entre $[0{,}10 - 0{,}17]s$).

  Dans des conditions basales, la fréquence cardiaque modifie de façon minime et inconstante la durée de PR.
- **Durée de QRS :** La durée du complexe QRS se mesure dans les dérivations périphériques. Les valeurs normales vont de $[0{,}06 - 0{,}10]s$. La durée de QRS est plus courte chez les enfants ($0{,}09s$ avant $15ans$), mais reste inchangée à partir de l'adolescence [11].

## I.10-Aquisition et traitement d'un signal ECG

La propagation de l'activité électrique cardiaque provoque, à la surface du corps, l'apparition de différences de potentiel qui peuvent être enregistrer aux différents points de référence c'est un électrocardiogramme ECG. La figure I-10 présente un exemple classique des potentiels relevés par des électrodes placées sur les poignets du patient, la chaine d'acquisition et de traitement est en figure I-9, il sert au diagnostic médical [12].

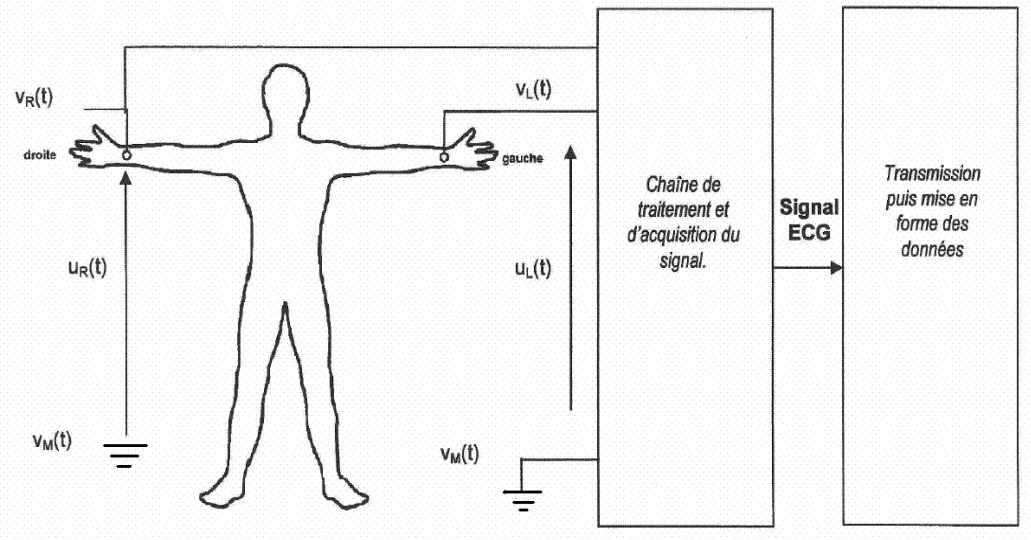



**Figure I-9 :** Acquisition du signal ECG

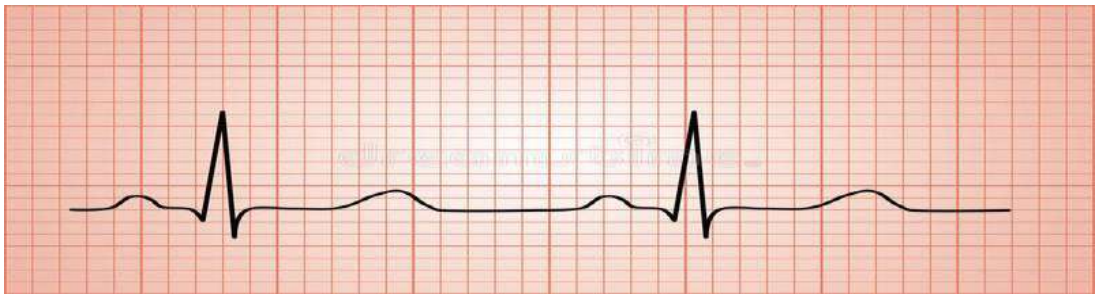


**Figure I-10 :** Signal ECG Normal

## I.11-Dérivation

Une dérivation est un circuit électrique déterminé par une paire d'électrodes (une différence de potentiel électrique). Pour obtenir une topographie complète du cœur, il faut enregistrer 12 dérivations (12 capteurs enregistreurs) de l'ECG et qui se répartissent en deux catégories [13].

### I.11.1-Trois dérivations bipolaires (ou dérivations standard)

Elles ont été déterminées par Einthoven (Einthoven, 1906) et elles sont dites bipolaires car le potentiel est mesuré entre deux électrodes. Elles sont obtenues à partir des potentiels du $DI, DII, DIII$ obtenues par permutation des électrodes placées sur le bras droit, le bras gauche et la jambe gauche de la manière suivante :

- $DI = VL - VR.$
- $DII = VF - VR.$
- $DIII = VF - VL.$
    - VR correspond au potentiel au bras droit.

- VL correspond au potentiel au bras gauche.
- VF correspond au potentiel dans la jambe gauche.

La jambe droite est reliée à la masse. Les vecteurs obtenus forment alors un triangle équilatéral appelé triangle d'Einthoven.

### I.11.2-Trois dérivations unipolaires aVR, aVL, aVF

Les dérivations unipolaires des membres permettent d'étudier l'activité électrique du cœur sur le plan frontal. Elles ont été déterminées par Wilson (Wilson et al, 1934). Ces dérivations permettent de mesurer la tension entre un point de référence et le bras droit, le bras gauche et la jambe gauche respectivement. Le point de référence est réalisé par la moyenne des signaux qui apparaissent sur les deux autres membres qui ne sont pas en observation. A cet effet, on utilise des résistances de valeur élevée, supérieure à $5M\Omega$. Les figure I-11 et figure I-12 montre les dérivations unipolaires :

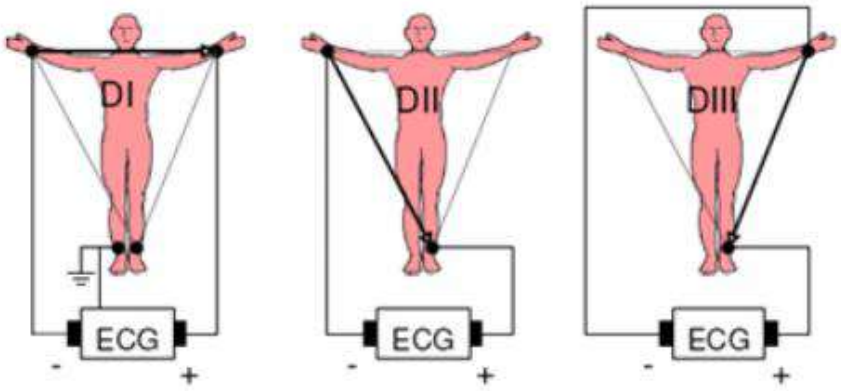



**Figure I-11 :** Montage d'Einthoven pour l'enregistrement bipolaires des membres

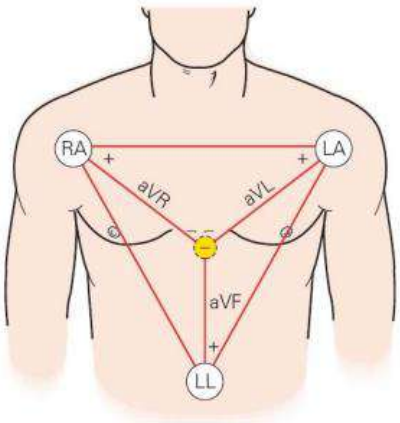



**Figure I-12 :** Dérivations Unipolaires

### I.11.3-Les Six dérivations précordiales

Ce sont des dérivations unipolaires mises au point par Wilson (Wilson et al, 1944). Elles sont posées sur le thorax et sont désignées par la lettre V suivie du numéro de leur emplacement. Le potentiel de l'électrode exploratrice est pris par rapport à la moyenne des potentiels VL, VR et VF. Six points, définis par Wilson, permettent d'obtenir les dérivations $V_1$ à $V_6$ (voir figure I-13).

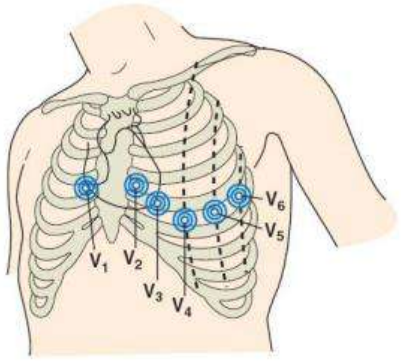



**Figure I-13 :** Les Dérivations Thoraciques

## I.12-Les Maladies Cardiovasculaires

Une maladie cardiaque décrit une série d'affections qui touchent le cœur. Parmi ces maladies de catégorie cardiaques, on cite celles des :

- Vaisseaux sanguins.
- Problèmes de rythme cardiaque (arythmie).
- Malformations cardiaques congénitales.

D'après l'organisation mondiale de la santé (OMS), les maladies cardiaques sont la cause la plus courante de mortalité en Algérie.

Le suivi et le traitement des personnes touchées par une pathologie cardiaque coûtent très cher au trésor public, ce qui représente un enjeu médical et financier en même temps. Dans ce contexte, les chercheurs scientifiques ont relevé le défi en proposant des systèmes intelligents permettant la détection des pathologies cardiaques offrant ainsi une aide au diagnostique médical tout en améliorant le quotidien des patients. Ces systèmes sont (inventés) développés et améliorés en exploitant les techniques les plus modernes en électronique et en informatique et ce dans le domaine de traitement du signal cardiaque [10].

## I.13-Les Causes des maladies cardiaques

Les pathologies cardiaques les plus fréquentes en Algérie sont : les maladies cardiovasculaires et les arythmies cardiaques.

### I.13.1-Cause des maladies Cardiovasculaires

L'athérosclérose est la cause la plus fréquente des maladies cardiovasculaires, définie comme étant une accumulation de plaques de graisses dans les artères qui épaissit et raidit les parois des artères, ce qui peut inhiber le flux sanguin dans les artères vers les organes et les tissus. Elle peut être causée également par des problèmes pouvant être corrigés tels qu'une mauvaise alimentation, l'obésité, le tabagisme et la sédentarité.

### I.13.2-Les Causes de l'Arythmie Cardiaque

Parmi les causes les plus courantes des rythmes cardiaques anormaux (arythmies), nous pouvons citer :

- Malformations cardiaques congénitales.
- Maladie des artères coronaires.
- Hypertension artérielle.
- Maladie cardiaque valvulaire.
- Diabète.
- Tabagisme.
- Consommation excessive d'alcool ou de caféine.
- Drogues.
- Stress.
- Automédication, les compléments alimentaires et les remèdes à base de plantes [10].

## I.14-Les Problèmes Cardiaques

### I.14.1-Rythme sinusal normal

Un rythme sinusal correspond au rythme cardiaque normal, c'est-à-dire piloté par le nœud sinusal avec conservation de la séquence « contraction des oreillettes » « contraction des ventricules ». L'électrocardiogramme dans ce cas ne révèle aucun trouble. Il doit présenter cet enchainement : l'onde P de dépolarisation des oreillettes (systole), l'espace PG qui est le temps de conduction auriculo-ventriculaire, l'ensemble QRS qui est une dépolarisation/repolarisation ventriculaire (qui dure de $[0{,}06 - 0{,}1]s$) et l'onde T qui est une repolarisation des ventricules (diastole ventriculaire). Son rythme est compris entre $[60 - 80]bpm$ avec un intervalle régulier

entre des battements normaux (voir figure I-14). Le cœur s'accélère normalement lors d'une activité physique, dans des circonstances physiologiques qui exigent un surcroît de demandes métaboliques et sous l'effet des émotions ou d'excitants tels que le café et le tabac [10].

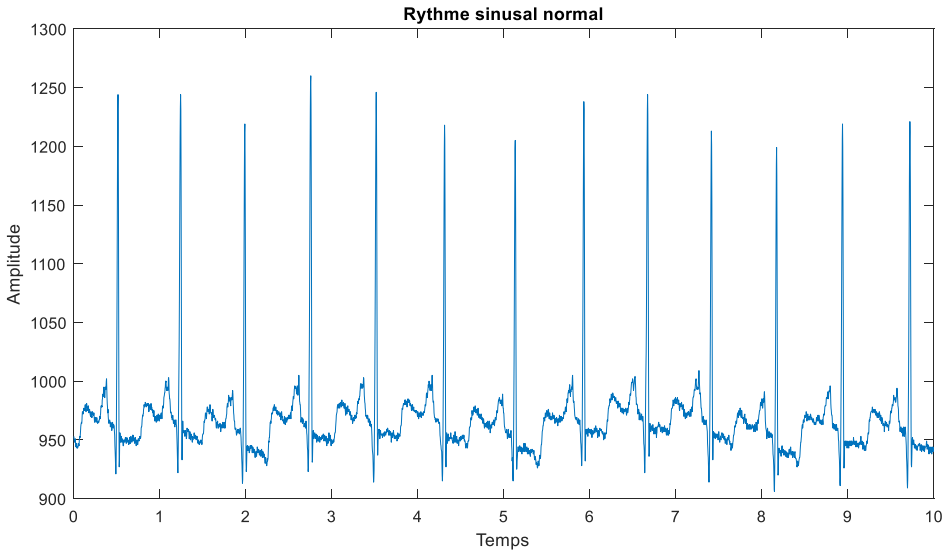



**Figure I-14 :** Rythme sinusal normal

### I.14.2-Fibrillation Atrial(Auriculaire) (AFIB)

La fibrillation atriale (auriculaire) survient lorsque les cavités supérieures du cœur, aussi appelées oreillettes ou atriums, se contractent de façon désynchronisée et frémissent (on dit aussi qu'elles présentent des fibrillations). Cela signifie qu'elles battent très rapidement et à un rythme irrégulier.

Le sang n'est pas pompé efficacement dans le reste de l'organisme, ce qui peut provoquer une sensation de faiblesse ou de la fatigue ou encore des sensations cardiaques inconfortables comme des palpitations.

Par ailleurs, le sang n'étant pas pompé efficacement dans la cavité, il y séjourne plus longtemps et des caillots peuvent ainsi s'y former. Ces caillots peuvent être ensuite dispersés dans la circulation ; si un tel caillot voyage jusqu'à une artère du cerveau, il peut en résulter un accident vasculaire cérébral(AVC) [14].

Le signal ECG de cette maladie montré sur la figure I-15.

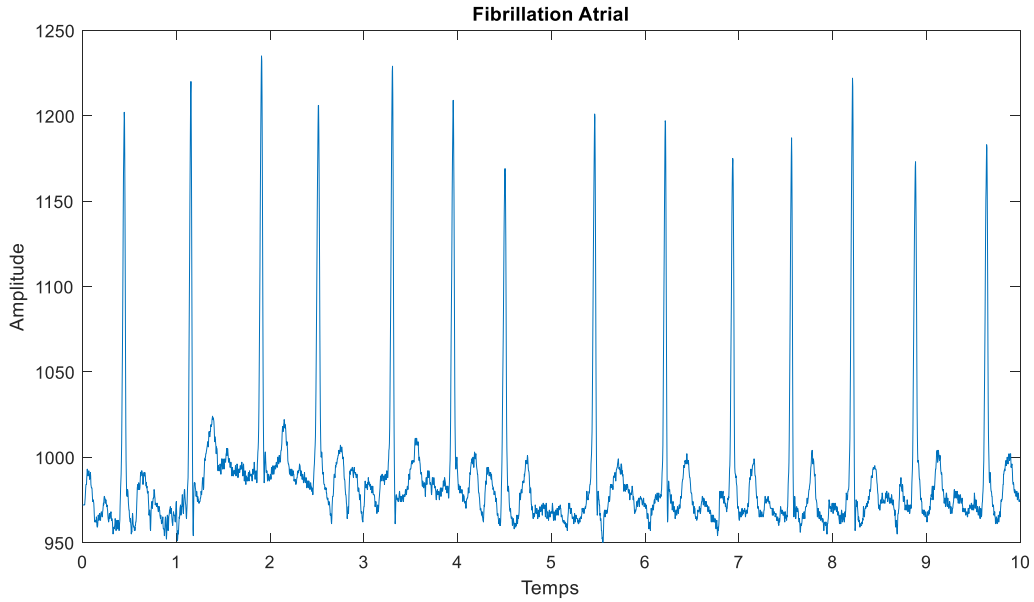



**Figure I-15 :** Fibrillation Atrial(auriculaire)

Les causes de la AFIB ne sont pas toujours identifiées :

- Peut-être provoquée par une anomalie cardiaque congénitale ou par une lésion causée à la structure du cœur à la suite d'une crise cardiaque ou en raison d'un trouble valvulaire par exemple.
- La fibrillation atriale peut également apparaître chez les patients qui ne présentent pas d'antécédents particuliers.

  Quel que soit la cause de la fibrillation auriculaire, le risque de développer ce type d'arythmie augmente rapidement avec l'âge [14].

### I.14.3-Contraction ventriculaire prématurée (extrasystole ventriculaire) (PVC)

L'extrasystole ventriculaire (voir figure I-16) est due à la présence d'une zone irritative dans le ventricule qui ne se soumet pas à l'harmonie de l'activation cardiaque. L'émotion, un déséquilibre de la teneur en sels minéraux, un état fiévreux ou une infection peuvent démasquer une propension à cette arythmie. Dans ces cas, celle-ci sera considérée comme bénigne, tandis que l'apparition d'extrasystoles accompagnant une maladie cardiaque peut dévoiler un problème qui nécessite un traitement. Si les extrasystoles se multiplient et que la stimulation électrique suit le mouvement, la fréquence cardiaque augmentera, on parlera alors de tachycardie [15].

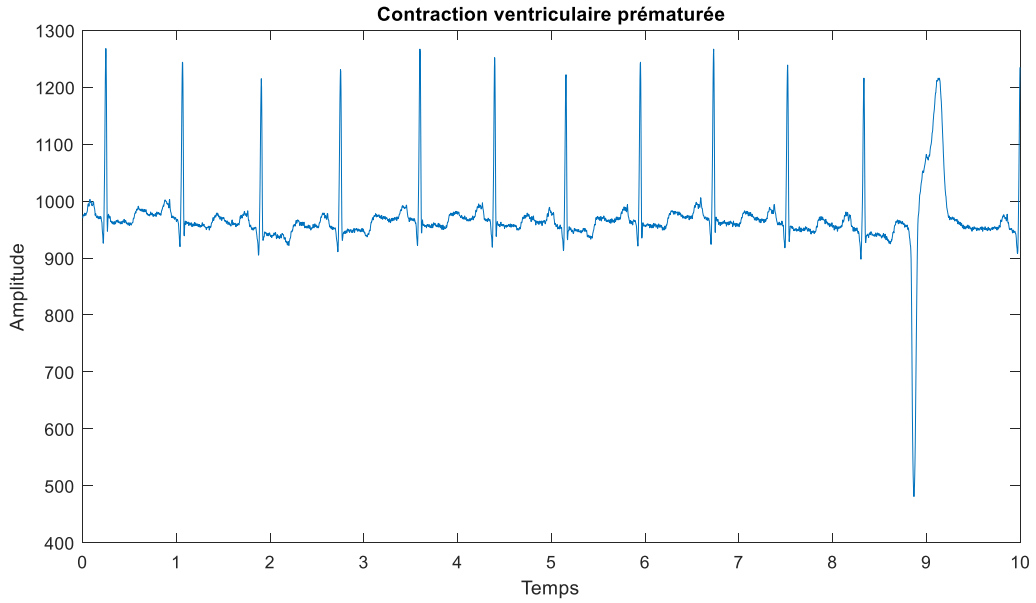



**Figure I-16 :** Contraction ventriculaire prématurée

Les contractions ventriculaires prématurées peuvent être causées par :

- Certains médicaments, y compris les décongestionnants et les antihistaminiques.
- Abus d'alcool ou de drogue.
- Stimulants tels que la caféine ou le tabac.
- Augmentation des niveaux d'adrénaline dans le corps due à l'exercice ou à l'anxiété.
- Blessure au muscle cardiaque due à une maladie [16].

### I.14.4-Left Bundle Branch Block Beat (Battement du Bloc de Branche Gauche) (LBBB)

Le bloc de branche gauche (LBBB) est une anomalie de la conduction cardiaque observée sur l'électrocardiogramme (ECG) [17].

Dans cette condition, l'activation du ventricule gauche du cœur est retardée, ce qui provoque la contraction du ventricule gauche plus tard que le ventricule droit.

Le signal ECG de cette maladie montré sur la figure I-17.

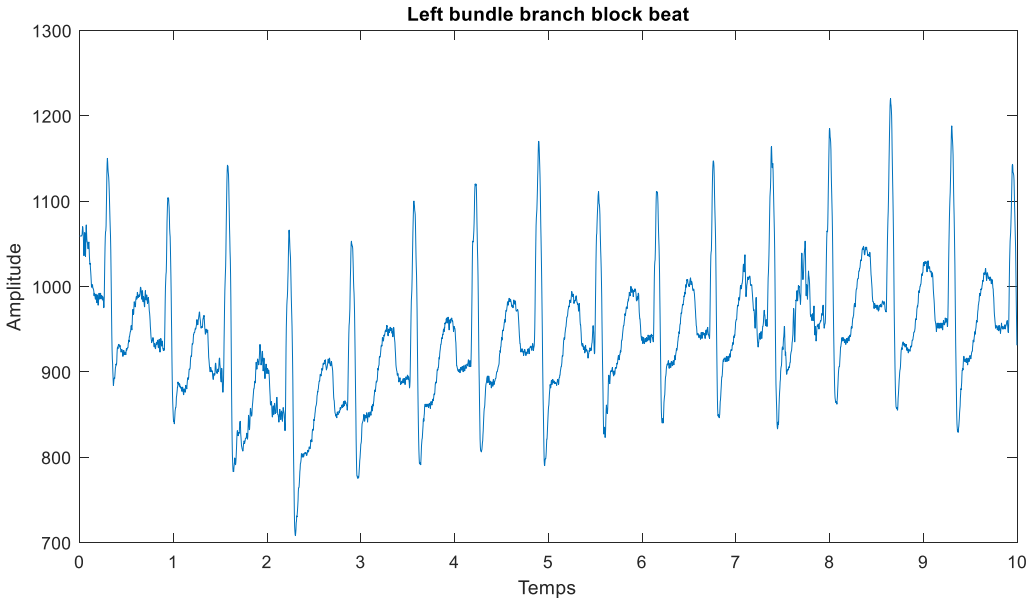



**Figure I-17:** Left bundle branch block beat

Parmi les causes de LBBB sont :

- Sténose aortique.
- Cardiomyopathie dilatée.
- Infarctus aigu du myocarde.
- Maladie coronarienne étendue.
- Maladie primaire du système de conduction électrique cardiaque [17].

## I.15-Conclusion

Dans ce chapitre, nous avons décrit l'anatomie du cœur humaine avec des figures et des schémas explicatifs, nous avons présenté les différentes caractéristiques d'un signal ECG et les problèmes cardiaques avec un signal parfait de chaque cas. Cette étude va nous permettre de bien analyser les différents signaux issus d'un ECG et qui seront utiles par la suite dans la classification des maladies cardiaques.

# Chapitre II : Intelligence Artificielle et Internet des Objets

## II.1-Introduction

Ce chapitre intitulé « **Intelligence Artificiel et Internet des Objets** », consiste à découvrir le domaine d'intelligence artificielle et ses sous-domaines tels que : l'apprentissage automatique (Machine Learning), l'apprentissage en profondeur (Deep Learning) avec une description des différents types de DL. D'autre part, on veut illuter : les réseaux neurones en générales (réseaux neurones artificiels et réseaux neurones à convolutions CNN) tout en détaillant les différents paramètres de chaque couche dans l'architecture des réseaux CNN. Enfin on veut faire une description de l'internet des objets et ses domaines d'application.

## II.2-Intelligence Artificielle

L'intelligence artificielle est une vaste branche de l'informatique qui s'intéresse à la construction des machines intelligentes programmées pour penser comme des humains et imiter leurs actions [18].

Le terme peut également être appliqué à toute machine qui présente des traits associés à un esprit humain tel que l'apprentissage et la résolution de problèmes, le raisonnement, la découverte de sens, la généralisation ou l'apprentissage des expériences passées.

Malgré les progrès continus de la vitesse de traitement informatique et la capacité de mémoire, il n'existe pas encore de programmes capables d'égaliser la flexibilité humaine sur un plus large domaine ou dans des taches nécessitant beaucoup de connaissances quotidiennes. D'autre part il existe certains programmes qui ont atteint les niveaux de performance d'experts et de professionnels humains dans l'exécution de certaines tâches spécifiques, de sorte que l'intelligence artificielle dans ce sens limité se retrouve dans des applications aussi diverses que le diagnostic médical, les moteurs de recherche informatiques et la reconnaissance de la voix ou de l'écriture manuscrite.

L'intelligence artificielle est une science interdisciplinaire aux approches multiples, mais les progrès de l'apprentissage automatique et de l'apprentissage en profondeur créent un paradigme changement dans pratiquement tous les secteurs de l'industrie technologique [18].

