

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de L'Enseignement Supérieur et de la Recherche Scientifique

Université Mouloud MAMMERY de TIZI-OUZOU



Faculté de Génie Electrique et D'informatique

Département D'informatique

MEMOIRE DE FIN D'ETUDES

En vue de l'obtention du Diplôme de Master Académique en informatique.

Spécialité : Réseaux, Mobilité et Systèmes Embarqués

Thème

**Reconnaissance d'activités humaines à l'aide de
capteurs de smartphone**

Dirigé par :

M^r LAGHROUCHE Mourad

Réalisé par :

M^{elle} BOULTACHE Thanina

M^r LAMZAOUI Amar

Devant le jury :

Président : M^r DAOUI Mehammed

Promoteur: M^r LAGHROUCHE Mourad

Examinatrice : M^{me} AOUDJIT Rachida

Invite : M^r ACHOUR Brahim

Promotion 2019-2020

Remerciement

Nous tenons à remercier notre encadreur Mr ACHOUR qui nous a guidés, et fait profiter de ses expériences tout au long de notre travail.

Notre profonde gratitude et nos sincères remerciements vont à notre promoteur Mr LAGHROUCHE à sa disponibilité, ses recommandations, sa confiance et son aide dans la correction de ce mémoire.

Nous tenons aussi à remercier l'ensemble des enseignants du département Informatique pour les efforts qu'ils ont déployés pour assurer notre formation.

Nous remercions les membres du jury pour avoir accepté d'évaluer notre modeste travail.

Enfin, nous adressons nos remerciements les plus sincères à nos familles, tous nos amis ainsi que toute personne ayant contribué de près ou de loin à la réalisation de ce projet.

En espérant que ce modeste travail soit à la hauteur et reflète ce que nous avons pu acquérir pendant ce travail.

Dédicace

À ma Mère qui a toujours travaillé pour assurer notre confort et réussite. À mon Père sans qui je ne serai jamais là où je suis maintenant.

À mes Sœurs

KAHINA et SONIA qui n'ont jamais cessé de me supporter, encourager et de croire en moi.

Ainsi qu'à toutes mes tantes et oncles, cousins, cousines pour leurs soutiens.

À mes chers Amis

Pour la meilleure compagnie qui puisse être et qui m'ont aidé et soutenu dans plusieurs situations.

À GDG_TIZI OUZOU

Qui est comme une deuxième famille pour moi.

Pour finir une dédicace particulière à mon binôme

« AMAR ».

Thanina

Dédicace

À Mes Parents et ma sœur Mélissa qui n'ont jamais cessé de me supporter et de croire en moi et qui m'ont encouragé.

À mes chers amis Mourad, Pedro, et Simo pour la meilleure compagnie qui puisse être et qui m'ont aidé dans plusieurs situations.

Sans oublier mon binôme « Thanina »

AMAR

Résumé

Dans le monde d'aujourd'hui, avec l'utilisation croissante des téléphones portables équipés d'une large gamme de capteurs embarqués tels que le GPS, le gyroscope, l'accéléromètre, etc. Il est devenu plus facile d'analyser des activités humaines quotidiennes afin de pouvoir garantir une surveillance de santé, le suivi de la condition physique ou les systèmes adaptatifs à l'utilisateur.

Dans ce contexte, nous présentons un système de reconnaissance d'activité en utilisant l'apprentissage supervisé à partir d'un ensemble de données contenant des enregistrements de signaux d'accéléromètre et gyroscope intégrés dans les smartphones de différents utilisateurs qui effectuent, à différents endroits, plusieurs activités physiques telles que la marche, le jogging, la montée des escaliers, etc. Pour construire le modèle, l'apprentissage en profondeur, les réseaux de neurones récurrents (RNN) et les réseaux de mémoire à court terme (LSTM) ont été utilisés. Après l'entraînement, le modèle a été enregistré et exporté vers une application Android pour les prédictions en temps réel, avec une interface utilisateur pour exprimer les résultats à l'aide de l'API de synthèse vocale.

Mots-clés : Reconnaissance d'activités humaines, Apprentissage machine, Apprentissage en profondeur (Deep Learning), Réseaux de neurones, Classification de données.

Abstract

In today's world, with the increasing use of cell phones equipped with a wide range of embedded sensors such as GPS, gyroscope, accelerometer, etc. It has become easier to analyze daily human activities so that health monitoring, fitness tracking or user adaptive systems can be guaranteed.