L'intelligence artificielle comprend désormais de nombreux sous-domaines, utilisant une variété de techniques, telles que :

- **Les Réseaux Neuronaux :** ils sont utilisés dans :
  - La modélisation du cerveau.
  - Prédiction de séries temporelles.

  - Classification.
- **Calcul évolutif :**
  - Algorithmes génétiques.
  - Programmation génétique.
- **Vision :**
  - Reconnaissance des objets.
  - Compréhension d'images.
- **Robotique :**
  - Contrôle intelligent.
  - Exploration autonome.
- **Systèmes experts :**
  - Systèmes d'aide à la décision.
  - Systèmes d'enseignement.
- **Traitement de parole :**
  - Reconnaissance et production de parole.
- **Traitement du langage naturel :**
  - Traduction automatique.
- **Planification :**
  - Planification.
  - Jeu.
- **Apprentissage automatique :**
  - Apprentissage de l'arbre de décision.
  - Apprentissage de l'espace des versions.

L'intelligence artificielle (AI) consiste à progresser la capacité d'apprendre aux machines et à imiter les actions humaines. Elle est principalement utilisée pour aider les humains, notamment en remplaçant les emplois très répétitifs et peu qualifiés. L'intelligence artificielle est un domaine complet qui comprend :

- Calcul évolutif.
- Systèmes experts.
- Intelligence artificielle symbolique.
- Machines à vecteurs de support (SVM).
- L'apprentissage automatique.
- L'apprentissage en profondeur.

- L'apprentissage par renforcement.

Et bien d'autre domaines [18].

## II.3-Machine Learning

L'apprentissage automatique (Machine Learning) est un sous-domaine de l'intelligence artificielle, le terme faisant référence à la capacité des systèmes informatiques à trouver indépendamment des solutions aux problèmes en reconnaissant des modèles dans des bases de données. Il construit aussi un modèle mathématique basé sur des données d'échantillons appelé « données d'entrainement », afin de rendre prédictions ou décisions sans être explicitement programmé pour exécuter la tâche [19].

Afin de permettre au logiciel de générer indépendamment des solutions, l'action préalable des personnes est nécessaire, tels que les algorithmes et les données nécessaires qui doivent être introduites dans les systèmes à l'avance et les règles d'analyse respectives pour la reconnaissance des modèles. Une fois ces deux étapes franchies, le système peut effectuer les tâches suivantes par l'apprentissage automatique :

- Trouver, extraire et synthétiser les données pertinentes.
- Faire prédictions sur la base des données d'analyse.
- Calcul des probabilités pour des résultats spécifiques.
- S'adapter à certaines évolutions de manière autonome.
- Optimisation des processus basés sur des modèles reconnus [19].

Un ensemble de donnée est fourni comme suit :

- **Ensemble de données d'entrainement :** C'est la base de données de connaissance classée et utilisée pour entrainer notre modèle d'apprentissage. Pendant cette phase on peut régler les paramètres de notre modèle en fonction des performances résultantes.
- **Ensemble de données de test :** Cette base de données non classée se compose des exemples jamais vus ultérieurement.

L'apprentissage automatique vise à établir un classificateur en apprenant l'ensemble d'apprentissage, puis à évaluer les performances du classificateur à travers l'ensemble de tests.

D'une manière générale, la Machine Learning fonctionne de la même manière que l'apprentissage humain. Par exemple : si un enfant voit des images avec des objets spécifiques dessus, il peut apprendre à les identifier et à les différencier. La Machine Learning fonctionne de la même manière. Grâce à la saisie de données et à certaines commandes, l'ordinateur peut

apprendre à identifier certains objets (personnes, objet…) et à les distinguer. A cet effet, le logiciel est alimenté en données et est formé. Par exemple :

Le programmeur peut indiquer au système qu'un objet particulier est un être humain (='Humain') et qu'un autre objet n'est pas un être humain (='Non Humain'). Le logiciel reçoit une rétroaction continue du programmeur. Ces signaux de rétroaction sont utilisés par l'algorithme pour adapter et optimiser le modèle. Avec chaque nouvel ensemble de données introduites dans le système, le modèle est encore optimisé afin qu'il puisse clairement distinguer les « Humains » et les « non Humains ».

### II.3.1-Types d'apprentissage automatique

#### II.3.1.1-Apprentissage Supervisé

Au cours d'apprentissage supervisé, des exemples de modèles sont définis à l'avance. Afin d'assurer une allocation adéquate des informations aux groupes de modèles respectifs des algorithmes, ceux-ci doivent alors être spécifiés. En d'autres termes, le système apprend sur la base de paires de données d'entrée et de sorties. Au cours de l'apprentissage contrôlé, un programmeur agit comme une sorte d'enseignant, fourni les valeurs appropriées pour une entrée particulière. L'objectif est d'entrainer le système dans le cadre de calculs successifs avec différentes entrées et sorties et d'établir des connexions [20].

#### II.3.1.2-Apprentissage Non Supervisé

Dans ce type d'apprentissage, l'intelligence artificielle apprend sans valeurs cibles prédéfinies et sans récompenses. Il est principalement utilisé pour l'apprentissage de la segmentation (Clustering). La machine essaie de structurer et de trier les données saisies selon certaines caractéristiques. Par exemple, une machine pourrait (très simplement) apprendre que des pièces de différentes couleurs peuvent être triées selon la couleur [20].

#### II.3.1.3-Apprentissage par Renforcement

Cette méthode d'apprentissage vise à utiliser des observations recueillies à partir de l'interaction avec l'environnement pour accomplir des actions qui maximiseraient la récompense ou minimiseraient le risque. Afin de produire des programmes intelligents (également appelé agents), ce type d'apprentissage passe par les étapes suivantes :

- L'état d'entrée est observé par l'agent.
- La fonction de prise de décision est utilisée pour rendre l'agent capable d'effectuer une action.

- Une fois l'action effectuée, l'agent reçoit une récompense ou le renforcement de l'environnement.
- Les informations sur la paire état-action concernant la récompense sont stockées.

En utilisant les informations stockées, les politiques pour une condition donnée peuvent être ajustées en fonction des actions pour fournir à notre agent (réseaux) une aide à la décision (prendre une décision) optimale [20].

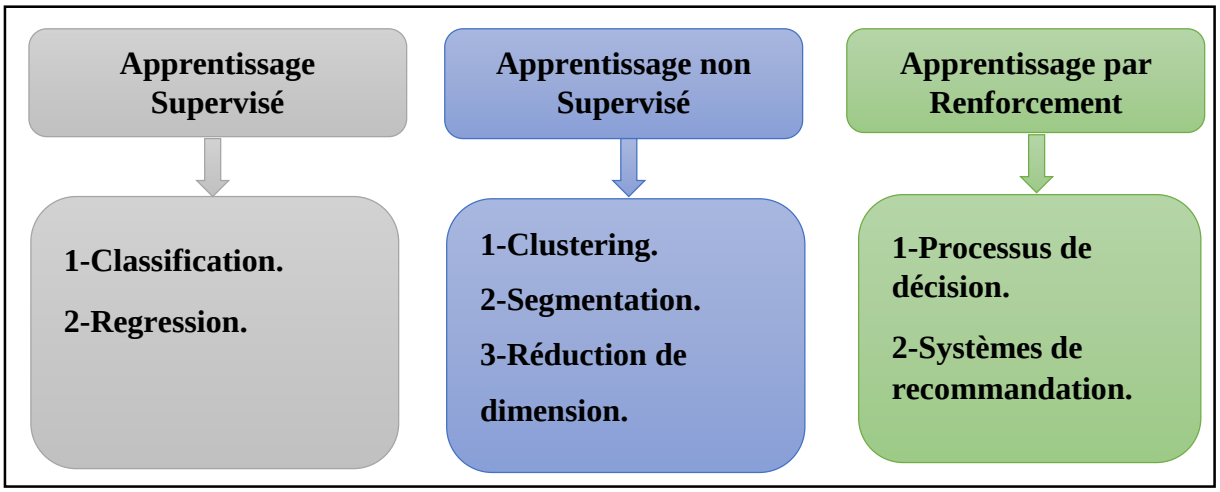



**Figure II-1 :** Types d'apprentissage automatique

## II.4-Deep Learning

L'apprentissage en profondeur (Deep Learning) est un type d'intelligence artificielle dérivé de l'apprentissage automatique, qui est essentiellement un réseau de neurones à trois couches ou plus. Ces réseaux neurones tentent de simuler le comportement du cerveau humain, lui permettant « d'apprendre » à partir d'un nombre extrême de données.

L'apprentissage en profondeur pilote de nombreuses applications et services d'intelligence artificielle (AI) qui améliorent l'automatisation en effectuant des taches analytiques et physiques sans intervention humaine. La technologie d'apprentissage en profondeur est à l'origine de produits et services de tous les jours tels que l'assistants numériques, les télécommandes de télévision vocale et la détection de la fraude par carte de crédit, ainsi que les nouvelles technologies telles que les voitures autonomes [21].

Le développement d'intelligence artificiel et ses sous-domaines par les années représenté dans la figure II-2.

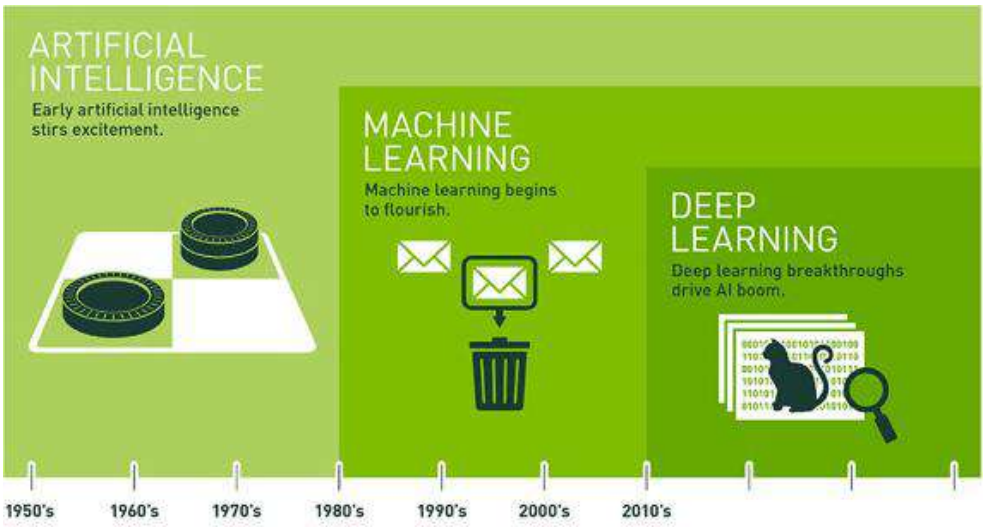



**Figure II-2 :** Domaines d'intelligence artificielle

Il existe deux types principaux d'apprentissage en profondeur :

### II.4.1-Apprentissage supervisé

Dans ce type d'apprentissage (voir figure II-3), les données d'entrainement que nous fournissons à notre système incluent des solutions [22].

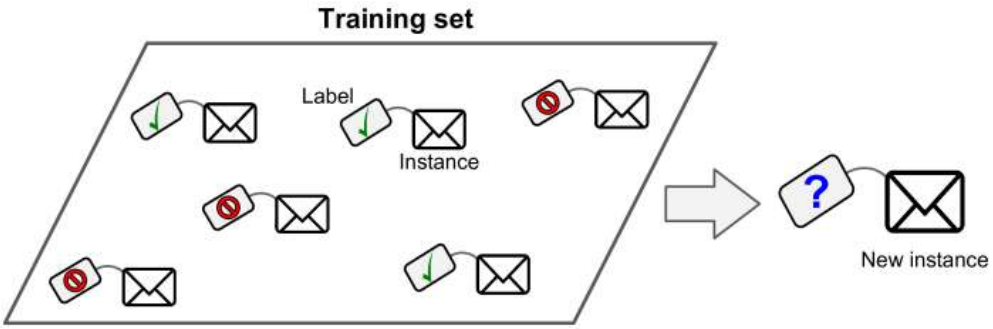



**Figure II-3 :** Un ensemble de donnée étiqueté (Classé) pour l'apprentissage supervisé et puis la classification d'une donnée inconnue [22]

Avec ce type d'apprentissage, le système peut apprendre à réaliser quelques étapes en étudiant plusieurs exemples de cette tâche.

Le système peut apprendre à reconnaitre une photo ou un objet après qu'on lui ait montré des millions photos ou objets.

Il est similaire à celui d'un étudiant qui passe un examen ou concours, en tenant Compte de ses connaissances antérieures, il fait de son mieux pour avoir la bonne réponse en examen.

### II.4.2-Apprentissage Non Supervisé

Ce type d'apprentissage est l'inverse de l'apprentissage supervisé. L'apprentissage non supervisé (voir figure II-4) prend uniquement les données sans classe (des données non classé), notre système tente de découvrir automatiquement les caractéristiques de chaque entrée sans indication de classe (sur les entrées) et classer chaque donnée selon ses caractéristiques.

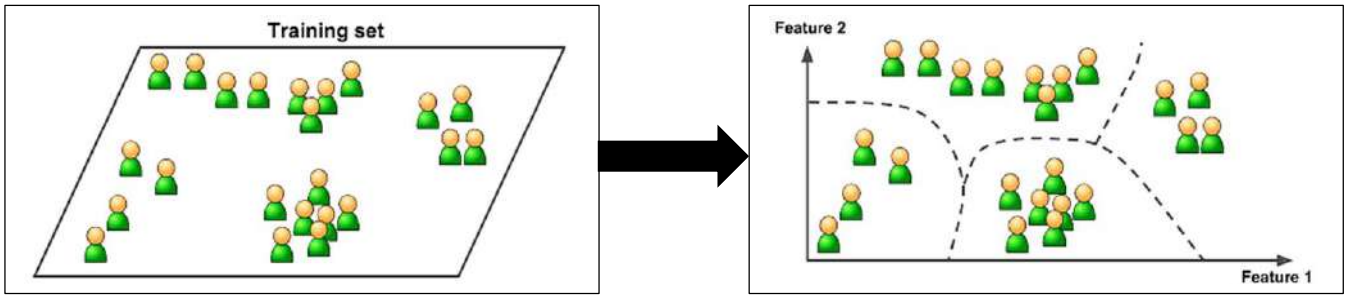



**Figure II-4 :** Apprentissage Non Supervisé, l'entrainement et classement des données selon leurs caractéristiques [22]

## II.5-Les Réseaux Neurones

Les réseaux neurones sont des modèles de calcul qui partagent certaines propriétés avec un cerveau humain dans lequel de nombreuses unités simples fonctionnent en parallèle sans aucune centralisation d'unité de contrôle.

Les poids entre les unités sont la façon (manière) principal d'évaluation de stockage d'information à long terme dans les réseaux neurones. La mise à jour des pondérations est la façon principal d'un réseau neurones pour apprendre des nouvelles informations.

Les poids entre les unités sont le principal moyen de stockage d'informations à long terme dans les réseaux de neurones. Le comportement des réseaux neurones dépend de sa propre architecture. L'architecture d'un réseau peut être définie par le nombre de neurones, le nombre de couches et les types de connexions entre ces couches.

### II.5.1-Neurone Biologique

Le cerveau humain est une collection de plus de 10 milliards de neurones interconnectés entre eux, chacun connecté à environ 10000 autres neurones. Le corps cellulaire du neurone s'appelle « Soma », ou' les entrées « Dendrites » et les sorties « Axones » relient soma à d'autres soma (figure II-5 ci-dessous). Chaque neurone est une cellule ou' s'effectuent des réactions biochimiques pour recevoir, traiter et transmettre des informations [21].

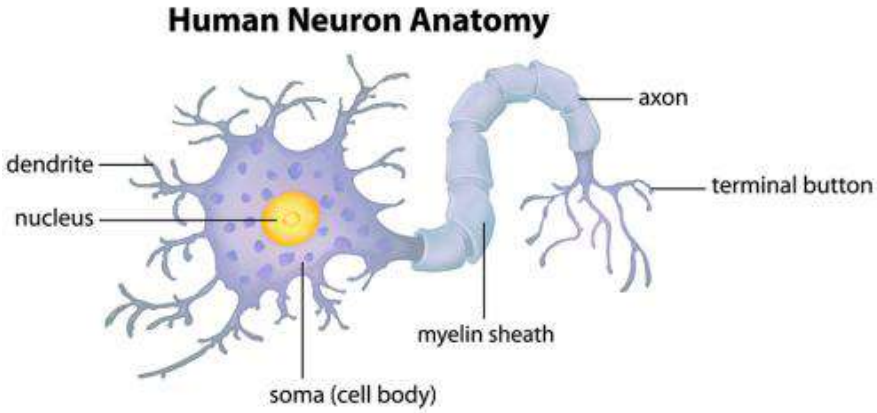



**Figure II-5 :** Neurone Biologique

Les réseaux arborescents de fibres nerveuses appelés dendrites sont connectés au corps cellulaire ou soma, où se trouve le noyau cellulaire. S'étendant du corps cellulaire une longue et unique fibre appelée axone, se ramifie en brins et sous brins, et est connectée à d'autres neurones par des synapses.

La transmission des signaux d'un neurone à un autre au niveau des synapses est un processus chimique complexe dans lequel des substances chimiques spécifiques sont libérées de l'extrémité émettrice de la jonction. L'effet est d'élever ou d'abaisser le potentiel électrique à l'intérieur du corps de la cellule réceptrice. Si le potentiel atteint le seuil, une impulsion est envoyée dans l'axone et la cellule est « activée » [23].

### II.5.2-Neurone Artificiel

Inspirés des réseaux neurones biologiques, les réseaux neurones artificiels sont des systèmes informatiques massivement parallèles constitués d'un très grand nombre de processeurs simples avec de nombreuses interconnexions. Les modèles ANN tente d'utiliser certains principes « organisationnels » censés être utilisé chez l'humain. Un type de réseau considère les nœuds comme des « neurone artificiels ». Ceux-ci sont appelés réseaux neurones artificiels ANN. Un neurone artificiel est un modèle informatique inspiré des neurones naturels (comme le montre la figure II-6).

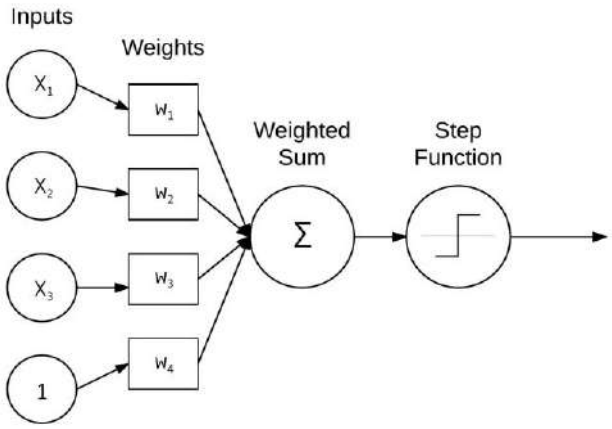



**Figure II-6 :** Structure d'un Réseau Neurone Artificiel ANN

Etant donné que la fonction des ANN est de traiter l'information, ils sont principalement utilisés dans les domaines qui y sont liés. Il existe une grande variété d'ANN qui sont utilisés pour modéliser de vrais réseaux neurones et d'étudier le comportement et le contrôle chez les animaux et les machines, mais il existe des ANN qui sont utilisées à des fins d'ingénierie, telle la reconnaissance de formes, la prévision et la compression de données. Ceux-ci se composent essentiellement d'intrants (comme les synapses), qui sont multipliés par des poids. Les poids attribués avec chaque flèche représentent le flux d'information. Ces poids sont ensuite calculés par une fonction mathématique qui détermine l'activation des neurones. Une autre fonction calcule la sortie des neurones artificiels (parfois en fonction d'un certain seuil). Les neurones de ce réseau additionnent simplement leurs entrées. Puisque les neurones d'entrée n'ont qu'une seule entrée, leur sortie sera l'entrée qu'ils ont reçue multipliée par un poids [21].

### II.5.3-Architecture des réseaux neurones artificiels

Chaque réseaux neurones est constitué par des neurones artificiels qui reçoit des entrées pour produire une entrée. Habituellement le réseau neurone se compose par :

- Une couche d'entrée.
- Des couches cachées.
- Une couche de sortie.

Comme la figure II-7.

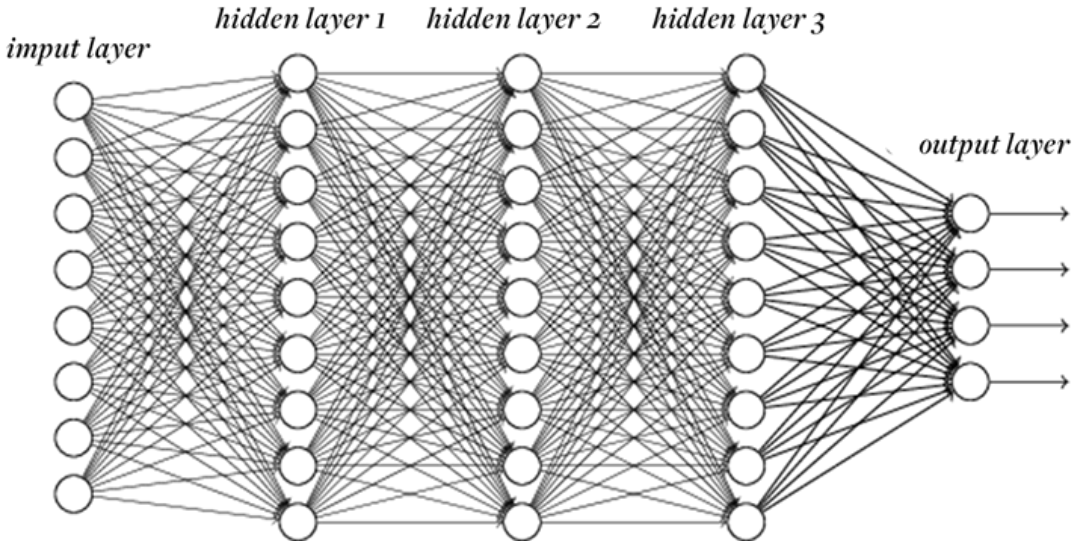



**Figure II-7 :** Architecture d'un Réseau Neurone Artificiel

### II.5.4-Le Perceptron

C'est une cellule dans le réseau neurone artificiel. Chaque neurone calcul sa sortie comme une somme pondérée des entrées, décalée d'un biais et suivie d'une fonction d'activation comme la figure II-8 [24].

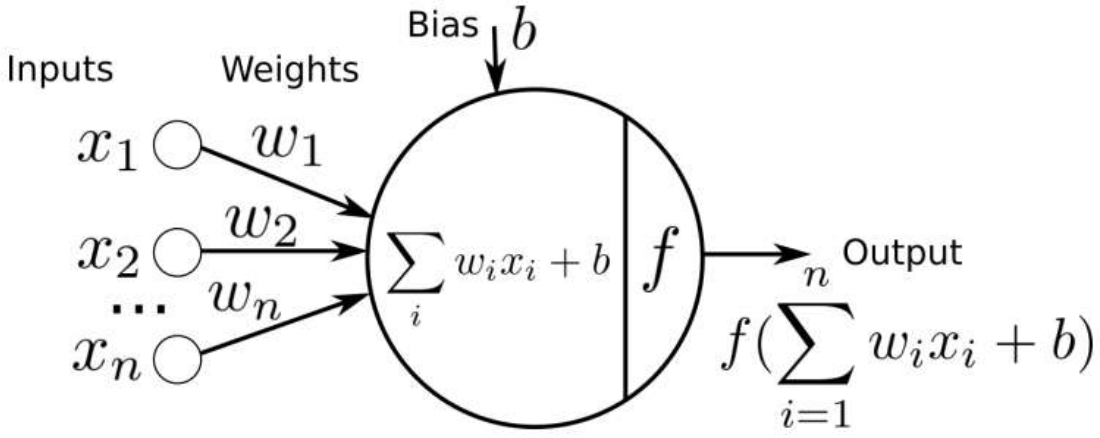



**Figure II-8 :** Principe de calcul d'un neurone artificiel

### II.5.5-Les Fonctions d'activation

Pour pouvoir exprimer des relations non linéaires entre l'entrée et la sortie, des fonction d'activation non linéaires sont appliquées. Lorsque la non linéarité est utilisée, un réseau neuronal devient une approximation de fonction universel [24].

#### II.5.5.1-Fonction seuil (Echelon unitaire)

C'est la fonction d'activation la plus simple, qui est utilisée par l'algorithme Perceptron.

$$F(x) = \begin{cases} 1 & x > 0 \\ 0 & Ailleurs \end{cases} \quad \text{(II-1)}$$

Et nous pouvons le voir dans l'équation ci-dessus, il s'agit d'une fonction d'échelon unitaire très simple. Si le pondéré :

$$\begin{cases} \sum_{i=1}^{n} w_i x_i > 0 \rightarrow \quad output = 1 \\ \sum_{i=1}^{n} w_i x_i > 1 \rightarrow \quad output = 0 \end{cases} \quad \text{(II-2)}$$

Le graphe de cette fonction est affiché dans la figure II-9 [21].

### II.5.5.2-La Fonction de Sigmoïde

Cette fonction est la plus courante dans l'histoire des réseaux neurones.

Cette fonction est le meilleur choix pour l'apprentissage de la simple fonction en escalier car elle est :

- Continue et différentiable partout.
- Symétrique autour de l'axe y.
- Se Rapproche asymptotiquement de ses valeurs de saturation.

$$F(x) = \frac{1}{1 + e^{-x}} \quad \text{(II-3)}$$

$$x = \sum_{i=1}^{n} w_i x_i \quad \text{(II-4)}$$

Cette fonction « **Sigmoïde** » est utilisée dans la dernière couche (étage) de sortie du réseau neurone pour la classification car ils convertissent l'entrée en une plage de probabilité (0, 1).

La fonction « **Sigmoïde** » est utilisée pour une classification binaire [21].

### II.5.5.3-La Fonction Softmax

La fonction « **Softmax** » est utilisée dans la dernière couche (étage) de sortie du réseau neurones pour la classification car elle convertit l'entrée en une plage de probabilité (0, 1).

Cette fonction « **Softmax** » utilisée pour la classification multi- classes [24].

### II.5.5.4-La Fonction Tangente Hyperbolique « Tanh »

Cette fonction a une forme similaire à la fonction « **Sigmoïde** », a également été fortement utilisée comme fonction d'activation jusqu'à la fin des années 1990. L'équation de cette fonction est la suivante :

$$F(x) = Tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad \text{(II-5)}$$

Cette fonction est centrée sur zéro, mais les gradients sont toujours tués lorsque les neurones deviennent saturés [21].

C'est une fonction non linéaire à utilisation générale (couramment utilisée).

Le graphe de cette fonction est affiché dans la figure II-9.

#### II.5.5.5-La Fonction ReLu (Rectified Linear Unit)

C'est une fonction non linéaire (voir figure II-9) à utilisation générale couramment utilisée.

L'équation de cette fonction :

$$F(x) = \max(0, x) \quad \text{(II-6)}$$

La fonction **ReLu** appelée aussi « fonction de rampe », la fonction est nulle pour les entrées négatives, mais augmente ensuite linéairement pour les valeurs positives. La fonction **ReLu** n'est pas saturable et elle est également extrêmement efficace en terme de calcul [25].

Cette fonction est la plus utilisée pour ne pas activer tous les neurones, et en même temps permet de réduire la quantité de calculs effectués lorsque les valeurs négatives sifflent. Alors toutes les valeurs inférieures ou égales à zéro deviennent nulles tandis que les valeurs positives sont évaluées.

#### II.5.5.6-La Fonction Leaky ReLu

C'est une fonction variante des ReLu, appelée Leaky ReLu permet un petit gradient non nul lorsque l'unité n'est pas active :

$$F(x) = \begin{cases} x & x \geq 0 \\ \alpha x & Ailleurs \end{cases} \quad \text{(II-7)}$$

En traçant la fonction ci-dessus sur la figure II-9**,** nous pouvons voir que cette fonction est en effet autorisée à prendre une valeur négative, contrairement aux ReLu traditionnels qui 'brident' la sortie de la fonction à zéro [25].

#### II.5.5.7-La Fonction ELU (Unités Linéaires Exponentielles)

Cette fonction est définie selon l'équation suivante :

$$F(x) = \begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & Ailleurs \end{cases} \quad \text{(II-8)}$$

La valeur de $\alpha$ est constante et définie lorsque l'architecture du réseau instanciée.

Une valeur typique pour $\alpha = 1.0$, la figure II-9 visualise la fonction d'activation de l'ELU.

Grace aux travaux de Clevert, les ELU obtiennent souvent une grande précision de classification que les ReLu [26].

### II.5.5.8-La Fonction Argmax

Elle est utilisée pour obtenir une classe discrète à partir de ces valeurs de probabilité continues, une fonction de décision est appliquée. La fonction de décision la plus simple pour la classification multi classe à étiquette unique consiste à prendre la grande valeur en utilisant la fonction « **Argmax** » [24].

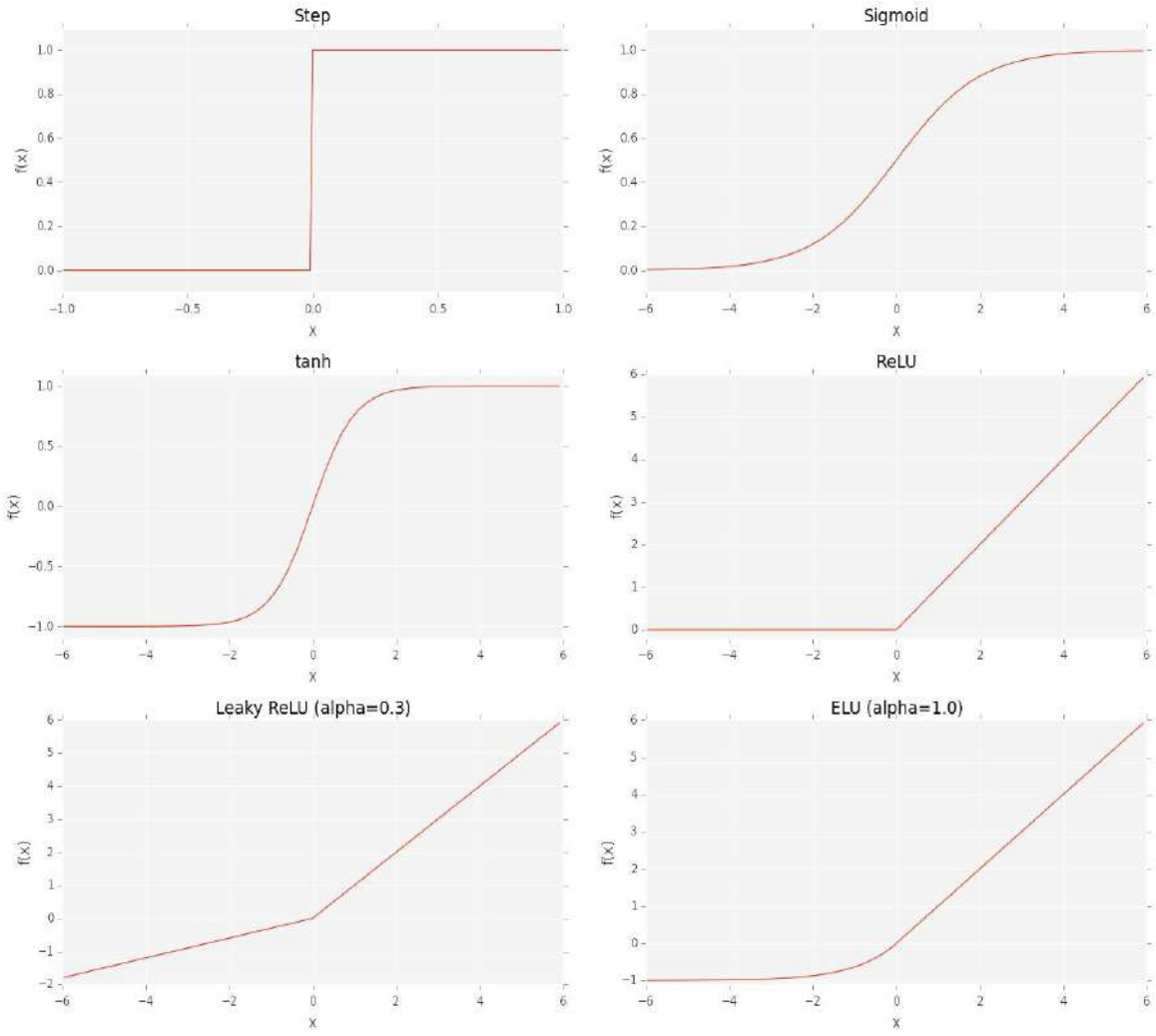



**Figure II-9 :** Les Différentes Fonctions d'Activation

### II.5.6-Types des réseaux neurones

Dans ce domaine, il existe plusieurs architectures des réseaux neurones utilisées dans les expérimentations de ces architectures. Les architectures les plus utilisables sont comme suit :

- FNN (Feed Forward Neural Network).

- RBF (Radial Basis Function Neural Network).
- SOM (Self Organization Map Neural Network).
- RNN (Recurrent Neural Network).
- CNN (Convolutional Neural Network).
- MNN (Modular Neural Network).

### II.5.6.1-Réseau neurones de convolution CNN

Le réseau neurones le plus populaires est le réseau neurones de convolution CNN. Son nom indique une opération arithmétique linéaire entre les vecteurs dite convolution.

Les réseaux CNN sont des technologies réussis qui ont été appliquées à des problèmes ou les données d'entrée sur lesquelles des prédictions doivent être faites, ont une topologie connue, comme une série chronologique (des signaux (vecteur 1D)) ou une image (matrice 2D). Les réseaux CNN ont introduit l'apprentissage en profondeur dans les temps modernes, ils résolvent l'un des problèmes de calcul les plus cruciaux de l'ère numérique de la vision par l'ordinateur. Avec la popularité des CNN, une augmentation de la recherche sur l'apprentissage en profondeur a été observée et se poursuit aujourd'hui [27].

figure II-10 représente une architecture de base pour un réseau CNN.

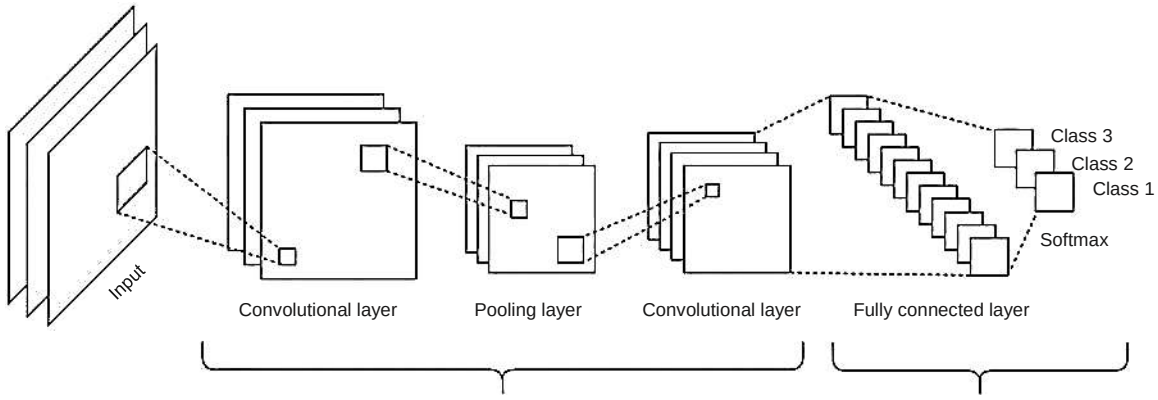



**Figure II-10 :** Architecture de Base pour un réseau CNN

### II.5.6.2-Architecture d'un réseau CNN

- **Couche de Convolution**

  Une couche de convolution est un composant fondamental de l'architecture d'un réseau CNN qui effectue l'extraction des caractéristiques, qui se propage généralement vers l'avant sur un ensemble de données d'apprentissage, et des paramètres apprenables. Il consiste en une combinaison d'opérations linéaire et non linéaire-qui sont comme suit :

- Opération de convolution (voir figure II-11).
- Fonction d'activation.

**Convolution :** La convolution est un type spécialisé d'opération linéaire utilisée pour l'extraction des caractéristiques, où un petit vecteur (tableau) de nombres, appelé filtre (Kernel)(noyau), il est appliqué sur l'entrée, qui est un vecteur de nombre, appelé donnée (Tensor). Un produit élément par élément entre chaque élément du filtre et la donnée d'entrée est calculé à chaque emplacement du tenseur et additionné pour obtenir la valeur de sortie dans la position correspondante de la donnée de sortie dans la position correspondante au donnée de sortie, appelée une carte de caractéristique. Cette procédure est répétée en appliquant plusieurs filtres pour former un nombre arbitraire des cartes de caractéristiques, qui représente les différentes caractéristiques des données d'entrée.

Les différents filtres peuvent être considérés comme des extracteurs des différentes caractéristiques. Il existe deux différents hyper paramètres clés qui définissent les opérations de convolution qui sont :

- La taille du filtre : généralement dans la plupart des applications est une matrice carrée ou un vecteur de tailles impaires qui sont :
  - Pour une matrice : 3 × 3, mais parfois 5 × 5 ou 7 × 7.
  - Pour un vecteur : 3 × 1, mais parfois 5 × 1 ou 7 × 1.
- Le nombre des filtres : détermine la profondeur des cartes d'entités de sortie.

L'opération de convolution décrite ne permet pas au centre de chaque filtre de chevaucher l'élément le plus externe de la donnée d'entrée et réduit la hauteur et la largeur de la carte des entités de sortie par rapport au donnée d'entrée [28].

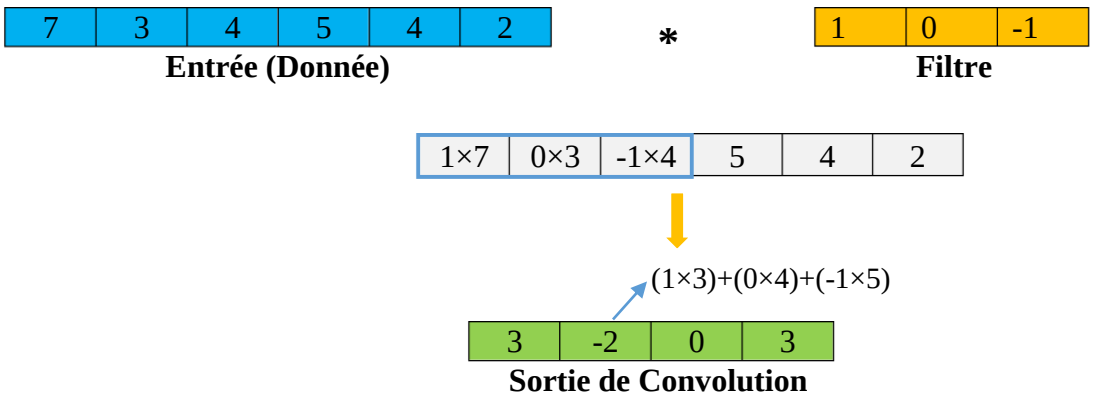



**Figure II-11 :** Opération de Convolution

- **Remplissage zéro (Padding) :** c'est une technique utilisée pour résoudre le problème de cette convolution qui ne permet pas au centre de chaque filtre de chevaucher les éléments les plus externe de donnée d'entrée, donc pour résoudre

ce problème on utilise cette technique de remplissage zéro (voir figure II-12), ou des lignes et des colonnes de zéros sont ajoutées de chaque côté de donnée d'entrée, de manière à ajuster le centre de filtre sur l'élément le plus externe et à conserver la même dimension dans le plan grâce à l'opération de convolution. Les architectures de CNN modernes utilisent généralement :

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

S'il n'y a pas de remplissage zéro, chaque carte d'entités (caractéristique) successive deviendrait plus petite après l'opération de convolution [28].

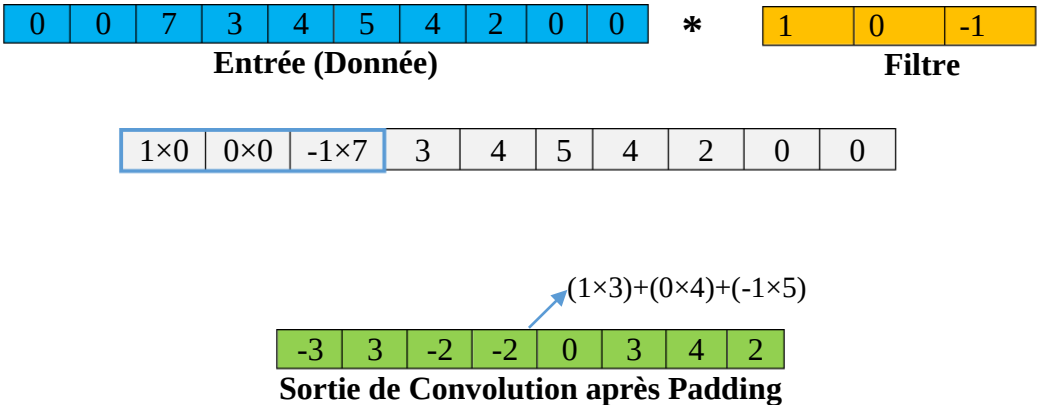



**Figure II-12 :** Opération de Padding

- ➢ **Foulé (balayage) (Stride) :** c'est la distance entre deux positions successives du filtre appelée foulée (stride), ce qui définit également l'opération de convolution. Le choix commun dans la plupart des applications d'une foulée (stride) est 1 ; cependant, une foulée (stride) supérieure à 1 est parfois utilisée afin d'obtenir (pour effectuer) un sous-échantillonnage des cartes d'entités.
- ➢ **Fonction d'activation :** les sorties d'une opération linéaire de convolution sont passés à travers une fonction non linéaire, telles que ReLu, ELu, LReLu où l'une des autres fonctions d'activation. ReLu est la fonction d'activation la plus utilisée dans la plupart des réseaux artificiels, nous indiquons que les fonctions d'activation sont appliquées à l'intérieur de l'architecture du réseau [28].

figure II-13 représente l'opération d'activation.

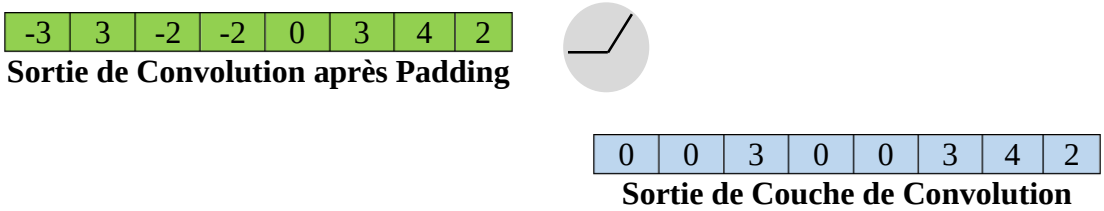



**Figure II-13 :** Sortie de Fonction d'activation

- **Couche de Pooling**

C'est une couche de regroupement qui donne une opération de sous échantillonnage typique qui réduit progressivement la taille (la longueur) des entrées, c'est-à-dire de diminuer le nombre des paramètres de calcul dans notre réseau CNN. Pooling aide à contrôler l'overfitting (sur ajustement). Il existe deux types de Pooling :

- **MaxPooling :** C'est la fonction la plus utilisable (courante) surtout dans les opérations de Max Pooling, qui extrait les patches des cartes d'entités d'entrée, il prend la valeur maximale et ignore tous les autres valeurs dans chaque patch. Un Max Pooling de taille 2×1(2×2) avec un stride (foulée) de 1(2), est le plus utilisé en pratique (voir figure II-14) [21].

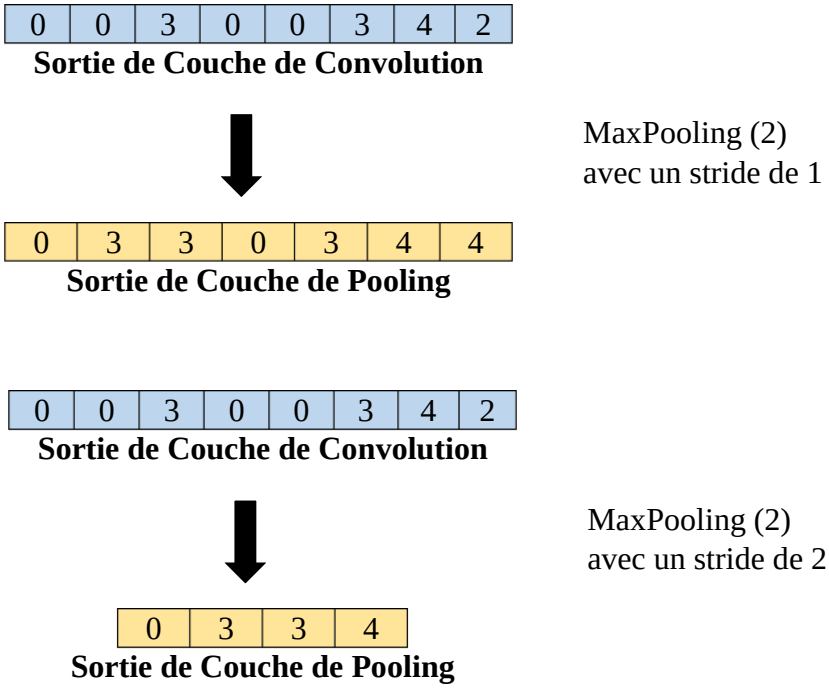



**Figure II-14 :** Opération de Max Pooling

- **Average Pooling (la Moyenne Pool) :** C'est une fonction similaire au Max Pooling, sauf que cette fonction extrait les patches des cartes d'entités d'entrée, prend la valeur moyenne et ignore toutes les autres valeurs dans chaque patch. Un Average Pooling (voir figure II-15) de taille 2×1(2×2) avec un stride (foulée) de 1(2) est le plus utilisé en pratique [28].

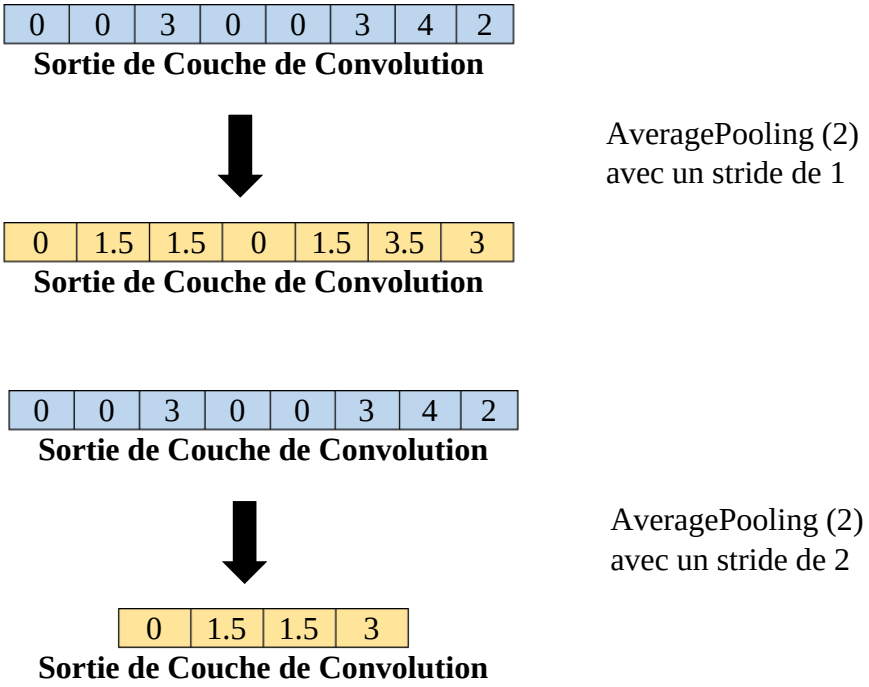



**Figure II-15 :** Opération de Average Pooling

- **Globale Average Pooling :** C'est une autre opération de Pooling qui mérite d'être notée, c'est une moyenne Pooling global. Une moyenne Pooling globale effectue un type extrême de sous-échantillonnage, ou' une carte d'entités (caractéristiques) avec une taille de longueur de 1 en prenant simplement la moyenne de tous les éléments de chaque carte d'entités (caractéristiques). Cette opération est généralement appliquée une seule fois avant la couche couche entièrement connectées. Les avantages de l'application de cette fonction sont :
  - Réduit le nombre de paramètres apprenables.
  - Permet au réseau CNN d'accepter des entrées de taille variable [28].

- **Batch Normalization (Normalisation par lots)**

Il est introduit pour la première fois en 2015 par **Ioffe** et **Szgedy** dans leur article. Cette couche de normalisation (abrégé par **BN)**, comme leur nom l'indique, permet d'accélérer l'apprentissage du réseau et en réduisant le décalage interne des covariances. Il est utilisé pour normaliser les activations d'un volume d'entrée donné avant de le passer dans la couche suivante du réseau. Mais il existe un inconvénient de la normalisation par lots (Batch Normalization) qui se traduit par le fait qu'elle peut ralentir le temps nécessaire pour former le réseau (même si vous aurez besoin de moins d'époques pour obtenir une précision raisonnable) de 2 à 3 fois en raison du calcul des statistiques par lot et normalisation [21].

- **Dropout (L'abandon)**

L'abandon (voir figure II-16) est en fait une forme de régularisation qui vise à aider à prévenir le sur apprentissage en augmentant la précision des tests, peut être au détriment de la précision d'entrainement. Pour chaque mini lot (mini batch) de notre ensemble d'apprentissage, les couches d'abandon, avec une probabilité $p$, déconnectent de manière aléatoire les entrées de la couche précédente à la couche suivante dans l'architecture du réseau [21].