In this context, we present an activity recognition system using supervised learning from a dataset containing recordings of accelerometer and gyroscope signals embedded in the smartphones of different users who perform, at different locations, several physical activities such as walking, jogging, climbing stairs, etc. To build the model, deep learning, recursive neural networks (RNN) and short-term memory networks (LSTM) were used. After training, the model was saved and exported to an Android app for real-time predictions, with a user interface for expressing the results using the text-to-speech API.

Keywords: Recognition of human activities, Machine learning, Deep Learning, Neural networks, Data classification.

Sommaire

Introduction générale.....	1
-----------------------------------	----------

CHAPITRE I : Un état de l'art sur la classification du comportement des Humains

1. Introduction.....	4
2. L'environnement intelligent	5
2.1 Ville intelligente.....	5
2.2 Maison intelligente.....	6
2.3 L'agriculture intelligente.....	6
3. Capteur intelligent	7
3.1. Capteurs attachés aux animaux.....	8
3.2. Capteurs présents dans les voitures.....	8
3.3 Capteurs attachés aux humains.....	9
4. Capteurs des humains	10
4.1 Surveillance de température	10
4.2 Localisation.....	11
4.3 Surveillance d'activité.....	11
5. La reconnaissance des activités humaines (RAH).....	11
5.1 Notions et Objectifs des 'RAH '	11
5.2 Domaines d'applications	12
5.3 Les différentes approches de RAH.....	13
6. Les capteurs utilisés dans les téléphones intelligents	18

6.1 Accéléromètre.....	19
6.2 Gyroscope.....	21
6.3 Capteur de gravité.....	21
7. Méthodes de Classification.....	22
7.1 Principe de la classification de formes.....	22
8. Techniques de classification	27
8.1 La classification supervisée.....	27
8.2 La classification non supervisée.....	31
9. Classification de l'activité physique.....	31
10. Conclusion	36

CHAPITRE II : Optimisation de l'architecture d'entraînement des données de l'activité physique par apprentissage en profondeur

1. Introduction.....	37
2. Les réseaux de neurones.....	38
2.1 Introduction.....	38
2.2 Définition.....	39
2.3 Applications.....	39
2.4 Fonctionnement.....	39
3. Modèle biologique.....	40
3.1 Définition et structure	40
3.2 Fonctionnement	41
4. Étude et synthèse d'un réseau de neurone formel.....	42
4.1. Structure des réseaux de neurone.....	44
4.1.1. Réseau mono-couche et réseau multi-couches.....	44
4.1.2. Réseaux récurrents et réseaux non récurrents.....	45

4.1.3. Fonctionnement d'un réseau	49
4.1.4. Apprentissage.....	49
4.1.5. Choix de l'échantillon d'apprentissage.....	50
5. Développement d'un réseau de neurones	50
5.1. Collecte des données.....	51
5.2. Analyse des données.....	51
5.3. Séparation des bases de données.....	52
5.4. Choix d'un réseau de neurones.....	52
5.5. Mise en forme des données pour un réseau de neurones	53
5.6. Apprentissage du réseau de neurones	53
5.7. Validation.....	53
6. Architecture proposée.....	54
6.1 Fonction de perte et optimisation.....	54
7. Conclusion.....	55

CHAPITRE III : Conception de la solution et résultats expérimentaux

1. Introduction	56
2. Conception architecturale.....	57
2.1. Acquisition de données.....	58
2.2 Prétraitement des données.....	60
2.2 Classification des données.....	65
3. Environnement et outils de développement	65
3.1 Acquisition des données.....	66
3.2 Développement du modèle.....	66

3.2.1 Google Colab.....	66
3.2.2. Python 3.6.....	67
3.2.3 Tensorflow.....	68
3.3. Développement de l'application.....	69
3.3.1 Le langage de programmation Java.....	69
3.3.2 Android Studio.....	69
4. Test et évaluation.....	70
4.1 Critères d'évaluation.....	70
4.2 Présentation des tests.....	70
4.3 Discussions.....	72
5. Implémentation du modèle sur une application Android.....	73
6. Conclusion.....	78
Conclusion Générale.....	79
Références bibliographiques	