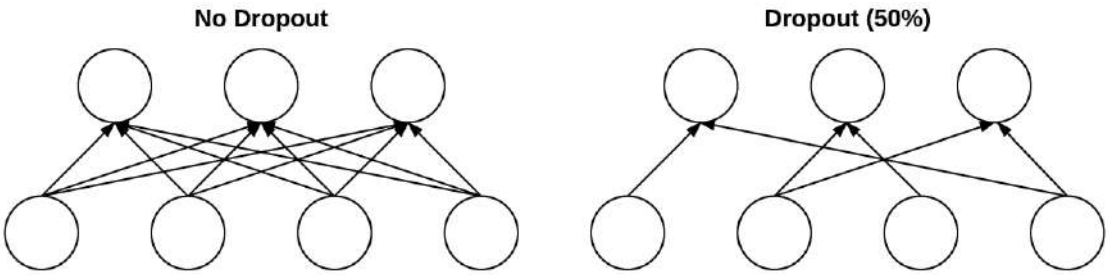



**Figure II-16 :** Opération d'abandon (Dropout)

- **Couche de connexion entière (Fully Connected layer)**

Les cartes d'entitées de sortie de la couche finale soit de convolution ou de Pooling sont généralement aplaties (Flatten), ce qui signifie transformer une matrice en un vecteur unidimensionnel (1D) et connecté à une ou plusieurs couches entièrement connectées, également appelées couches denses, où chaque entrée est connectée à chaque sortie par un poids apprenable. Une fois que les entités extraites par les couches de convolution et échantillonnées par la couche de Pooling, elles sont mappées par un sous ensemble de couches entièrement connectées aux sorties finales du réseau, telles que les probabilités pour chaque classe dans les taches de classification. La dernière couche entièrement connectée a généralement le même nombre de classes. Chaque couche entièrement connectée est suivie par une fonction non linéaire, telle que ReLu (voir figure II-17) [28].

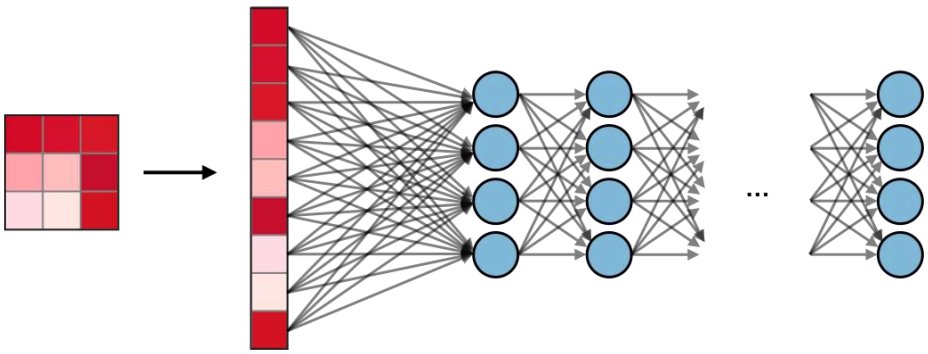


**Figure II-17 :** Couche entièrement connectée (Fully Connected layer)

- **Fonction d'activation de la dernière couche entièrement connectée (Softmax) :** la fonction d'activation appliquée à la dernière couche entièrement connectée est généralement différente des autres. Une fonction d'activation appropriée doit être sélectionnée en fonction de chaque tache. Cette fonction d'activation **Softmax** est appliquée à la tache de classification multi classe, qui normalise les valeurs réelles de sortie de la dernière couche entièrement connectée aux probabilités de classe cible ou' chaque valeur varie entre 0 et 1 et toutes les valeurs sont égales à 1 [28].

## II.6-Internet des Objets

L'IoT est l'acronyme d'Internet of Things (Internet des Objets). Selon l'UIT (Union Internationale des Télécommunications), l'Internet des Objets est défini comme (une infrastructure mondiale pour la société de l'information, qui permet de disposer de services qui évoluent en interconnectant des objets (physiques ou virtuels) grâce aux technologies de l'information et de la communication interopérables existantes ou en évolution) [29].

Internet des objets est une technologie récente qui crée un réseau mondial des machines et des appareils capables de communiquer et d'échanger des données entre eux via l'internet.

Il y a une différence entre l'internet des objets et l'internet. L'Internet des objets peut créer des informations sur les objets connectés pour analyser et prendre des décisions ; en d'autres termes, on peut dire que l'internet des objets est plus intelligent que l'internet. Les caméras de sécurité, les capteurs, les véhicules, les bâtiments et les logiciels sont des exemples de choses qui peuvent échanger des données entre elles [30].

La figure II-18 représente les différentes architectures d'iot dans : le passé, présent et les futures architectures.

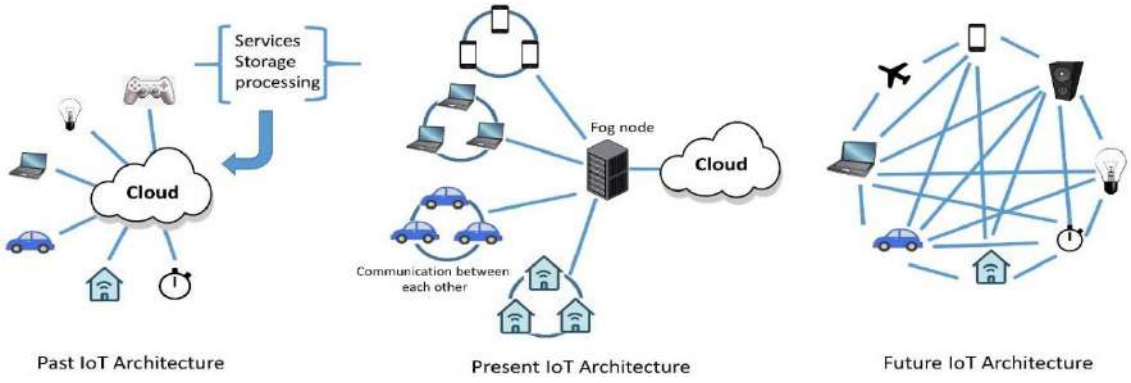



**Figure II-18 :** Architecture d'IOT

### II.6.1-Architecture d'IoT

Architecture IoT proposé d'un point de vue technique, est divisé en trois couches :

- **Couche de perception :** Sa fonction principale est d'identifier les objets et de recueillir les informations. Elle est formée principalement par : des capteurs, des actionneurs et de station de surveillance (tels que téléphone portable, tablette, PC, smartphone, etc.).
- **Couche de réseau :** il s'agit d'un réseau convergé composé par :
  - Des réseaux privés câblés/sans fil.
  - Internet des systèmes d'administration de réseau.

  Sa fonction principale est de transmettre les informations et les données obtenues à partir de la couche de perception.
- **Couche d'Application :** c'est un ensemble des solutions intelligentes qui appliquent la technologie IOT pour satisfaire les besoins des utilisateurs [31].

### II.6.2-Les Composantes d'IOT

Les composants d'IOT sont cinq. Un objet connecté est avant tout un objet ayant une fonction mécanique/électrique propre, qui peut être conçu pour être connectable directement, ou qui existe déjà, ou encore ajouté a posteriori. La fonction d'objet de connexion collecte les données à partir des capteurs pour traiter ces données, communiquer à l'aide d'une fonction de connexion et recevoir des instructions pour effectuer des actions.

Les fonctions de connexion d'objets nécessitent une puissance notamment lors du traitement des données directement dans l'objet.

#### II.6.2.1-Les Capteurs

Les capteurs sont des dispositifs permettant de mesurer et transformer une grandeur physique observée (comme température, luminosité, mouvement, etc) en une autre grandeur digitale utilisable par des logiciels. Il existe une grande variété de capteurs de (plusieurs) tous types, les objets connectés ont souvent la fonction de captation des grandeurs physique sur leurs lieux d'utilisation [32].




**Figure II-19 :** Les différentes Capteurs

### II.6.2.2-Réseaux des capteurs

Les réseaux des capteurs sans fil ne cessent de s'accroitre dans les différents domaines de la vie courante, à savoir le domaine militaire, agriculture, domotique, le domaine de santé. Cela est justifié par la contribution efficace de tels réseaux à la modernisation des techniques et méthodes de surveillances et de contrôle à distance des phénomènes dans des environnements de différentes natures.

Pour répondre aux besoins de communication entre eux, les capteurs sont équipés par des dispositifs sans fil pour transmettre et recevoir les données. Cependant, cela ne suffit pas pour rendre un ensemble de capteurs accessibles ou du moins de manière interopérables et simplifier. Un réseau des capteurs est un dispositif dont les plus petits éléments sont capables de faire des transmissions sans fil [33].

### II.6.2.3-Energie

C'est la contrainte la plus importante dans les réseaux des capteurs vitalité. Il existe 4 sources d'énergies sont :

- Alimentation filaires pour les objets ayant un accès à une prise de courant.
- Une pile ou une batterie pour les objets qui n'y ont pas un accès au prise ou de manière occasionnelle (recharge).
- Capteurs d'énergie (photo voltaïque, piezo électrique, thermoélectrique, cinétique, etc.) pour rallonger la durée de vie des objets à très faible consommation.
- Les objets passifs sans piles qui sont alimentés par des ondes électromagnétique.

L'énergie est l'un des grands défis pour les objets connectés, pour garantir une grande durée de vie possible sans maintenance [32].

### II.6.2.4-Actionneurs

Sont des dispositifs qui transforment la donnée digitale en un phénomène physique pour crée une action, c'est l'inverse du capteur. Par exemple : afficheurs, alarme, caméra, etc.

### II.6.2.5-Connectivité

La connectivité d'objet est assurée par une petite antenne radio fréquence qui va permettre la communication de l'objet vers un ou plusieurs réseaux IOT. Les objets pourront d'une part remonter des informations telles que :

- Leur identité.

- Leur état.
- Une alerte.
- Les données de capteurs.

D'autre part peuvent recevoir des informations telles que :

- Commandes d'action.
- Les données [30].

### II.6.3-Domaine d'application

L'IOT est censés être utilisables dans plusieurs domaines. Parmi les domaines principaux nous citons :

#### II.6.3.1-Villes intelligentes

Les villes intelligentes ou smart city sont en croissance dans les pays qui ont un grand développement technologique. Il existe des systèmes permettent de :

- Contrôle du fonctionnement de la ville.
- Contrôle des activités de peuple.
- Contrôle de la gestion des bâtiments.
- Contrôle de sécurité.

L'IOT permettent aux villes de devenir plus écologique et durable. Une ville est dite intelligente, alors il utilise l'IOT pour :

- Améliorer l'efficacité.
- Réduire ses couts.
- Améliorer le cadre de vie des citoyens [34].

Figure II-20 représente une infographie pour une ville intelligente.

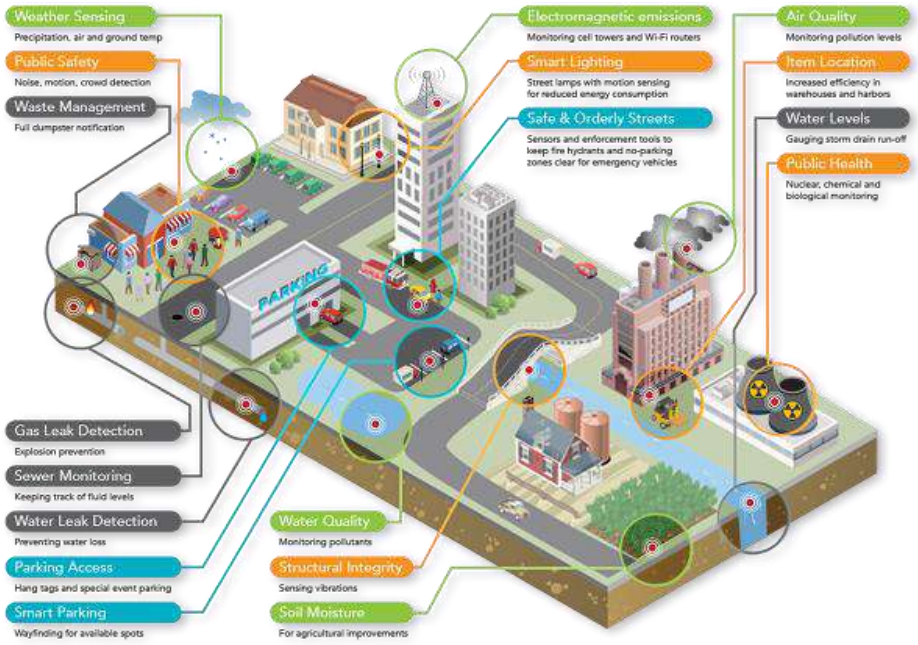



**Figure II-20 :** Infographie d'une ville intelligente

### II.6.3.2-Domotique

La maison du future sera un objet connecté à l'internet accessible à distance par ses propriétaires via des smartphones, tablette ou PC connectés. La porte, la télévision, le thermostat, le réfrigérateur, les parapluies, les montres, etc. de telle sorte tous ces objets sont connectée informe les parents par internet d'utilisation d'objet par les enfants.

Domotique regroupe l'ensemble des technologies permettant l'automatisation des équipements d'un habitat.

Elle vise à apporter des fonctions de confort comme :

- Commandes à distance.
- Gestion d'énergie (optimisation de l'éclairage et du chauffage)
- Sécurité comme des alarmes.
- Communiquer avec des personnes extérieures.

Il offerts des services comme :

- Il assure une protection des personnes par la prévenir des risques d'accident (fuite de gaz, incendie).
- Assure une vie quotidienne confortable principalement pour les personnes âgées et les handicapées.

- Facilite l'économie d'énergie grâce à la réactivité maitrisé d'une maison intelligente [35].



**Figure II-21 :** Maison intelligente

### II.6.3.3-Agriculture

L'agriculture intelligente a pour objet de renforcer la capacité des systèmes agricoles, de contribuer à la sécurité alimentaire en intégrant le besoin d'adaptation et le potentiel d'atténuation dans les stratégies de développement de l'agriculture durable [36].

L'application de l'IOT à un grand rôle dans l'agriculture pour faciliter les différentes problèmes d'agricultures comme l'arrosage et l'épandage d'engrais.



**Figure II-22 :** Agriculture intelligente

### II.6.3.4-La Santé (eHealth)

L'IOT a un rôle très important dans le secteur de la santé en introduisant des nouvelles technologies n'est pas seulement pour les hôpitaux, aussi dans le lieu de travail et à portée de main, qu'il s'agisse de suivre des documents médicaux, de suivre des signes vitaux ou télé thérapie.

L'IOT aura de nombreuses applications dans le secteur de la santé où l'objectif est d'arriver à prévenir des situations graves et de suivre à distance des patients atteints des

maladies chroniques et agir rapidement si cela s'est avéré nécessaire. Des capteurs corporels implantés dans le corps du patient récoltent des informations relatives aux paramètres médicaux, telles que la température, la glycémie, le rythme des battements du cœur ou même la tension artérielle (, saturation d'O2). Les informations récoltées seront stockées et traitées sur internet (plus précisément sur un cloud) et mise à la disposition du médecin qui pourra les consulter n'importe quand et depuis n'importe quel dispositif connecté à l'internet (exp : smart phone, tablette) [37].

La figure II-23 un exemple d'application d'IOT dans le domaine médical.

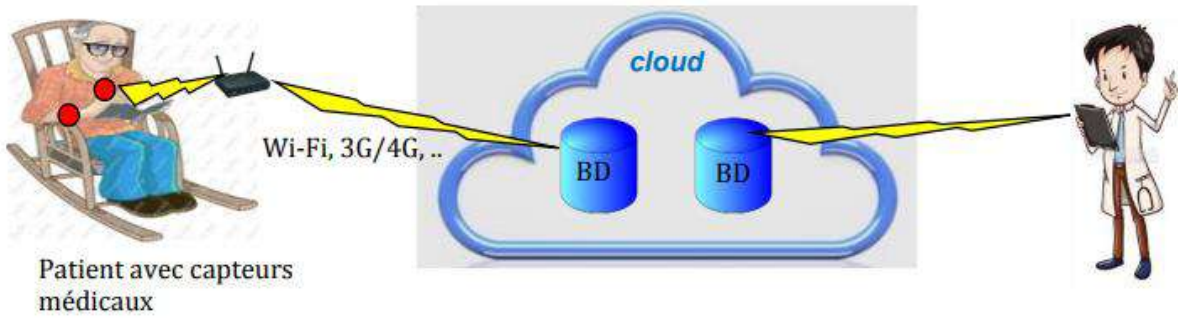



**Figure II-23 :** L'application d'IOT dans le domaine médical

## II.7-Conclusion

Dans ce chapitre, nous avons décrit l'intelligence artificiel et ses sous domaines : ML et DL avec ses différents types à l'aide des schémas explicatifs. Ensuite, nous avons présenté les réseaux neurones en général et les réseaux CNN en détail avec des figures et des organigrammes explicatifs des différentes couches d'architecture CNN. D'autre part nous avons aussi présenté l'internet des objets et ses domaines d'application.

# Chapitre III : Implémentation et test expérimental

## III.1-Introduction

Le présent chapitre a pour but d'analyser la base de données des signaux ECG, appliquer un pré traitement sur cette base selon des étapes bien définies, créer un réseau CNN 1D pour la reconnaissance et la classification des maladies cardiovasculaires. A partir des résultats obtenus, le meilleur des modèles CNN sauvegarder pour le test utilisé par la suite dans l'étude expérimental final. Dans ce cas des signaux ECG récolté par un capteur d'ECG AD8232 seront appliqués comme entrés dans notre réseau CNN pour une éventuelle testes. Ensuite, une implémentions sur une carte raspberry Pi 3B+ est effectué avec l'affichage des résultats au niveau d'une plateforme Node-Red.

## III.2-Base de donnée (1000 Fragments ECG)

Pour tester le classifieur réalisé, nous avons utilisé les signaux électrique du cœur (ECG) de la base de données MIT-BIH Arrhythmia disponible online sur le site [38].

Cette base de données est nommée : 1000 fragments ECG.

### III.2.1-Description de Base de données

Cette base de données est décomposée comme suite :

- Des signaux ECG de 45 patients dont :
  - 19 Femme de : $[23-89]ans$.
  - 26 Homme de : $[32-89]ans$
- Les signaux ECG contenaient 17 classes qui sont :
  - Rythme Sinusal Normal.
  - Rythme du simulateur cardiaque.
  - 15 types de dysfonctionnements cardiaques.
- Tous les signaux ECG ont été enregistrés à une fréquence d'échantillonnage de $360Hz$ et un gain de $200adu/mV$.
- Pour l'analyse, 1000 fragments (fichiers) de $10s$ (3600 échantillons (points)) du signal ECG (Sans chevauchement) ont été sélectionnés au hasard.

La description des signaux collectés est prescrite dans le tableau III-1 :

| Numéro | Classe (Maladies) | Fragments | Patients |
|---|---|---|---|
| **1** | Rythme Sinusal Normal | 283 | 23 |
| **2** | Fibrillation Auriculaire | 135 | 6 |
| **3** | Contraction Ventriculaire Prématurée | 133 | 14 |
| **4** | Battement du Bloc de Branche Gauche | 103 | 3 |
| Totale | | 654 | |

**Tableau III-1 :** Description des données collectées

L'aspect important est l'équilibre approprié des données. Le nombre de fragments de signal correspondant aux évolutions cardiaques physiologiques ne doit pas être significativement supérieur au nombre de signaux ECG pour les autres classes. Cela peut entrainer une augmentation artificielle de l'efficacité de la reconnaissance des maladies cardiaques. Pour avoir un bon équilibre de données entre les différentes maladies, on a étudié les quatre maladies mentionnées dans le tableau III-1.

## III.3-Les logiciels utilisés

### III.3.1-Matlab

Matlab c'est abréviation de « Matrix (Maths) LaBoratory », c'est une plateforme de programmation conçue spécifiquement pour les ingénieurs et les scientifiques afin d'analyser et de concevoir des systèmes et des produits qui transforment notre monde. Le cœur de Matlab est le langage Matlab, c'est un langage de programmation multi paradigme propriétaire et un environnement informatique numérique développé par MathWorks [39].

Matlab permet de :

- Analyse des données.
- Développement des algorithmes.
- Création des modèles et des applications.
- Manipulations matricielles.
- Traçage des fonctions et des données.
- Mise en œuvre d'algorithme.
- Création d'interfaces d'utilisateurs.
- Création d'interfaçage avec des programmes écrits dans d'autres langages [39].

Les différents détails sur Matlab dans l'annexe A.

### III.3.2-Python

Python c'est le langage de programmation à haut niveau, interprété et polyvalent. Sa philosophie de conception met l'accent sur la lisibilité du code grâce à l'utilisation d'une indentation significative. Python est dynamiquement typé et ramassé. Il prend en charge plusieurs paradigmes :

- Programmation.
- Y Compris la programmation structurée.
- Programmation des objets orientée.
- Fonctionnelle.

Il est souvent décrit comme un langage « piles incluses » en raison de ses bibliothèques standard complète.

Guidp Van Rossum a commencé à travailler sur Python à la fin des années 1980 en tant que successeur du langage de programmation ABC. La première version est publiée pour la première fois en 1991 sous le nom 0.9.0.

Python 2.0 en 2000 a introduit de nouvelles fonctionnalités telles que :

- La compréhension des listes.
- La collecte des ordures à détection de cycle.
- Le comptage des références.
- La prise en charge d'Unicode.

Les différents détails sur python dans l'annexe B.

## III.4-Les bibliothèques utilisées

### III.4.1-Tensorflow

Tensorflow est une plateforme open source de bout en bout dédié au Machine Learning et Deep Learning. Elle propose un écosystème complet et flexible d'outils, de bibliothèques et de ressources communautaires permettant aux chercheurs d'avancer dans le domaine du ML et DL. Et aux développeurs de créer et déployer facilement des applications qui exploitent cette technologie [40].

### III.4.2-Keras

Keras est une API de Deep Learning écrite en Python, s'exécutant sur la plateforme d'apprentissage automatique Tensorflow. Il a été développé en mettant l'accent sur la

possibilité d'une expérimentation rapide. Être capable de passer de l'idée au résultat le plus rapidement possible, est la clé pour faire de bonnes recherches.

Keras est :

- Simple.
- Flexible.
- Powerful.

Keras est l'API de haut niveau de TensorFlow 2 : une interface accessible et hautement productive pour résoudre les problèmes de Machine Learning (ML). Il fournit des abstractions et des blocs de construction essentiels pour le développement et l'expédition de solutions de Machine Learning (ML) avec une vitesse d'itération élevée.

Keras permet aux ingénieurs et aux chercheurs de tirer pleinement parti de l'évolutivité et des capacités multi plateformes de TensorFlow, nous pouvons exécuter keras sur TPU ou sur GPU, et nous pouvons exporter nos modèles keras pour les exécuter dans un navigateur ou sur un appareil mobile [41].

### III.4.3-Sklearn

Scikit-learn (Sklearn) est la bibliothèque la plus utile et la plus robuste pour la Machine Learning (Apprentissage automatique) en python. Il donne une sélection d'outils efficaces pour l'apprentissage automatique (Machine Learning) et la modélisation statistique, y compris la classification, la régression, le clustering et la réduction de la dimensionnalité via une interface de cohérence en python. Cette bibliothèque, qui est en grande partie écrite en python, est construite su Numpy, Scipy et Matplotlib [42].

### III.4.4-Matplotlib

Matplotlib est l'un des paquets Python les plus populaires utilisés pour la visualisation des données. Il s'agit d'une bibliothèque multi plateforme permettant de créer des tracés 2D à partir de données classées dans des tableaux. Il donne une API od'objet orientée qui aide à intégrer des tracés dans des applications à l'aide de boites à outils Python GUI telles que :

- PyQt.
- WxPthonotTkinter.

Il peut également être utilisé dans :

- Les shells Python.

- Les shells IPython.
- Les ordinateurs portables Jupyter.
- Les serveurs d'applications web [43].

## III.6-Prétraitement des Signaux ECG

Dans cette partie nous avons pris des signaux électriques ECG de différentes maladies cardiaques de notre base de données selon les classes qu'on a choisi (4 classes), chaque fichier a une durée de $10s$ avec une fréquence d'échantillonnage de $360Hz$, ce qui signifie que chaque fichier contient $3600$ échantillons. Nous avons traité ces fichiers selon les étapes de l'organigramme de la figure III-1.

Nous avons pris en charge 654 signaux avant de segmenter.

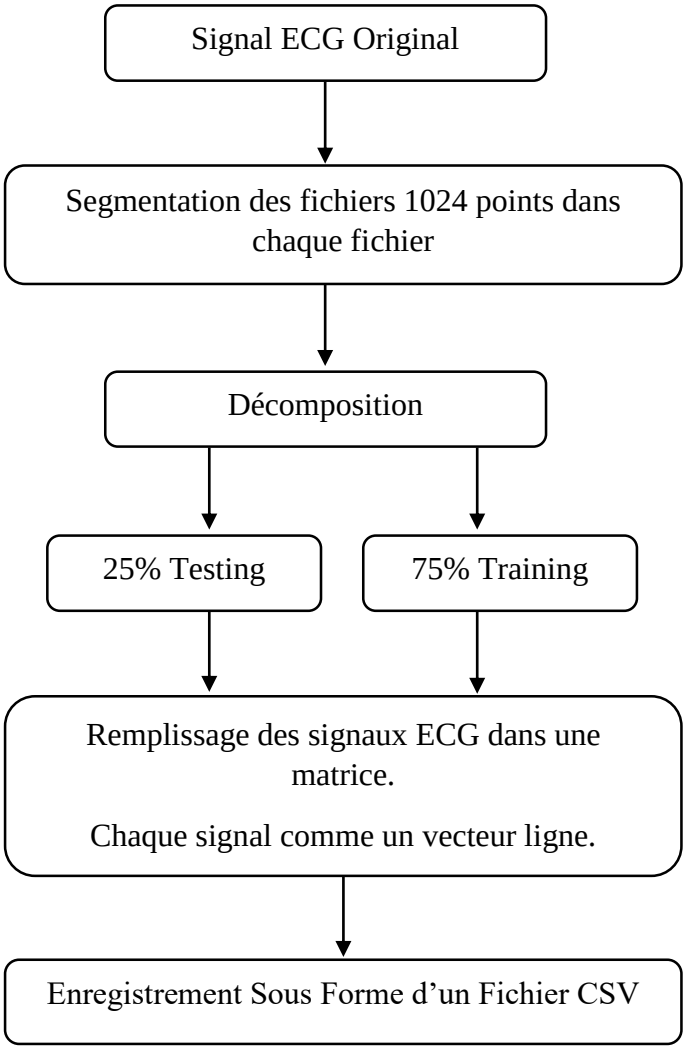



**Figure III-1 :** Organigramme de Traitement des données

### III.6.1-Segmentation des fichiers

Chaque fichier du signal ECG contient 3600 points, pour cela nous avons divisé les signaux ECG en différents segments de 1024 points dans chacun comme le montre la figure III-2.

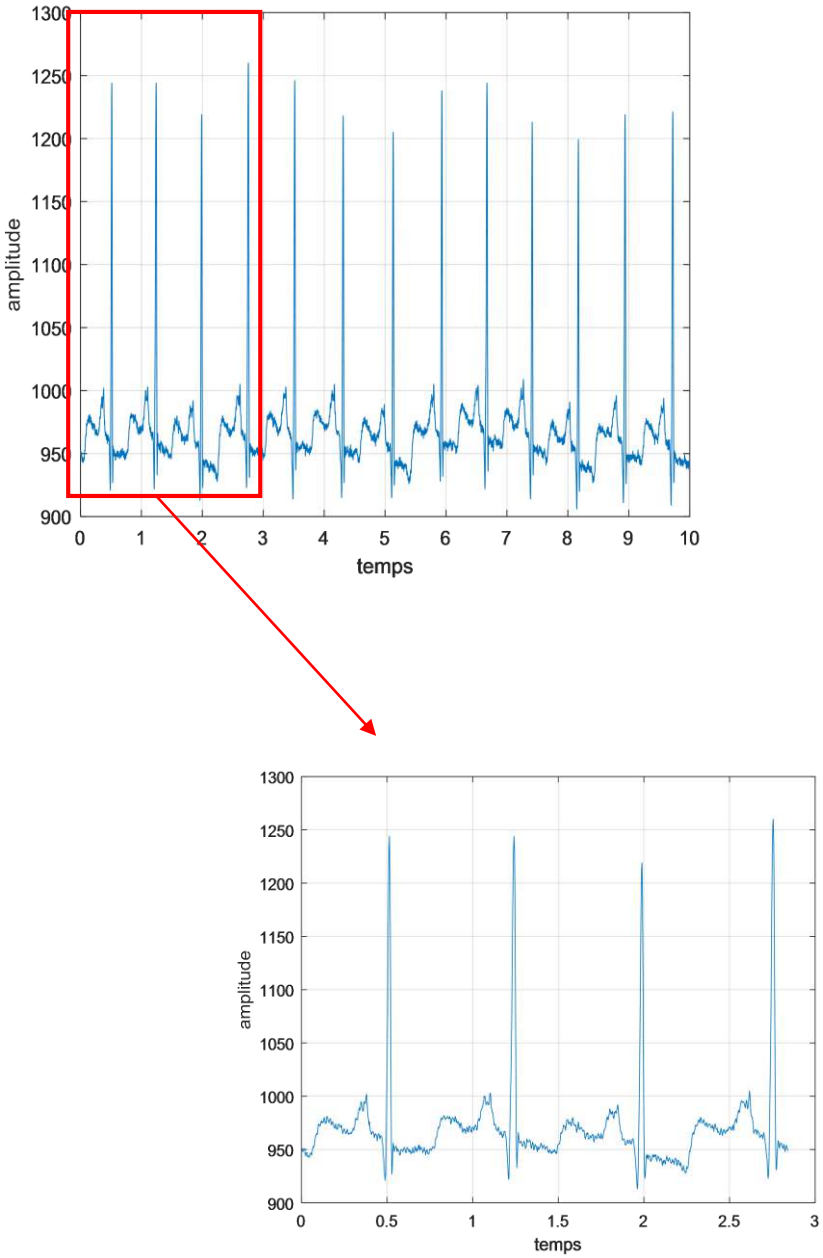



**Figure III-2 :** Segmentation des fichiers

La segmentation du signal ECG consiste à prendre des segments qui contiennent au minimum une période ou plus. Une période contient :

- Onde P.
- Intervalle PR.
- Segment PR.
- Complexe QRS.
- Intervalle QT.
- Onde T.
- Segment ST.

Notre segmentation consiste à décomposer chaque signal en plusieurs signaux de longueurs 1024 points. Cette méthode permet de localiser les différentes caractéristiques d'ECG. Nous avons utilisé cette méthode de segmentation pour augmenter notre base de données.

### III.6.2-Décomposition

Dans cette étape les signaux ECG sont divisés en deux parties (voir figure III-3):

- Partie 1 : 75% des signaux ECG originaux pour la partie d'entrainement du réseaux neurones.
- Partie 2 : Les 25% des signaux ECG qui reste seront réservé pour la phase validation et test du réseau neurone, dont 15% pour la Validation et 10% pour le test final.

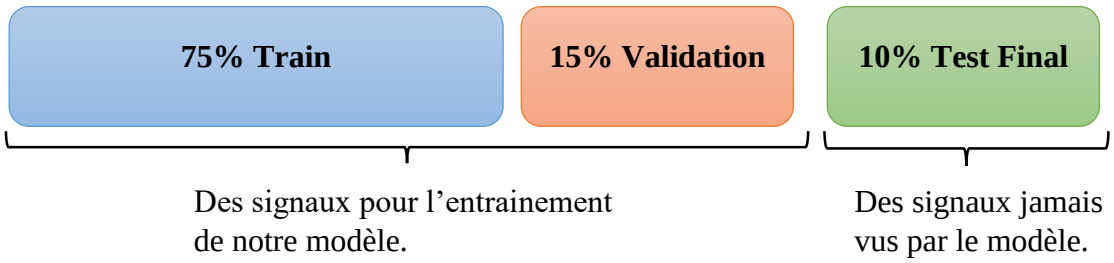



**Figure III-3 :** Décomposition des fichiers

### III.6.3-Remplissage des signaux

Après la segmentation et le prétraitement que nous avons exécuté, on procède à la prochaine étape qui sera comme suit : prendre chaque signal ECG comme un vecteur ligne et placer les différents signaux dans une matrice avec un ordre bien défini (voir figure III-4**)**.

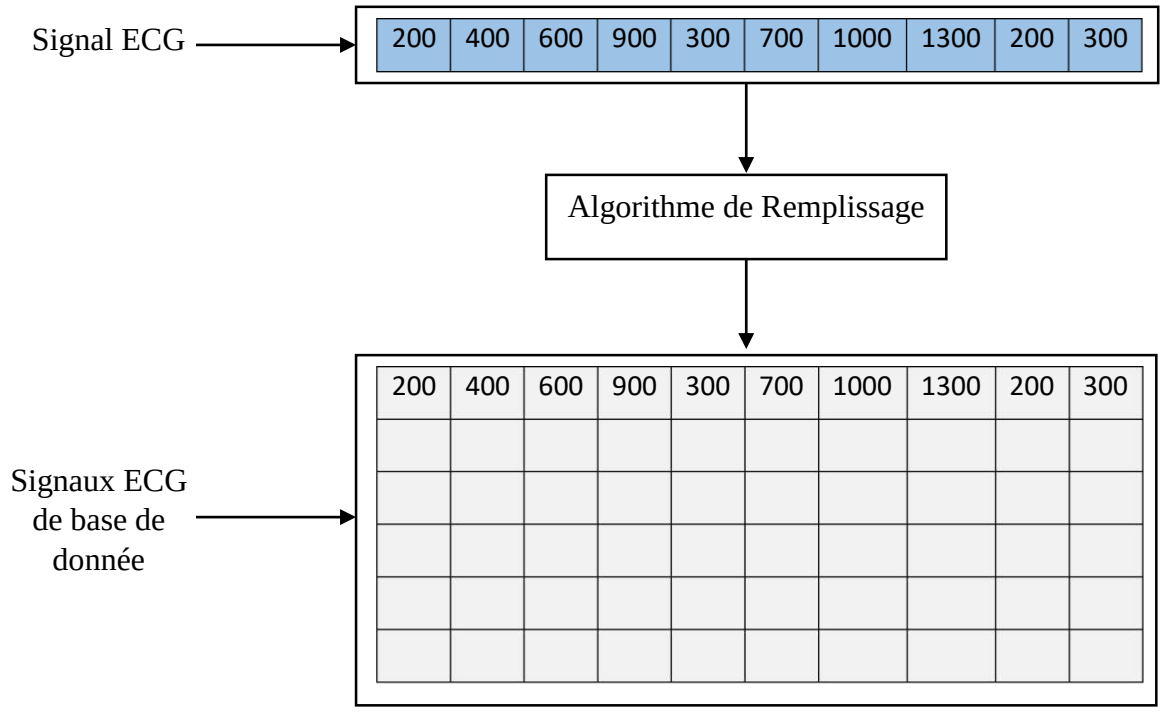



**Figure III-4 :** Remplissage des signaux.

### III.6.4-Enregistrement

Après avoir effectué les différentes étapes précédentes, nous avons enregistré les signaux ECG dans un fichier CSV (voir figure III-5).

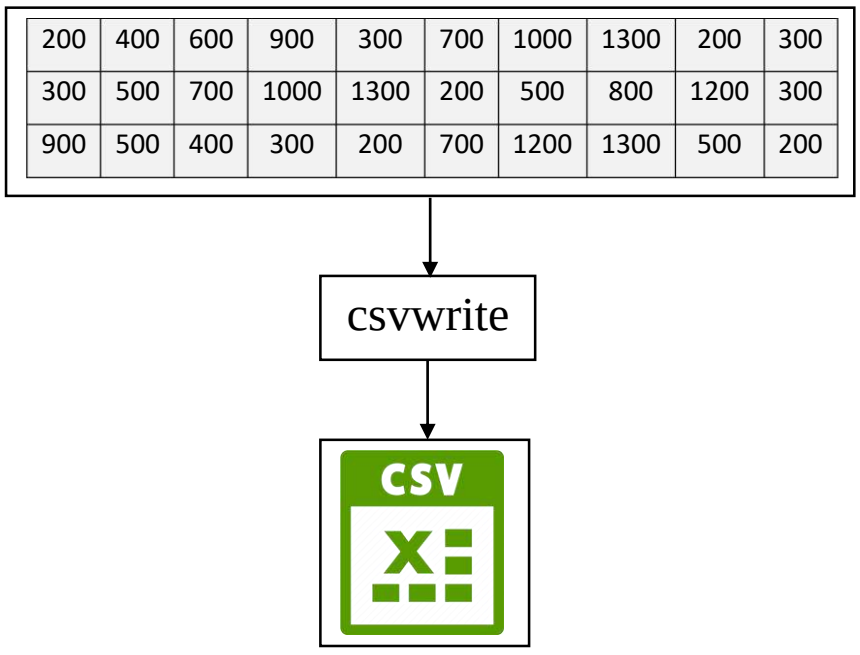



**Figure III-5 :** Enregistrement sous forme csv

### III.6.5-Normalisation

La Normalisation est un processus d'organisation d'une base de données pour réduire la redondance et améliorer l'intégrité des données.

La normalisation est une étape importante de la conception des bases de données, car elle contribue à améliorer :

- La vitesse d'exécution.
- La précision.
- L'efficacité.

Dans cette partie nous avons utilisé la commande **rescale** à l'aide du Matlab comme le montre l'organigramme de la figure III-6 :

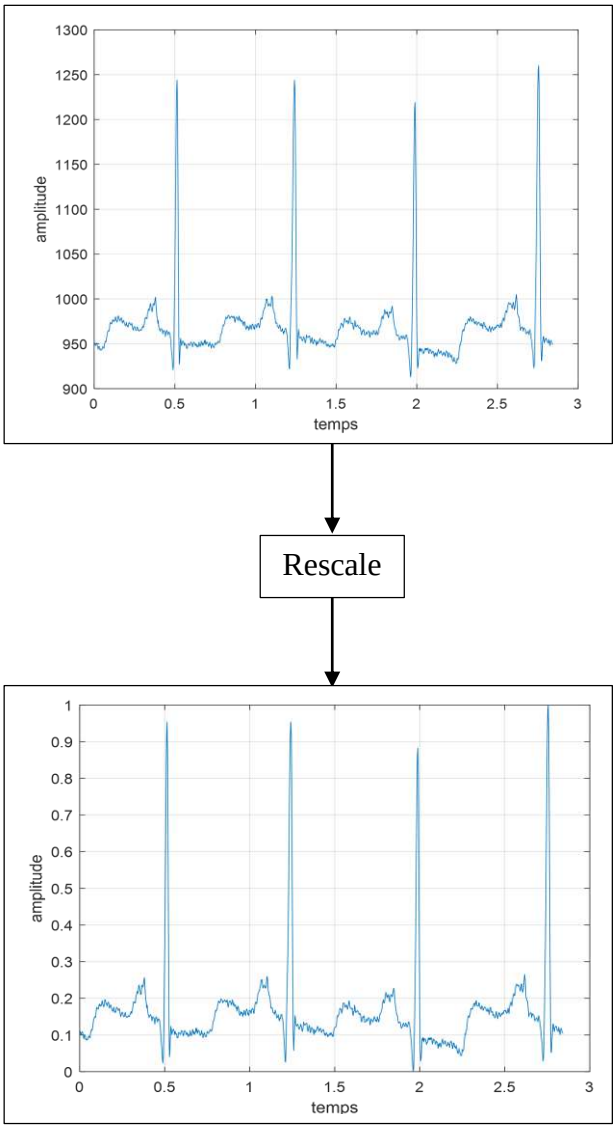



**Figure III-6 :** Normalisation du signal

### III.6.6-Distribution de base de données

La figure III-7 représente la distribution de base de données.

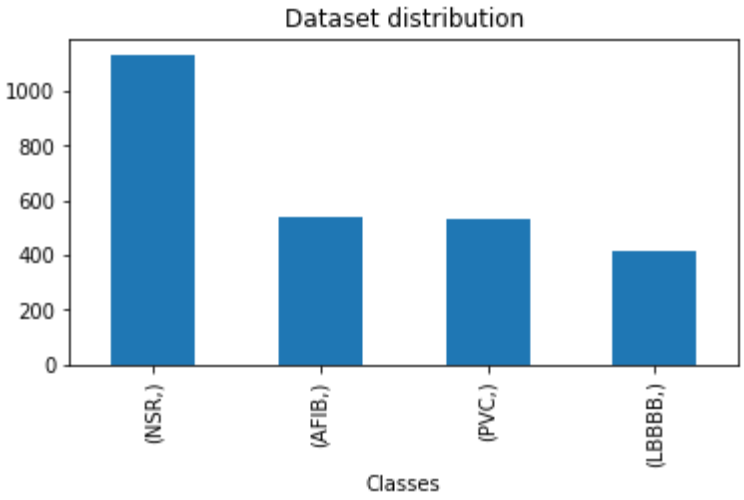



**Figure III-7 :** Distribution la Base de données

Les étapes précédentes du prétraitement nous à permettant de générer 2616 signaux ECG répartis comme suit :

NSR ⟶ Rythme Sinusal Normal ⟶ 1132

AFIB ⟶ Fibrillation Auriculaire ⟶ 540

PVC ⟶ Contraction Ventriculaire Prématuré ⟶ 532

LBBB ⟶ Battement du Bloc de Branche Gauche ⟶ 412

Le diagramme de dispersion (Scatter plot) (voir Figure III-8) est largement utilisé pour tracer des points de données sur un axe horizontal et un axe vertical dans le but de montrer à quel point une variable est affectée par une autre.

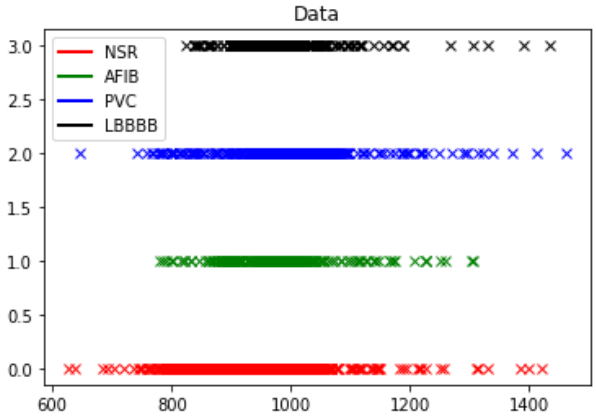



**Figure III-8 :** Scatter plot du Base de données

## III.7-Modèle CNN

Nous avons créé plusieurs architectures (modèle) CNN afin de choisir celle qui offre les meilleurs caractéristiques (précision, nombre de paramètres …). Figure III-9 représente les différentes étapes à suivre pour entrainer le réseau CNN.

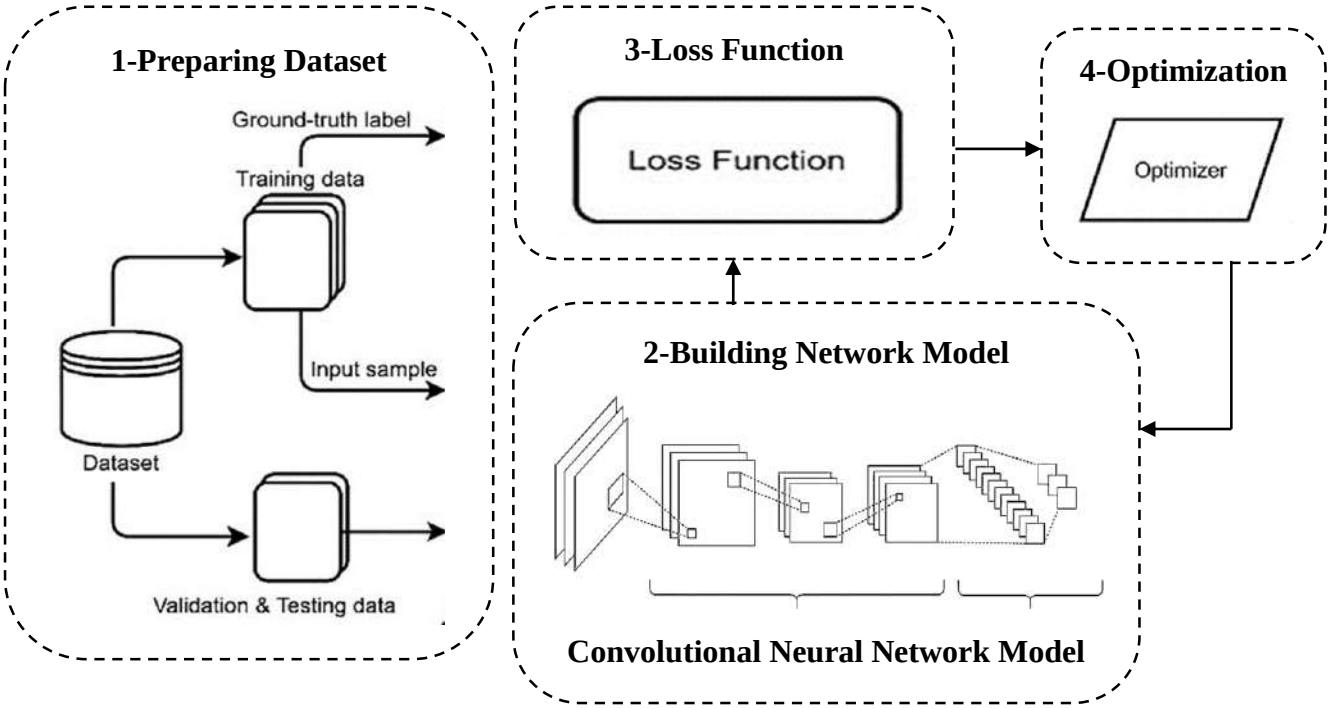



**Figure III-9 :** Diagramme schématique de procédure d'entrainement des réseaux CNN

### III.7.1-Architecture utilisée

Après avoir fait plusieurs essais, nous avons pu élaborer les trois meilleures architectures afin d'avoir de bons résultats. (Tableau III-2**).** Pour améliorer l'efficacité du traitement, on intercale entre les couches de traitements une fonction d'activation « ReLu (Rectified Linear Unit) » sur les signaux des sorties.

| | **Architecture 1** | | **Architecture 2** | | **Architecture 3** | |
|---|---|---|---|---|---|---|
| | Conv1D (32,5) | | Conv1D (32,3) Padding | | Conv1D (32,3) Padding | |
| | BatchNormalization | | BatchNormalization | | BatchNormalization | |
| | AveragePooling1D(3) | | AveragePooling1D(2) | | MaxPooling1D(2) | |
| | Conv1D (64,3) | | Conv1D (64,3) Padding | | Conv1D (64,3) Padding | |
| | AveragePooling1D(2) | | AveragePooling1D(2) | | MaxPooling1D(2) | |
| | Conv1D (64,3) | | Conv1D (64,3) Padding | | Conv1D (64,3) | |
| | AveragePooling1D(2) | | AveragePooling1D(2) | | AveragePooling1D(2) | |
| | Conv1D (64,3) | | Conv1D (64,3) | | Conv1D (64,3) | |
| | AveragePooling1D(2) | | AveragePooling1D(2) | | AveragePooling1D(2) | |
| | Conv1D (128,3) | | Conv1D (128,3) | | Conv1D (128,3) | |
| | AveragePooling1D(2) | | AveragePooling1D(2) | | AveragePooling1D(2) | |
| | Conv1D (128,3) | | Conv1D (128,3) | | Conv1D (128,3) | |
| | AveragePooling1D(2) | | AveragePooling1D(2) | | AveragePooling1D(2) | |
| | Conv1D (256,3) | | Conv1D (256,3) | | Conv1D (256,3) | |
| | AveragePooling1D(2) | | AveragePooling1D(2) | | AveragePooling1D(2) | |
| | Flatten | | Conv1D (512,3) | | Conv1D (512,3) | |
| | Dense(100) | | AveragePooling1D(2) | | AveragePooling1D(2) | |
| | Dense(64) | | Flatten | | Flatten | |
| | Dense(4) | | Dense(100) | | Dense(100) | |
| | | | Dense(10) | | Dense(32) | |
| | | | Dropout(0.3) | | Dropout(0.3) | |
| | | | Dense (4) | | Dense(4) | |
| Entrainement | **Précision** | 98% | **Précision** | 99% | **Précision** | 97% |
| | **Perte** | 0.2 | **Perte** | 0.1 | **Perte** | 0.11 |
| Validation | **Précision** | 87% | **Précision** | 89% | **Précision** | 95% |
| | **Perte** | 1.2 | **Perte** | 0.6 | **Perte** | 0.15 |

**Tableau III-2 :** Les meilleurs Architectures élaborées

Le tableau III-2 montre les trois différentes architectures qu'on a obtenu dans notre recherche, pour la meilleure architecture et le meilleur modèle, on procède avec le même nombre d'époques (de 50 époques). Les résultats obtenus sont exprimés en termes de la précision et les pertes d'entraînement et de validation**.**

La première architecture que nous présentons dans le Tableau III-2 est composé de :

- 7 couches de convolution.
- 7 couches de AveragePooling.
- Une couche de BatchNormalization.
- 3 couches de connexion entière (fully connected layer).

Notre signal de 1024 points passe par la première couche de convolution. Cette couche est composée par 32 filtres de longueur 5, chaque couche de convolution est suivie par une fonction d'activation ReLu (prendre les valeurs positives seulement), alors que la sortie de cette couche (features maps) est de taille 1020×32. Cette sortie est injectée à l'entrée de la couche de BatchNormalization, ensuite on applique AveragePooling1D pour réduire la taille du signal. La sortie de cette couche, est de taille 340×32. Cette dernière et à son tour été injectée comme entré dans la deuxième couche de convolution qui se compose par 64 filtres de longueur 3, cette couche de convolution est suivie par une fonction d'activation ReLu, la sortie est de taille 340×64. Ensuite, on applique AveragePooling1D pour réduire la taille de cette sortie à 169×64. On procède de la même manière pour la troisième et quatrième couche de convolution et de AveragePooling1D. De même, pour la cinquième et la sixième couche de convolution qui sont composées de 128 filtres de longueur (taille) 3, suivie par une fonction d'activation ReLu et une couche d'AveragePooling1D. La sortie de ces deux couches passe par la septième couche de convolution qui est elle aussi composée par 256 filtres de longueur (taille) 3, suivie par l'application d'une fonction d'activation ReLu, ensuite une couche d'AveragePooling1D. La sortie de ces couches est de taille 3×256.

Après les 7 couches de convolution, la couche de BatchNormalization et les 7 couches de l'AveragePooling1D, nous utilisons trois couches de connexion entière. La première couche de 100 neurones avec une fonction d'activation ReLu, suivie par une deuxième couche de 64 neurones avec la même fonction d'activation ReLu, et la dernière couche est un softmax qui permet de calculer la probabilité des 4 classes (nombre de classe dans notre base de données).

Cette architecture donne comme résultats une précision de 98% pour l'entrainement, et 87% pour la validation. Pour les pertes on a obtenu 0.2 pour l'entrainement, et 1.2 pour la validation.

Pour améliorer la précision et minimiser les pertes nous avons effectués quelques changements au niveau d'architecture. D'abord un premier changement dans la première couche de convolution qui est composée par 32 filtres mais de longueur (taille) 3, suivie par une fonction d'activation ReLu. Ensuite l'application d'AveragePooling1D pour réduire la taille du signal en 2 après la couche de BatchNormalization. Nous avons ajouté le « Padding » dans les trois premières couches de convolution.

Cependant une huitième couche de convolution composée par 512 filtres de longueur (taille) 3 fut ajoutée, suivie par une fonction d'activation ReLu alors la sortie sera de taille 4×512. Enfin l'application de l'AveragePooling1D sur cette sortie nous donne une sortie de cette couche de taille 2×512. Pour la couche de connexion entière nous avons remplacé la couche de 64 neurones avec une couche de 10 neurones, puis une couche de Dropout.

Cette architecture donne des résultats de précision (entrainement 99%, validation 89%) et des pertes (entrainement 0.1, validation 0.6) alors on remarque que la précision a augmenté et les pertes a diminué.

La troisième architecture, nous avons exécuté des petits changements sur l'architecture précédente. En premier lieu, nous gardons le « Padding » pour les deux premières couches de convolution seulement. Nous avons remplacé les deux premières couches de l'AveragePooling1D avec les couches de MaxPooling1D.

Concernant la couche de connexion entière, nous avons gardé la couche de 100 neurones suivie par une fonction d'activation ReLu et remplacé la couche de 10 neurones avec une couche de 32 neurones suivie aussi par une fonction d'activation ReLu.

Cette architecture donne un meilleur résultat de précision (entrainement 97%, validation 95%) et des pertes (entrainement 0.11, validation 0.15). Ainsi, on en conclut que la troisième architecture est la meilleure.

Après plusieurs essais et changements au niveau de l'architecture, nous avons conclu que cette architecture (voir figure III-10) est l'architecture la plus précise car elle donne des meilleurs résultats comparés aux autres architectures. On a un nombre total des paramètres : 703304 paramètres.

Nous avons choisi cette architecture car notre entrée (donnée (signaux ECG)) est un vecteur ligne de longueur 1024 points (une ligne et 1024 colonnes).

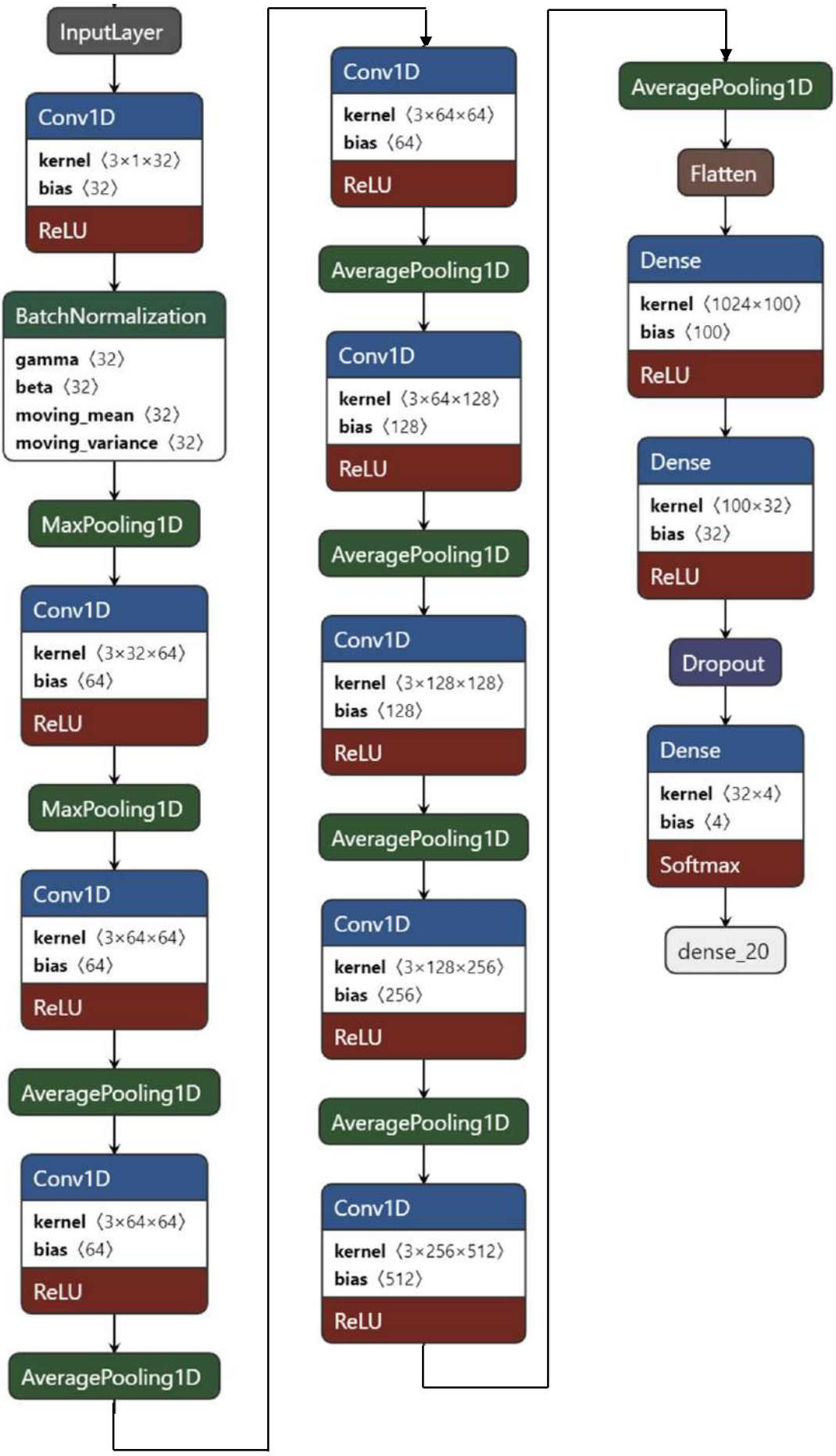



**Figure III-10 :** L'architecture utilisée

### III.7.2-Méthode utilisée pour l'optimisation

Nous avons utilisé plusieurs (nombreuses) méthodes pour avoir un très bon résultat de performance de classification et pour sauvegarder le meilleur modèle. Ces méthodes sont :

#### III.7.2.1-KFold Cross Validation

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

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

La figure III-11 représente un schéma explicatif sur cette méthode.

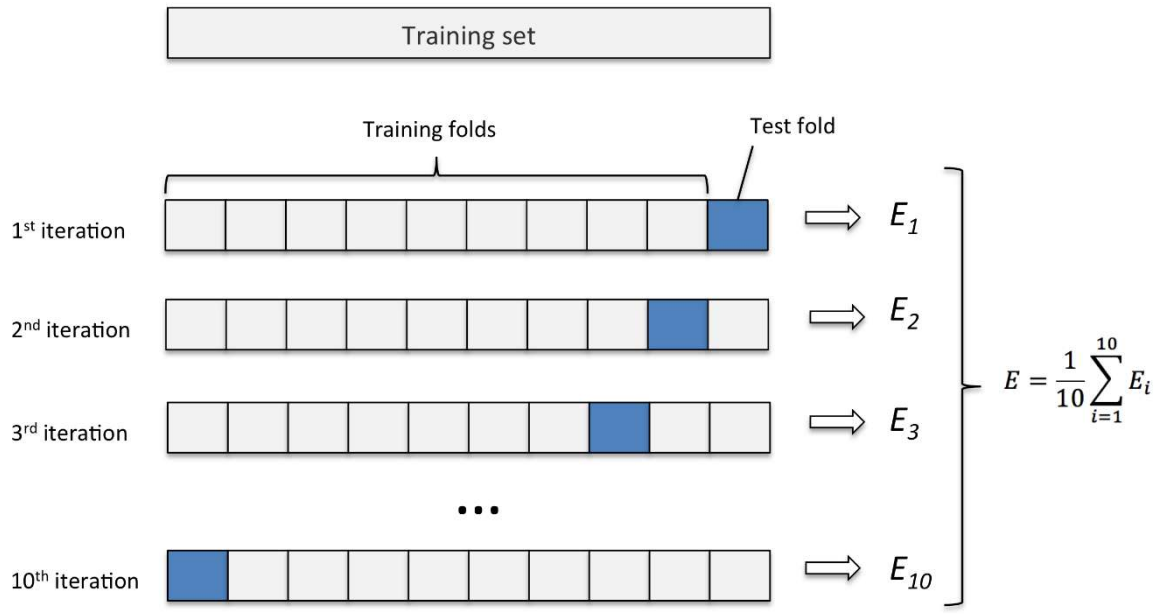



**Figure III-11:** K-Fold Cross Validation

### III.7.2.2-Callbacks Model Check Point

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

Nous avons utilisé cette fonction pour sauvegarder le modèle le plus précis et en comparant avec la précision de la prochaine époque, s'il augmente nous pouvons prendre ce modèle et écrasé l'ancien modèle pour garder le meilleur modèle [44].

### III.7.2.3-Compiler

Avant de lancer l'entrainement, nous devons configurer les processus suivants :

- **Loss :** C'est une fonction utilisée par le modèle afin de minimiser les erreurs. Nous utilisons le « categorical_crossentropy », car nous avons une classification multi classe.
- **Optimizer :** Nous avons utilisé l'optimizer stochastique « Adam », car il donne de très bons résultats.
- **Metrics :** Dans le cas d'un problème de classification, nous utilisons metrics=['accuracy'].

### III.7.3-Les Résultats obtenus

Afin de montrer les résultats obtenus par notre modèle CNN 1D, nous avons illustré les différents résultats en termes de :

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

Nous avons pris un nombre d'époques égal à 50 époques.

### III.7.3.1-Graphe de précision et des pertes

La figure III-12 représente les graphes de : précision et perte d'entrainement et de validation en fonction du nombre d'époques, on remarque que la précision pour la validation augmente proportionnellement avec le nombre des époques ainsi que l'entrainement. D'autre part la perte marque une diminution inversement proportionnelle avec le nombre d'époques.

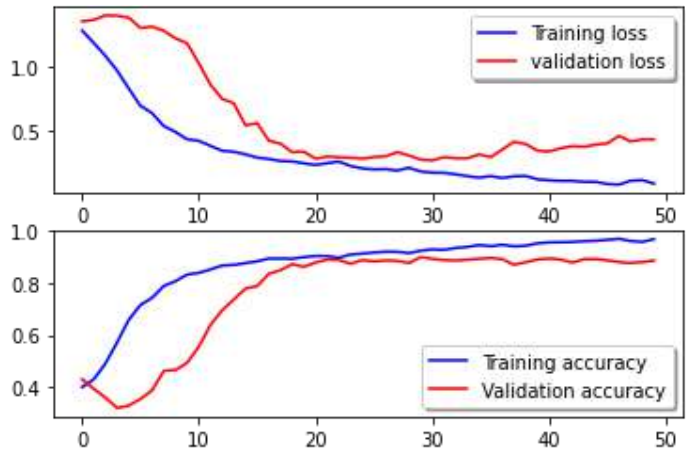



**Figure III-12 :** Graphe de précision et des pertes

### III.7.3.2-Rapport de classification

Cette figure (figure III-13) illustre le rapport de classification, qui représente l'évaluation des performances du modèle de classification (apprentissage en profondeur). Il affiche la précision, recall, f1-score1 et le support de notre modèle. Ainsi, il nous permet de mieux comprendre les performances globales de notre modèle.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.98 | 0.90 | 0.94 | 94 |
| 1 | 0.98 | 1.00 | 0.99 | 44 |
| 2 | 0.83 | 0.95 | 0.89 | 41 |
| 3 | 1.00 | 1.00 | 1.00 | 30 |
| accuracy | | | 0.95 | 209 |
| macro avg | 0.95 | 0.96 | 0.95 | 209 |
| weighted avg | 0.95 | 0.95 | 0.95 | 209 |

**Figure III-13 :** Rapport de Classification

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

$$F1 - Score = \frac{Precision + Recall}{2}$$

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

D'après l'analyse du rapport de classification, nous avons remarqué que le taux de précision est de 83% avec un recall de 95% et f1-score de 89% pour la troisième classe PVC, ce n'est pas un bon résultat par rapport aux autres classes car les signaux de cette classe sont similaires à celle de la première classe NSR.

### III.7.3.3-Matrice de Confusion

La Figure III-14 représente la matrice confusion, elle résume les résultats de prédictions et permet d'évaluer les performances de notre modèle de classification. Chaque ligne correspond à une classe réelle, chaque colonne correspond à une classe estimée. Nous avons 4 classes d'où notre matrice de confusion est de 4×4.

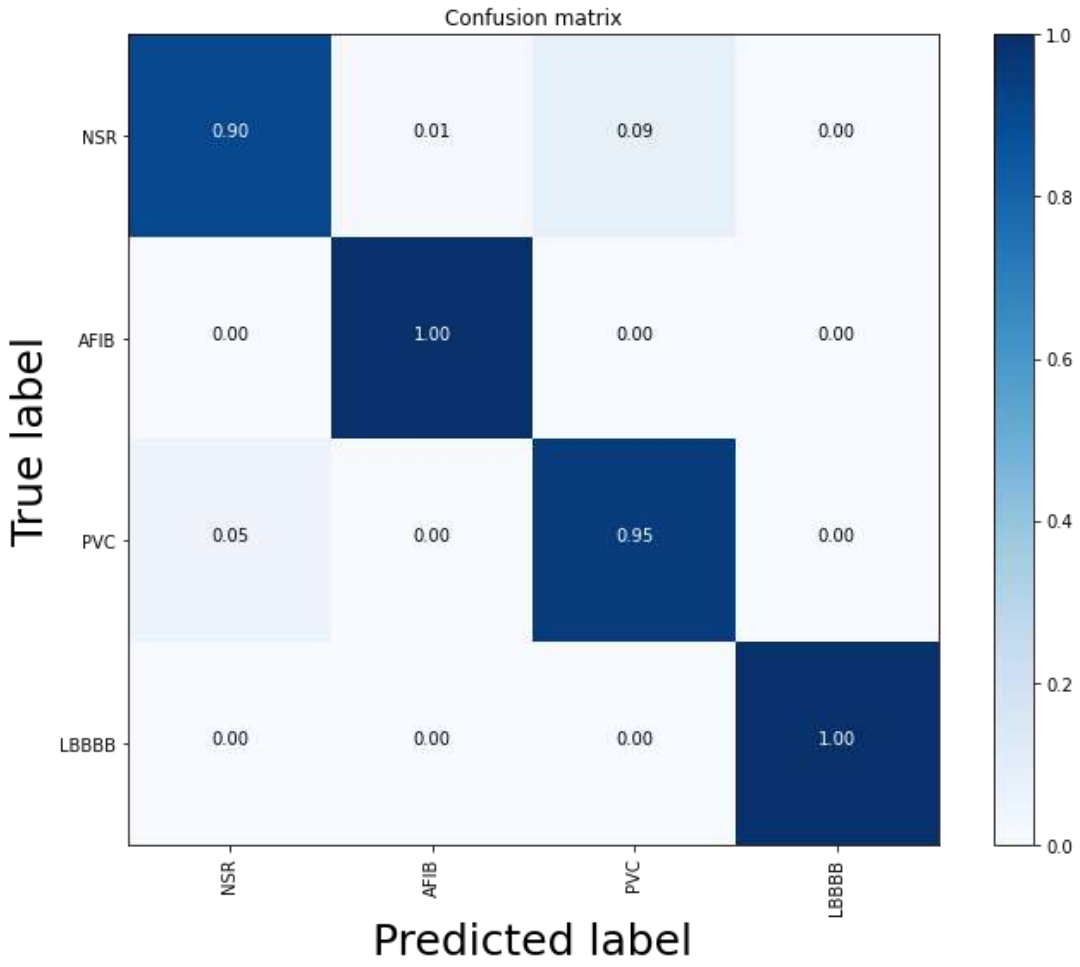



**Figure III-14 :** Matrice de confusion

D'après l'analyse de cette matrice, nous avons vu que dans la première classe NSR est de 90% classé normal mais l'autre 10% est mal classé, il est classé comme une PVC car les signaux sont similaires (voir figure III-15).

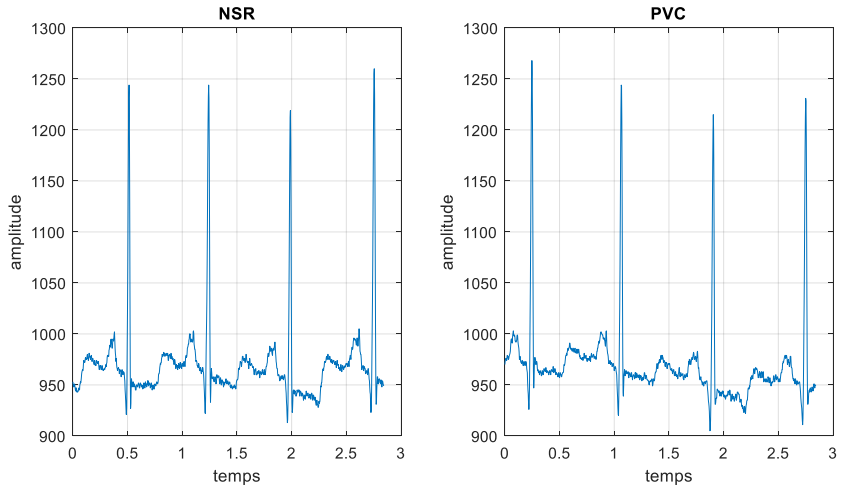



**Figure III-15 :** Signaux NSR et PVC

Concernant la classe PVC à 5% classé ailleurs comme une NSR (1ère classe) car les signaux sont similaires.

Les résultats du meilleur modèle qui donne une grande précision sont mentionnés dans le tableau ci-dessous (Tableau III-3) :

| | **Précision** | **Perte** |
|---|---|---|
| **Entrainement** | 97% | 0.11 |
| **Validation** | 95% | 0.15 |

**Tableau III-3 :** Meilleur Résultat du réseau

## III.8-Comparaison

Le tableau III-4 et la figure III-16 illustrent les meilleurs résultats avec les différentes méthodes utilisées :

| | **Avec KFold** | **Sans K-Fold** | **Avec Normalisation et K-Fold** |
|---|---|---|---|
| **Précision d'entrainement** | 97% | 92.9% | 94.8% |
| **Perte d'entrainement** | 0.11 | 0.17 | 0.11 |
| **Précision de validation** | 95% | 91.4% | 92.3% |
| **Perte de validation** | 0.15 | 0.21 | 0.22 |

**Tableau III-4 :** Les meilleurs Résultats de cette architecture

Nous avons suivi la méthode de K-Fold cross validation d'une part pour optimiser et améliorer la précision du modèle et d'autre part pour éviter et minimiser le bruit.

Par la suite le modèle s'entraine sur tous les cas possibles de chaque classe.

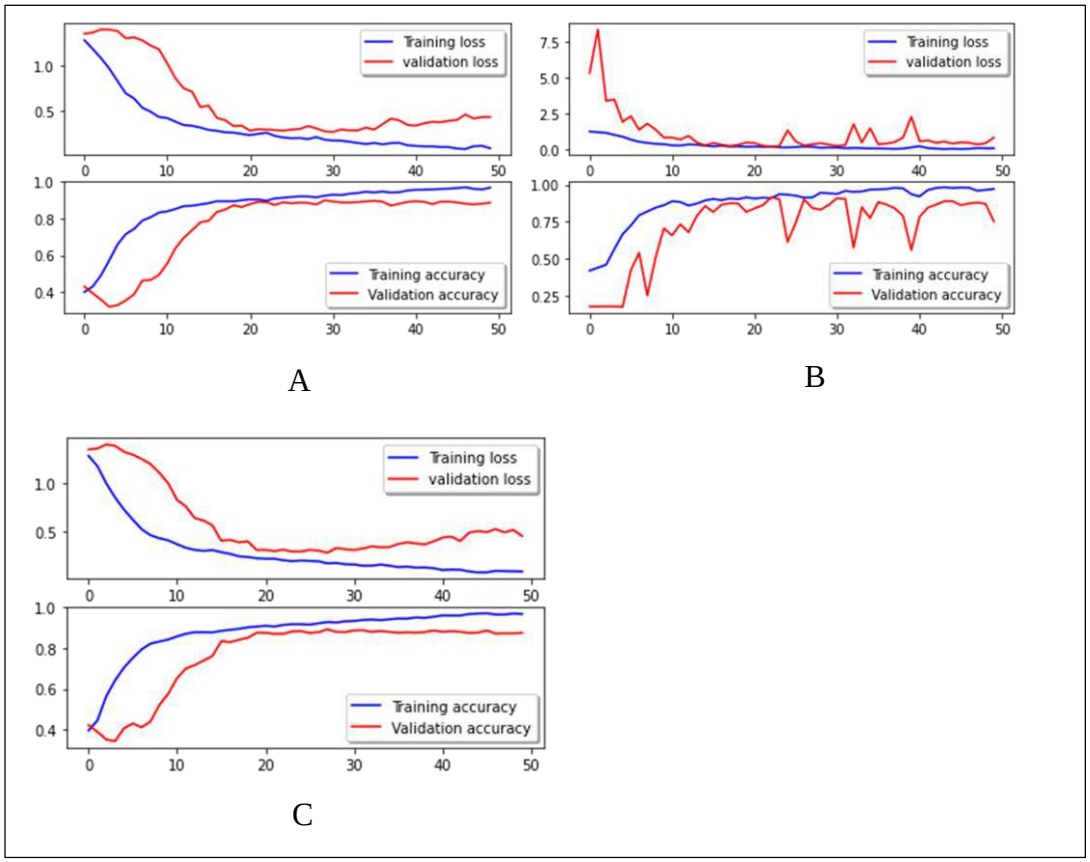



**Figure III-16 :** Les Graphes de précision et perte des différentes méthodes A : Avec K-Fold, B : Sans K-Fold, C : Avec Normalisation et K-Fold

## III.9-Test expérimentale

### III.9.1-Matériels utilisé

#### III.9.1.1-Carte Arduino Uno

Arduino UNO (voir figure III-17) est l'unité principale que nous avons utilisée dans notre système proposé. Elle se compose d'une micro-puce ATmega328P comme un microcontrôleur et de composants complémentaires qui facilitent la programmation et l'interfaçage avec d'autres circuits. Cette carte peut se programmer avec logiciel Arduino IDE.




**Figure III-17 :** Arduino Uno

### III.9.1.2-Capteur ECG AD8232

Le capteur ECG AD 8232 est un module économique utilisé pour mesurer les activités électriques produites par le cœur. En raison de la nature extrêmement bruyante des signaux ECG produits par le cœur. AD 8232, il est conçu pour l'extraction et l'amplification des petits signaux en présence de bruit de fond. Les principales sources de ces bruits sont des activités musculaires et des artefacts de mouvement produits par le corps du patient [45]. La figure III-18 montre un module de capteur qui n'est pas nocif et qui mesure la fréquence cardiaque et donne le signal qui peut être directement connecté à n'importe quel contrôleur comme MSP430, Arduino, Raspberry-pi. Cette conception aide à obtenir le signal de sortie rapidement à partir d'un contrôleur intégré ou d'un convertisseur analogique-numérique (ADC) dispositif ultrabasse consommation [46].

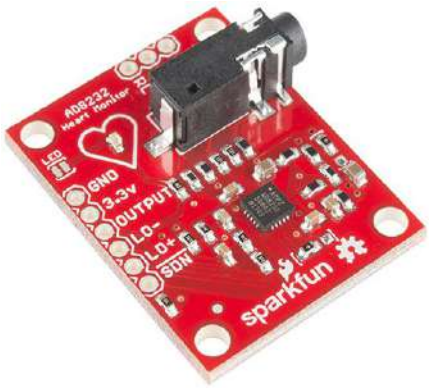



**Figure III-18 :** Capteur ECG AD8232

### III.9.1.3-Les Electrodes

Les électrodes utilisées dans ce projet sont des électrodes jetables (voir figure III-19) qui offrent une alternative rapide, facile et plus hygiénique aux électrodes réutilisables ou aux électrodes humides. Ces électrodes sont pré-gélifiées et sont appelées électrodes sèches. Au lieu

d'utiliser un système d'électrodes à 12 dérivations et de conserver l'appareil en vrac, il a été utilisé un système d'électrodes à 3 dérivations où les électrodes sont placées sur les poignets comme le montre le tableau III-6 [46].

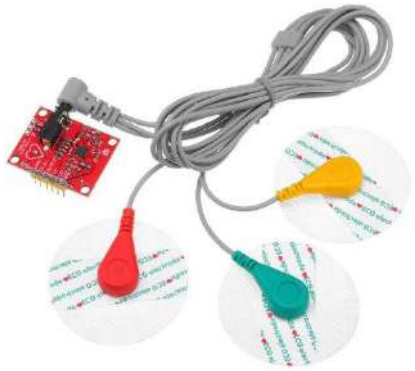


**Figure III-19 :** Les électrodes utilisé

### III.9.2-Logiciels utilisés

#### III.9.2.1-Arduino IDE

Arduino IDE est un logiciel open source qui est principalement utilisé pour écrire et compiler le code dans un module Arduino. Il s'agit d'un logiciel Arduino officiel, rendant la compilation de code trop facile pour que même une personne ordinaire sans connaissances techniques préalables puisse se familiariser avec le processus d'apprentissage. Il est facilement disponible pour les systèmes d'exploitation tels que Mac, Windows, Linux.

Le code principal, connu sous le nom croquis, créé sur la plateforme IDE générera finalement un fichier Hex qui est ensuite transféré et téléchargé dans la carte.

L'environnement IDE contient principalement deux parties de base :

- Editeur : est utilisé pour écrire le code requis
- Compilateur : est utilisé pour compiler et télécharger le code dans le module Arduino.

Cet environnement prend en charge les langages C et C++ [47].

### III.9.3-Réalisation du nœud d'acquisition

Arduino Uno implique un convertisseur analogique numérique (ADC) qui réduit la complexité du système. Nous avons réalisé le montage (connexion entre le capteur d'ECG AD8232 et Arduino Uno) présenté dans la f igure III-20

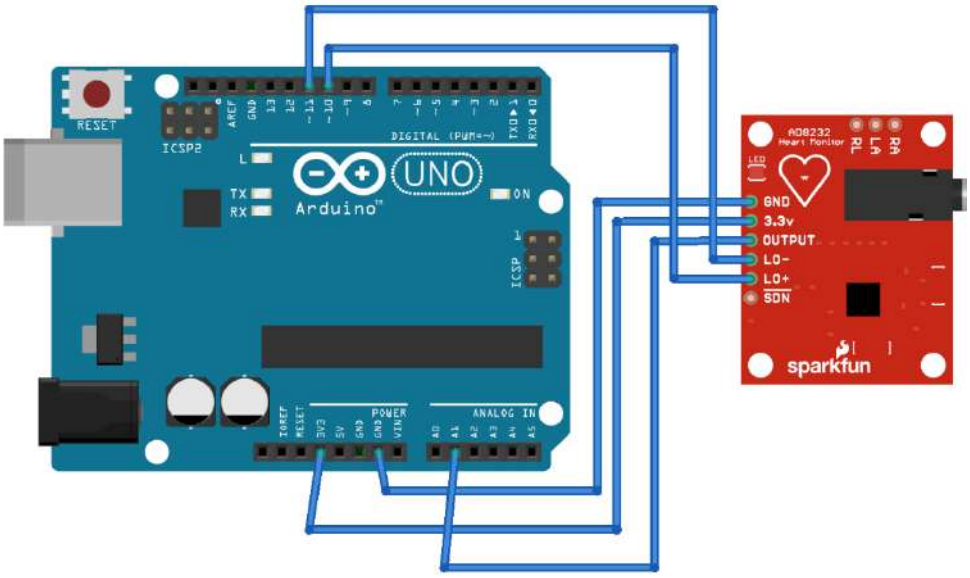



**Figure III-20 :** Réalisation du nœud d'acquisition

Le capteur AD8232 contient 6 branches. Ce capteur est attaché à la carte Arduino Uno à l'aide de fils de connexion. Le détail complet de la connexion avec l'Arduino Uno est donné dans le tableau ci-dessous tableau III-5.

| Fonction | Branche AD8232 | Carte Arduino Uno |
|---|---|---|
| **Power Supply** | 3.3V | 3.3V |
| **Output Signal** | OUTOUT | A0 |
| **Lead Off Detect+** | LO+ | 10 |
| **Lead Off Detect-** | LO- | 11 |
| **Ground** | GND | GND |
| **Shut Down** | SND | Pas de connexion |

**Tableau III-5 :** L'interconnexion entre ECG AD8232 et Arduino Uno

Le placement des électrodes sur le corps humain est représenté dans la figure III-21 et tableau III-6 :

| L'Electrode | Placement |
|---|---|
| Rouge | Bras Droit |
| Jaune | Bras Gauche |
| Vert | Jambe Droite |

**Tableau III-6 :** Placements des électrodes

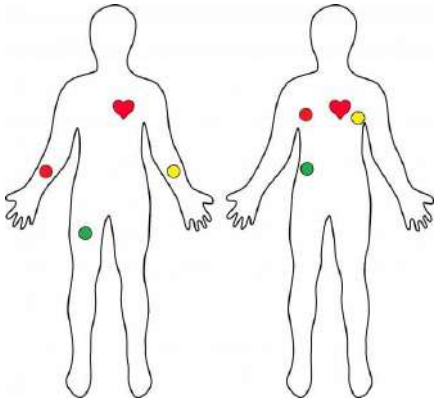


**Figure III-21 :** Placement des électrodes

Une fois le code réalisé et exécuté, nous pouvons avoir les signaux au niveau de traceur série et les valeurs sur le moniteur série.

Le signal mesuré avec AD8232 est présenté dans la figure III-22.

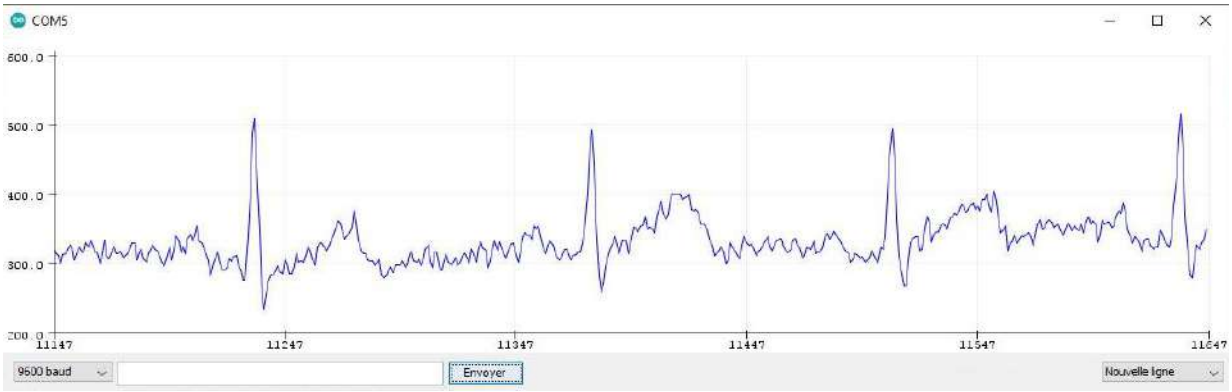



**Figure III-22 :** Signal récolté

Les signaux récoltés et sauvegardés dans cette partie seront pris en charge dans le test final.

### III.9.4-Résultats des tests

Dans cette partie nous utilisons les signaux ECG de test final qui ne sont pas exploités ultérieurement par notre modèle CNN.

Nous utilisons aussi les signaux sauvegardés avec l'AD8232 dans cette partie de test.

La figure III-23 représente les différentes étapes pour tester notre modèle.

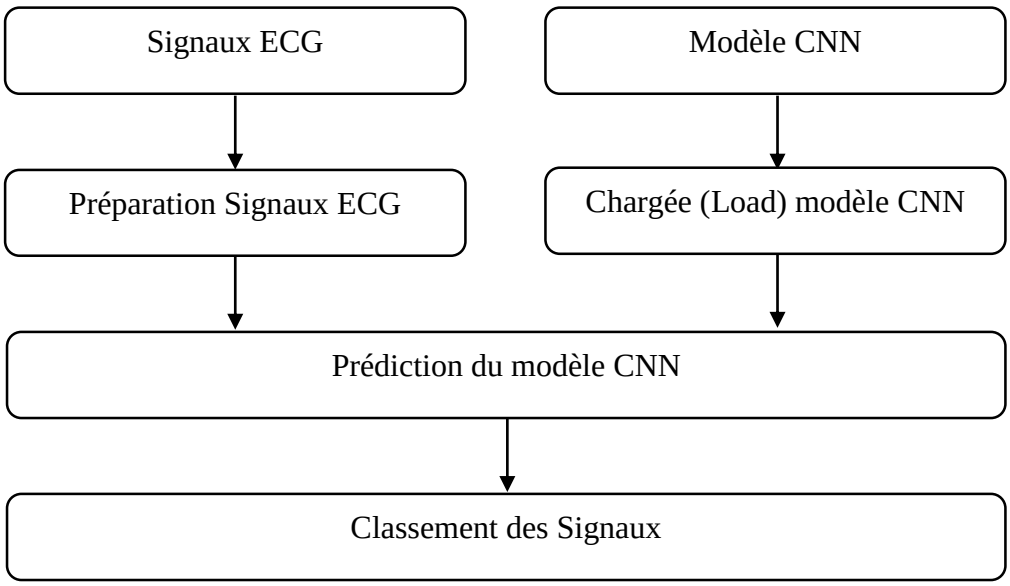



**Figure III-23 :** Organigramme de test expérimental

### III.9.4.1-Chargement du modèle

Pour tester le modèle des résultats précédentes, nous avons utilisé la commande de « **load_model** » pour charger notre modèle.

### III.9.4.2-Préparation des signaux

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

### III.9.4.3-Prédiction et classification des signaux

Après la préparation des signaux et le chargement de notre propre modèle CNN 1D, nous utilisons la commande « **modèle.predict** » pour classifier les signaux . Les résultats de cette prédiction sont donnés dans la figure III-24**.**

```
[4, 0, 0, 0]
[0, 4, 0, 0]
[1, 0, 3, 0]
[0, 0, 0, 4]
```

**Figure III-24 :** Classement des signaux

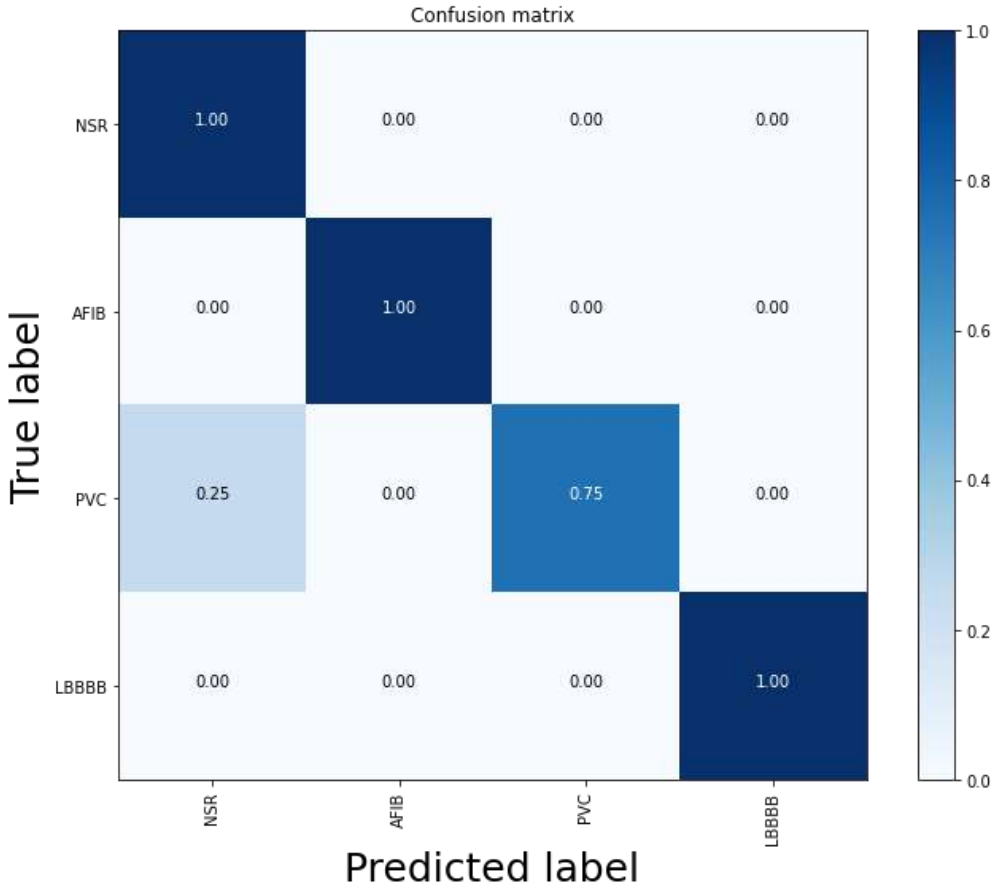



**Figure III-25 :** Matrice de confusion

La figure III-25 représente les résultats de classification des signaux des quatre classes utilisées par des signaux (de test finale) non exploités ultérieurement par notre modèle CNN.

Les signaux utilisés, sont récoltées (sauvegarder) par un capteur d'ECG le AD8232.

D'après cette matrice on remarque qu'il y a un problème dans la classification du signal de la troisième classe PVC qui est classé comme un signal de la première classe car les deux classes sont similaires.

Nous avons estimé la précision et l'erreur absolue de test avec les commandes « **accuracy_score** », « **mean_absolute_error** », les résultats obtenus sont les suivants (voir Tableau III-7) :

| | **Précision** | **Erreur absolue** |
|---|---|---|
| **Résultats** | **0.9375** | **0.125** |

**Tableau III-7 :** Les résultats de test expérimental

## III.10-Implémentation

Les étapes de l'implémentation sont représentées dans l'organigramme de la figure III-26.

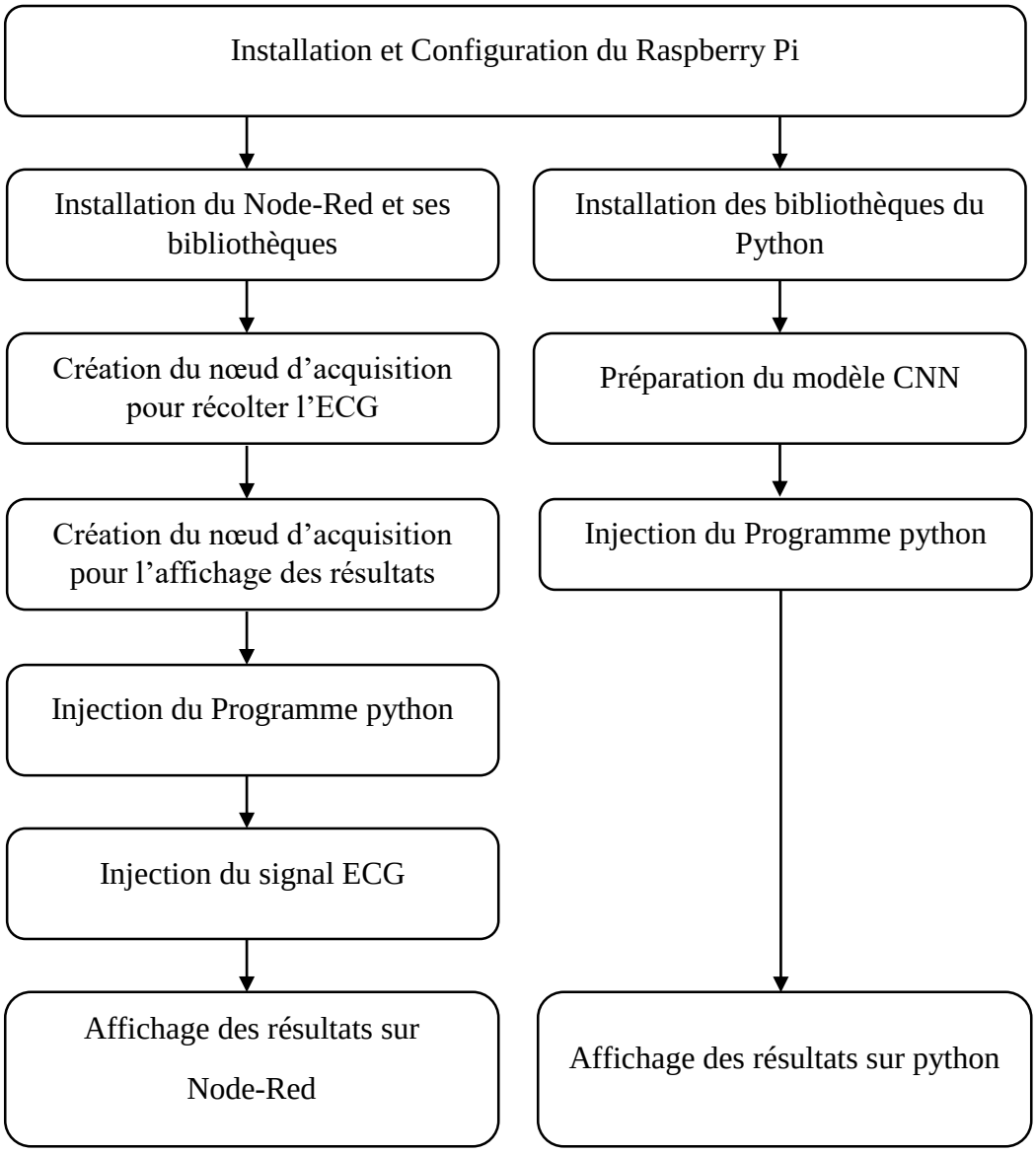



**Figure III-26 :** Organigramme des étapes d'implémentation

### III.10.1-Raspberry Pi

Raspberry Pi est une carte d'ordinateur peu couteuse, petite et portable. Il est peut-être utilisé pour se brancher sur un écran d'ordinateur ou un téléviseur, un clavier, une souris, une clé USB, etc. Raspberry Pi a intégré des logiciels tels que Scratch qui permettent aux utilisateurs de programmer et de concevoir des animations, des jeux ou des vidéos intéressantes. De plus, les programmeurs peuvent également développer des scripts ou des programmes en utilisant le langage Python ; c'est le principal langage de base du système d'exploitation Raspbian.

Raspberry Pi, comme tout autre ordinateur, utilise un système d'exploitation. L'option Linux appelée Raspbian convient parfaitement à Raspberry Pi car elle est gratuite et open source [48].

### III.10.2-Raspberry Pi 3B+

Dans ce travail nous avons opté pour le Raspberry Pi 3 modèle B+ (voir Figure III-27) comme une plateforme embarquée dans la phase des tests pratiques. Il est doté d'un processeur quad core 64 bits fonctionnant à $1.4GHz$, d'un réseau local sans fil bi-bande $2.4GHz$ et $5GHz$, d'un Bluetooth $4.2/BLE$, d'une connexion Ethernet plus rapide et d'une capacité PoE via un HAT PoE séparé [49].



**Figure III-27 :** Raspberry Pi 3B

Les composants de Raspberry Pi 3B+ dans l'annexe C.

Le tableau des caractéristiques de Raspberry Pi 3B+ dans l'annexe D.

### III.10.3-Tensorflow Lite

TensorFlow Lite propose un ensemble d'outils pour la convertir et exécuter des modèles sur des appareils embarqués. Il est utilisé pour convertir les modèles keras déjà formés et enregistrer du format « h5 »au format « tflite » plus léger, adapté au raspberry Pi et à d'autres appareils embarqués. Il est également livré avec une boite à outils pour la quantification post-formation et la formation sensible à la quantification des modèles, avec des possibilités d'accélérer l'interface intégré ou d'assurer une grande précision lors de l'utilisation de modèles quantifiés [50].

### III.10.4-Préparation du modèle

Comme première étape, nous avons converti le modèle CNN keras qui est déjà sauvegardé du format « **.h5** » au format « .**tflite** » pour être adapté au Raspberry Pi, la figure III-28 représente cette conversion.

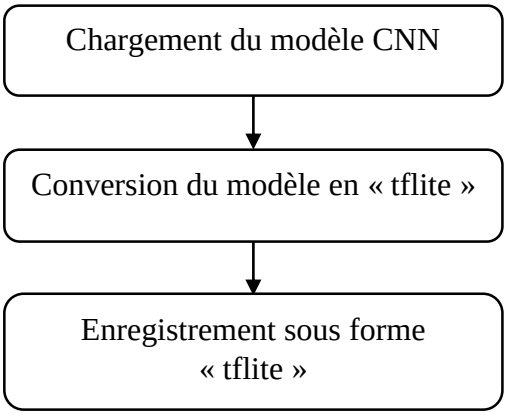



**Figure III-28 :** Préparation des données

- **Chargement du modèle :** nous avons chargé notre propre modèle réalisé avec la commande « **load_model** ».
- **Conversion du modèle :** nous avons utilisé la bibliothèque « **tflite.TFLiteConverter.from_keras_model** » pour convertir le modèle keras de la forme « **h5** » a la forme « **tflite** » pour être adapter au Raspberry Pi.
- **Enregistrement du modèle :** nous avons ouvert un nouveau fichier sous forme « tflite » et sauvegardé le modèle converti dans ce fichier.

### III.10.5-Résultat d'implémentation

Après avoir effectué la partie de préparation des données et le chargement de notre modèle, nous avons essayé de tester notre modèle sur des signaux de test finale avec une fonction de prédiction. Ensuite nous avons calculé la durée de prédiction, la figure III-29 représente les résultats de prédiction pour un seul signal.

```
[   1 1024    1]
1
Notre cas est de classe : Fibrillation Auriulaire
prediction time is 11.379744999885588 ms
```

**Figure III-29 :** Résultats de la prédiction d'un seul signal

De l'étude approfondi de ces résultats de Raspberry Pi 3B+, nous avons déduit que la prédiction est très rapide et fiable par rapport à un médecin.

Nous avons également testé sur plusieurs signaux au même temps et affiché le temps de prédictions, la figure III-30 représente les différents résultats obtenus.

```
Notre cas est de classe : Rythme Sinusal Normal (NSR)
Notre cas est de classe : Rythme Sinusal Normal (NSR)
Notre cas est de classe : Rythme Sinusal Normal (NSR)
Notre cas est de classe : Rythme Sinusal Normal (NSR)
Notre cas est de classe : Fibrillation Auriculaire (AFIB)
Notre cas est de classe : Fibrillation Auriculaire (AFIB)
Notre cas est de classe : Fibrillation Auriculaire (AFIB)
Notre cas est de classe : Fibrillation Auriculaire (AFIB)
Notre cas est de classe : Contraction Prematuré Ventriculaire (PVC)
Notre cas est de classe : Rythme Sinusal Normal (NSR)
Notre cas est de classe : Contraction Prematuré Ventriculaire (PVC)
Notre cas est de classe : Contraction Prematuré Ventriculaire (PVC)
Notre cas est de classe : Battement du Bloc de Branche Gauche (LBBBB)
Notre cas est de classe : Battement du Bloc de Branche Gauche (LBBBB)
Notre cas est de classe : Battement du Bloc de Branche Gauche (LBBBB)
Notre cas est de classe : Battement du Bloc de Branche Gauche (LBBBB)
prediction time is 580.3981240001121 ms
```

**Figure III-30 :** Résultats de la prédiction pour 16 cas au même temps

Nous avons remarqué que pour 16 signaux (personnes), 15 signaux (personnes) sont classés justes é l'exception d'un seul signal mal classé car les signaux de la troisième et la première classe sont un peu similaires.

Cependant, On a une durée de prédiction de 0.580s qui est très rapide et fiable par rapport au diagnostique d'un médecin cardiologue.

| | **PC** | | **Raspberry Pi** |
|---|---|---|---|
| | **Modèle h5** | **Modèle tflite** | **tflite** |
| **1 personne** | 0.072s | 0.021s | 0.011s |
| **16 personnes** | 0.140s | 0.110s | 0.580s |

**Tableau III-8 :** Résultats des implémentations

D'après les résultats du tableau, on a conclué que :

- Le modèle sous forme « tflite » est plus rapide pour la prédiction que celle de la forme « h5 » car il est plus léger, quantifié et assure une grande précision.
- Pour le Raspberry Pi 3B+, il a une durée de prédiction est très restreinte par rapport aux autres cartes.
- On a également remarqué que la durée de prédiction très restreinte par un PC dont le modèle testé est sous forme tflite comparée à celle du Raspberry Pi sachant que le modèle a été testé sur plusieurs cas.

### III.10.6-Node-Red

C'est un outil de visualisation, Node-Red est développé par IBM pour programmer des applications IOT et connecter des périphériques matériels, des API et des services en ligne dans

le cadre de l'IOT. Il fournit un éditeur basé sur un navigateur web pour connecter facilement les flux à l'aide des différents nœuds de la palette et déployer dans leur environnement d'exécution en un seul clic. Les flux créés dans le Node-Red sont stockés à l'aide de « **JSON** » et peuvent être facilement importés et exportés pour être partagés avec d'autres.

### III.10.6.1-Editeur Node-Red

La fenêtre d'éditeur se compose des composants suivants (figure III-31) :

- **L'entête :** contenant :
  - Le bouton de déploiement.
  - Le menu principal.
  - Le menu d'utilisateur.
  - L'état d'authentification de l'utilisateur.
- **La palette :** contenant :
  - Les nœuds disponibles.
- **L'espace de travail :** c'est l'espace principal au milieu, où les flux sont créés.
- **La barre de débogage et information :** se situer à droite.




**Figure III-31 : Editeur Node-Red.**

### III.10.6.2-Les nœuds

- **Les nœuds d'entrées :** La palette à gauche contient les différents nœuds d'entrées de base installées par défaut. Ils couvrent les mécanismes de communication de base.
- **Les nœuds de traitements :** Ces nœuds effectuent des fonctions de traitements spécifiques.

- **Les Nœuds de sorties :** sont essentiellement des images miroir de l'ensemble de base des nœuds d'entrées, ils fournissent un moyen d'envoyer des données sur le même ensemble des protocoles.

### III.6.3-Réception et visualisation des données

La réception des données par le nœud d'entrée (serial) est à base d'Arduino Uno. Les trames de type json qui arrivent, sont fractionnées pour extraire les données relatives du capteur afin de les présenter sur interface graphique et les enregistrer sur un fichier csv. Pour cela nous utilisons les nœuds suivants :

- Serial input.
- Json.
- Debug.
- Function.
- File write.
- Chart.

La liaison des blocs est réalisée dans la figure III-32.

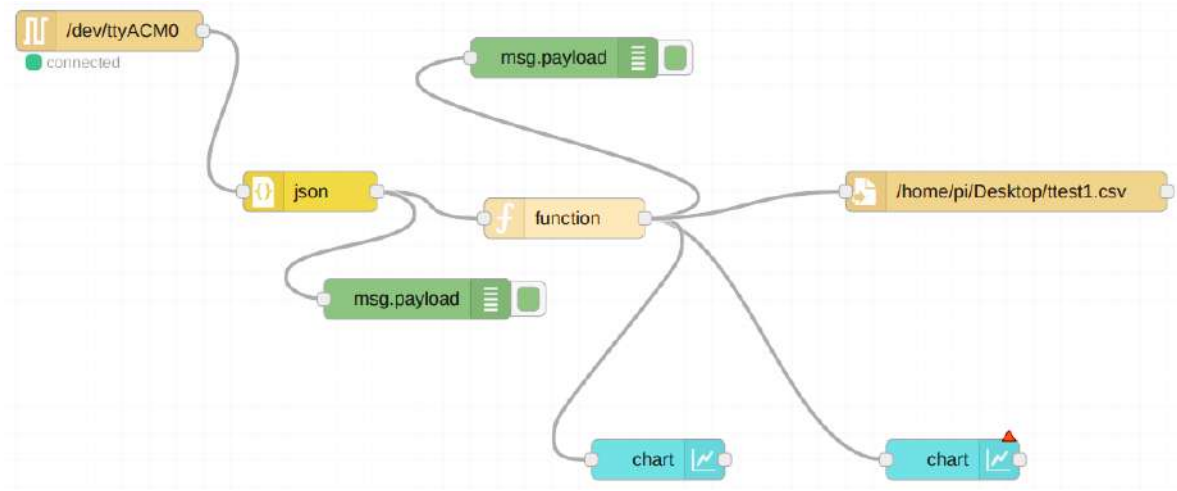



**Figure III-32 :** Schéma des nœuds pour visualiser et sauvegarder les signaux

Le fichier résultant csv contient le signal récolté par le capteur d'ECG AD8232.

Les signaux sauvegardés dans cette partie seront pris en charge dans la partie de test et d'affichage.

### III.6.4-Réception et visualisation des résultats de prédiction

Nous avons utilisé d'autres nœuds pour exécuter (voir figure III-33) et visualiser les résultats de prédiction, ils sont nommés comme suit :

- Pythonshell in : pour injecter le programme python.
- Debug : pour l'affichage des résultats.
- File read : pour charger le fichier csv du signal.
- Timestamp.
- Csv.

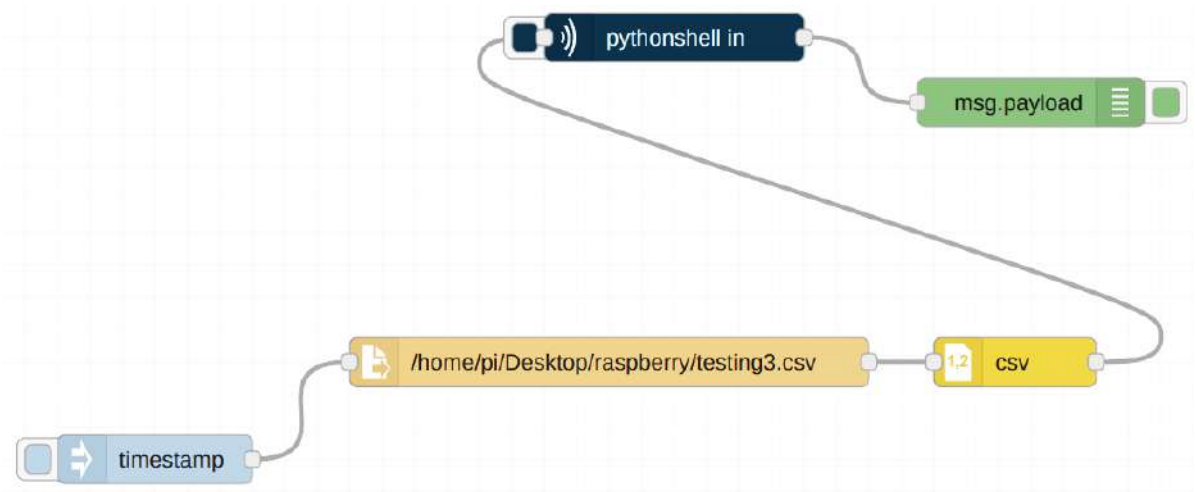



**Figure III-33 :** Schéma des nœuds pour l'affichage des résultats

Les signaux utilisés dans cette partie de test sont des signaux enregistrés dans l'étape précédente.

Les signaux injectés pour le test doivent être convertis en format json pour la compatibilité avec la plateforme.

Après l'exécution du programme python nous affichons les résultats comme la figure III-34 le montre :

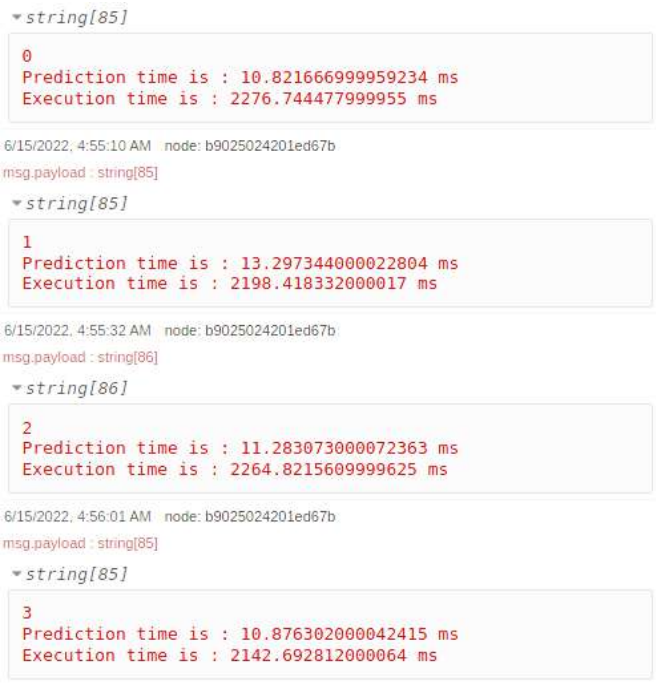



**Figure III-34 :** Les résultats de prédictions

La durée d'exécution est la durée totale de :

- Préparation du signal.
- Conversion en format json.
- Prédiction.
- Affichage des résultats.

## III.12-Conclusion

Dans ce chapitre, nous avons découvert une base de données des signaux ECG qui a été traitée et segmentée pour être sauvegarder sous forme d'un fichier csv. Ensuite on a réalisé l'étape de traitement pour servir d'entrée à un réseau CNN 1D afin de l'entrainer à détecter les différentes maladies cardiaques. La dernière étape consiste à tester ce réseau sur des signaux de test final et ce avant d'effectuer l'implémentation sur la carte Raspberry Pi 3B+ et mesurer le temps d'exécution avec l'affichage des résultats sur la plateforme Node-Red.

# Conclusion générale

# Conclusion générale

L’objectif de ce projet est de développer un système intelligent capable de détecter et classer les maladies cardiovasculaires systématiquement à l'aide des réseaux CNN. À travers ce travail On a déduit qu'une segmentation des signaux ECG est efficace dans la classification des signaux ECG car elle permit d'élargir et équilibrer la base de données.

Le système intelligent proposé a été testé par des signaux ECG qui sont récoltés à l’aide d’un capteur ECG AD8232. Les résultats de la prédiction obtenus, nous mènent à conclure le type de maladie cardiovasculaire dont souffre la personne concernée.

Les tests ont apporté des résultats concluant en matière de justesse de classification, de détection et du temps d’exécution, raison pour laquelle nous avons implémenté ce système intelligent sur la carte Raspberry Pi 3B+.

Après l’implémentation on a testé pour une seconde fois ce système intelligent afin de déceler l’efficacité du système et déduire la durée d’exécution dans une carte Raspberry Pi 3B+.

On espère finalement que le système intelligent de détection et de classification des maladies cardiovasculaires que nous avons développé, puisse être utilisé par les cardiologues et les services des urgences au niveau des hôpitaux où la précision et le temps peuvent sauver des vies humaines en danger. Cela promet d’innover dans le système de santé pour un avenir meilleur.

# Bibliographies

# Bibliographies


[1]. https://www.who.int/fr/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds), consulté le 05/06/2022.

[2]. Cabrol, C., R. Vialle, and H. Guérin-Surville, *Anatomie du cœur humain.* Niveau PCEM2-EIA cardiologie. Université Pierre et Marie Curie, 2002.

[3]. St-Jean, R., *Le role de la phase du cycle cardiaque sur le potentiel evoque moyen a un stimulus somesthesique chez l'humain*1982: Université du Québec à Montréal.

[4]. Zemzemi, N., *Etude théorique et numérique de l'activité électrique du coeur : applications aux électrocardiogrammes*, 2009. p. 1 vol. (247 p.).

[5]. D.Timmis, A., A. Nathan, and I. D.sllivan, *En bref.*

[6]. Abbas, A.K. and R. Bassam, *Phonocardiography Signal Processing*2009: Morgan & Claypool Publishers.

[7]. Panagides, V., *ECG électrocardiogramme*2017: Editions Vernazobres-Grego.

[8]. Houghton, A.R., D. Gray, and F. Jan, *Maîtriser l'ECG: De la théorie à la clinique*2011: Elsevier Health Sciences France.

[9]. *Larousse médical*2006: Larousse.

[10]. BOUKHENOUS, M.S., *Implémentation sur FPGA d'un système intelligent pour la surveillance de l'état de santé des patients souffrants d'arythmie cardiaque*, 2021, Université des Sciences et de la Technologie Houari Boumediène.

[11]. Chapelon-Abric, C., *Électrocardiogramme normal de l'adulte.* EMC-Cardiologie-Angéiologie, 2004. **1**(1): p. 2-10.

[12]. François Christophe, S.P., *Génie électrique - Exercices et problèmes corrigés - Les grandes fonctions de la chaîne d'information - IUT, BTS, CPGE (TSI et ATS), écoles d'ingénieurs*2018: Editions Ellipses.

[13]. Beya, O., *Analyse et reconnaissance de signaux vibratoires: contribution au traitement et à l'analyse de signaux cardiaques pour la télémédecine*, 2014, Dijon.

[14]. Kirchhof, P., et al., *2016 ESC Guidelines for the management of atrial fibrillation developed in collaboration with EACTS.* European Heart Journal, 2016. **37**(38): p. 2893-2962.

[15]. ALI, K., *Doctorat en sciences*, 2017, Université Blaise Pascal, Clermont Ferrand France.

[16]. Simpson Jr, R.J., et al., *Prevalence of premature ventricular contractions in a population of African American and white men and women: the Atherosclerosis Risk in Communities (ARIC) study.* American heart journal, 2002. **143**(3): p. 535-540.

[17]. Shah, F.Y., et al., *Mortality and Incidence Rate of SARS-CoV-2 (COVID-19) Infection in Patients Admitted and Operated for Hip Fracture during SARS-CoV-2 pandemic in a London Hospital.* Revista Brasileira de Ortopedia, 2021. **56**: p. 594-600.

[18]. Huang, S.-C. and T.-H. Le, *Principles and labs for deep learning*2021: Academic Press.

[19]. Zhang, X.-D., *Machine Learning*, in *A Matrix Algebra Approach to Artificial Intelligence*, X.-D. Zhang, Editor 2020, Springer Singapore: Singapore. p. 223-440.

[20]. Mohammed, M., M.B. Khan, and E.B.M. Bashier, *Machine learning: algorithms and applications*2016: Crc Press.

[21]. Rosebrock, A., *Deep Learning for Computer Vision with Python: Starter Bundle*2017: PyImageSearch.

[22]. Géron, A., *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*2019: " O'Reilly Media, Inc.".

[23]. Abraham, A., *Artificial neural networks.* Handbook of measuring system design, 2005.

[24]. Nordby, J.O., *Environmental sound classification on microcontrollers using Convolutional Neural Networks*, 2019, Norwegian University of Life Sciences, Ås.

[25]. Maas, A.L., A.Y. Hannun, and A.Y. Ng. *Rectifier nonlinearities improve neural network acoustic models*. in *Proc. icml*. 2013. Citeseer.

[26]. Clevert, D.-A., T. Unterthiner, and S. Hochreiter, *Fast and accurate deep network learning by exponential linear units (elus).* arXiv preprint arXiv:1511.07289, 2015.

[27]. Ketkar, N. and J. Moolayil, *Convolutional Neural Networks*, in *Deep Learning with Python: Learn Best Practices of Deep Learning Models with PyTorch*, N. Ketkar and J. Moolayil, Editors. 2021, Apress: Berkeley, CA. p. 197-242.

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

[29]. Hassija, V., et al., *A Survey on IoT Security: Application Areas, Security Threats, and Solution Architectures.* IEEE Access, 2019. **7**: p. 82721-82743.

[30]. Ismail, Y., *Internet of things (iot) for automated and smart applications*2019: BoD–Books on Demand.

[31]. Domingo, M.C., *An overview of the Internet of Things for people with disabilities.* journal of Network and Computer Applications, 2012. **35**(2): p. 584-596.

[32]. Puccinelli, D. and M. Haenggi, *Wireless sensor networks: applications and challenges of ubiquitous sensing.* IEEE Circuits and systems magazine, 2005. **5**(3): p. 19-31.

[33]. Yick, J., B. Mukherjee, and D. Ghosal, *Wireless sensor network survey.* Computer networks, 2008. **52**(12): p. 2292-2330.

[34]. Simard, J., *La ville intelligente comme vecteur pour le développement durable: le cas de la ville de Montréal*, 2015, Université de Sherbrooke.

[35]. Miori, V. and D. Russo. *Anticipating health hazards through an ontology-based, IoT domotic environment*. in *2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*. 2012. IEEE.

[36]. Zhao, J.-c., et al. *The study and application of the IOT technology in agriculture*. in *2010 3rd international conference on computer science and information technology*. 2010. IEEE.

[37]. Sahraoui, S., *Mécanismes de sécurité pour l'intégration des RCSFs à l'IoT (Internet of Things)*, 2016, Université de Batna 2.

[38]. http://www.physionet.org, consulté le 25/04/2022.

[39]. https://www.mathworks.com/discovery/what-is-matlab.html, consulté le 29/05/2022.

[40]. https://www.tensorflow.org/, consulté le 29/05/2022.

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

[42]. Hao, J. and T.K. Ho, *Machine Learning Made Easy: A Review of Scikit-learn Package in Python Programming Language.* Journal of Educational and Behavioral Statistics, 2019. **44**(3): p. 348-361.

[43]. https://matplotlib.org/, consulté le 25/05/2022.

[44]. https://keras.io/api/callbacks/model_checkpoint/, consulté le 25/05/2022.

[45]. Singh, P. and A. Jasuja. *IoT based low-cost distant patient ECG monitoring system*. in *2017 International Conference on Computing, Communication and Automation (ICCCA)*. 2017. IEEE.

[46]. Prasad, A.S. and N. Kavanashree. *ECG monitoring system using AD8232 sensor*. in *2019 International Conference on Communication and Electronics Systems (ICCES)*. 2019. IEEE.

[47]. Fezari, M. and A. Al Dahoud, *Integrated Development Environment "IDE" For Arduino.* WSN applications, 2018: p. 1-12.

[48]. Zhao, C.W., J. Jegatheesan, and S.C. Loon, *Exploring iot application using raspberry pi*. International Journal of Computer Networks and Applications, 2015. **2**(1): p. 27-34.

[49]. Sałuch, M., et al. *Raspberry PI 3B+ microcomputer as a central control unit in intelligent building automation management systems*. in *MATEC Web of Conferences*. 2018. EDP Sciences.

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

# Annexe

# Annexe A

**Tableau de Donnée du Matlab.**

| **Matlab** | | |
|---|---|---|
| **Créateur** | Clever Moler. | |
| **Développé par** | MathWorks.. | |
| **Première version** | 1984. | |
| **Dernière Version** | R2022a (09/03/2022). | |
| **Ecrit en** | ➢ C.<br>➢ C++. | ➢ Fortran.<br>➢ Java. |
| **Système d'exploitation** | ➢ Microsoft Windows.<br>➢ Mac OS. | ➢ Linux. |
| **Environnement** | ➢ Linux.<br>➢ Unix. | ➢ Mac OS.<br>➢ Windows. |
| **Format Lus** | ➢ Matlab M-File.<br>➢ Mat-fille.<br>➢ Mat4.<br>➢ Matlab app installer. | ➢ BigTIFF.<br>➢ Matlab Simulink.<br>➢ Matlab license passcode.<br>➢ Matlab compiler Project. |
| **Formats écrits** | ➢ Matlab M-Fille.<br>➢ Matlab Simulink.<br>➢ Matlab app installer. | ➢ Matlab compile Project.<br>➢ Matlab Support package. |
| **Type** | Calcul numérique. | |
| **Licence** | Propriétaire. | |
| **Site web** | www.mathworks.com | |

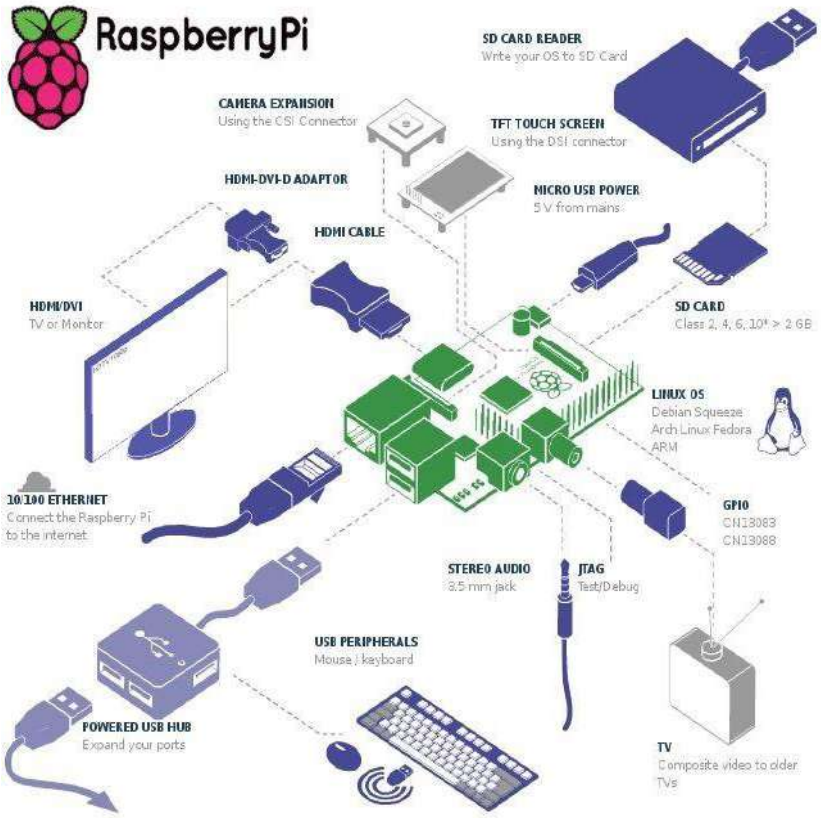
# Annexe B

**Tableau de Donnée du Python.**

| **Python** | | |
|---|---|---|
| **Première Version** | 20 février 1991. | |
| **Paradigmes** | Objet, impératif et fonctionnel. | |
| **Auteur** | Guido Van Rossum. | |
| **Développeurs** | Python Software Fondation. | |
| **Dernière Version** | 3.10.4 (24/03/2022).<br>3.9.13 (17/05/2022). | |
| **Typage** | ➢ Fort.<br>➢ Dynamique. | ➢ Duck typing.<br>(dactylographie de canard) |
| **Influencé par** | ➢ ABC.<br>➢ C.<br>➢ Eiffel.<br>➢ ICON.<br>➢ Modula-3. | ➢ Java.<br>➢ Perl.<br>➢ Smalltalk.<br>➢ Tcl. |
| **A influencé** | ➢ Ruby.<br>➢ Groovy. | ➢ Boo.<br>➢ Julia. |
| **Implémentations** | ➢ CPython.<br>➢ Jython. | ➢ IronPython.<br>➢ PyPy |
| **Ecrit en** | ➢ C pour CPython.<br>➢ Java pour Jython.<br>➢ C# pour IronPython. | ➢ Python pour lui-même.<br>➢ Python pour PyPy. |
| **Paradigme** | Multi Plateforme. | |
| **Système d'exploitation** | ➢ Windows.<br>➢ Mac OS. | ➢ Linux/Unix.<br>➢ Android. |
| **Licence** | Python Software Fondation License. | |
| **Site web** | www.python.org | |
| **Extension de fichier** | ➢ Py.<br>➢ Pyc.<br>➢ Pyd.<br>➢ Pyo. | ➢ Pyw.<br>➢ Pyz.<br>➢ Pyi. |

# Annexe C

## Les composants de la carte Raspberry Pi 3B+



# Annexe D

**Tableau des Caractéristiques de Raspberry Pi 3B+.**

| | |
|---|---|
| **Processeur** | Broadcom BCM2837B0, Cortex-A53 64-bit SoC @1.4GHz. |
| **Ram** | 1GB LPDDR2 SDRAM |
| **Connéctivité** | 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless Lan, Bluetooth 4.2, BLE<br>Gigabit Ethernet over USB 2.0<br>4 USB 2.0 Ports |
| **Accès** | En-tête GPIO étendu à 40 Broches |
| **Vidéo & Son** | Full Size HDMI<br>Port d'affichage MIPI DSI pour connecter un écran tactile Raspberry Pi<br>Port de caméra MIPI CSI pour connecter une caméra Raspberry Pi<br>Sortie stéréo à 4 pôles et port vidéo composite |
| **Carte SD** | Port Micro SD pour le chargement du système d'exploitation et stockage des données |
| **Input Power** | 5V/2.5A DC via un connecteur micro USB<br>5V DC via un GPIO<br>Alimentation par Ethernet (PoE) activée (nécessite un HAT PoE distinct) |
| **Environnement** | Température de fonctionnement $[0-50]°C$ |
| **Système** | Debian GNU/Linux 11 (bullseye) |