Liste des Figures

Figure 1 : Les caractéristiques d'une ville intelligente.....	5
Figure 2 : Maison intelligente.....	6
Figure 3 : Exemple de serre intelligente.....	7
Figure 4 : Collier connecté pour chien.....	8
Figure 5 : Capteurs présents dans une voiture autonome.....	9
Figure 6 : Ensemble de capteurs attachés à des endroits différents du corps humain.....	10
Figure 7 : Classification des approches de reconnaissance de l'activité humaine.....	13
Figure 8 : Quelques technologies utilisées pour les HAR.....	15
Figure 9 : Technologie RFID (balise et antenne).....	18
Figure 10 : Référentiels utilisés dans les terminaux mobiles.....	19
Figure 11 : Signaux d'accélération collectés à partir de terminaux mobiles.....	20
Figure 12 Données collectées à partir d'un gyroscope.....	21
Figure 13 Données collectées par le capteur de gravité.....	22
Figure 14 Processus de classification.....	23
Figure 15 Distance euclidienne entre X et les éléments des deux classes Gr1 et Gr2.....	28
Figure 16: Discrimination à deux classes.....	29
Figure 17. Vue simplifiée d'un réseau artificiel de neurones.....	30
Figure 18 Représentation d'un neurone humain.....	41
Figure 19. Modèle d'un neurone artificiel.....	41
Figure 20 : Structure du neurone formel.....	43
Figure 21 : Architecture d'un réseau statique.....	45

Figure 22 : Schéma d'un réseau de neurones récurrents à une unité reliant l'entrée et la sortie du réseau. A droite la version « dépliée » de la structure.....	46
Figure 23 : Schéma d'un réseau LSTM à une unité.	48
Figure 24 : Procédure de développement du système de classification comportementale ...	57
Figure 25 : Exemple de la récolte des données d'un participant.....	58
Figure 26 : Cycle d'activité physique.....	59
Figure 27 Distribution des activités par participants.....	61
Figure 28. Distribution du nombre d'enregistrement en fonction des activités.....	62
Figure 29 Signaux accéléromètre durant différentes activités.....	63
Figure 30 Découpage des données en séquence (1)	63
Figure 31 Découpage des données en séquence (2)	64
Figure 32 Segmentation du DataSet	64
Figure 33 : La Fonction Fit.....	65
Figure 34 Capture d'écran de l'application.....	66
Figure 35. Environnement de Google Colab.....	67
Figure 36 Logo Python.....	67
Figure 37 Logo Tensorflow.....	68
Figure 38. Logo Java.....	69
Figure 39. Logo Android Studio.....	69
Figure 40 Graphe représentant le taux d'exactitude.....	71
Figure 41 Capture d'écran de l'application (Activité: Rester debout)	72
Figure 42. Capture d'écran de l'application (Activité: Marcher)	74
Figure 43 Capture d'écran de l'interface principal de l'application de visualisation.....	75
Figure 44. Capture d'écran des interfaces secondaires de l'application de visualisation.....	76

Liste des tableaux

Tableau 1 : Matrice de confusion dans le cas binaire.....	26
Tableau 2 : Résumé des travaux effectués dans le domaine de classification des activités physiques. Tiré de : Robertas Damasevicius et al [35]	35
Tableau 3 : Aperçu de la base de donnée	60
Tableau 4 : Matrice de confusion du classificateur.....	72

Liste des Formules

Formule 1: Accélération linéaire.....	20
Formule 2 : Taux de classification.....	26
Formule 3 : Déduction de taux d'erreur.....	26
Formule 4 : Paramètres d'évaluation d'un classificateur binaire.....	27
Formule 5 : Calcul distance euclidienne pour déterminer la similarité.....	28
Formule 6 : L'état courant d'un neurone.....	46
Formule 7 : L'état courant d'un neurone en appliquant la fonction d'activation.....	47
Formule 8 : Etat de sortie.....	47
Formule 9 : Input gate.....	48
Formule 10: Forget gate.....	49
Formule 11: Output gate.....	49
Formule 12 : Fonction L2-norm.....	54

Liste des abréviations

TIC : Technologies de l'information et de la communication

IOT: Internet of Things

BAN: Body Area Network

HAR : Human activity recognition (reconnaissance d'activités humaine)

ARC : Chaîne de reconnaissance des activités

SVM : Support-Vector-Machine

CSI : Channel State Information

RFID : Radio frequency identification

Acc : Accéléromètres

Gyr : Gyroscope

Grav : Gravité

ACP : Analyse en composantes principales

KNN : K-Nearest Neighbors classifier

RNN: Recurrent neural network

LSTM: Long Short Term Memory

L2-norm: Least Squared Error

Introduction Générale

1. Contexte et Motivations

La définition de l'activité physique la plus répandue est celle qui a été proposée par Caspersen, en 1985. Il s'agit de « l'ensemble des mouvements corporels produits par la mise en action des muscles squelettiques et entraînant une augmentation substantielle de la dépense énergétique au-dessus du métabolisme de repos » [1]. Quant à la notion de la reconnaissance de l'activité physique, la plupart des travaux la définissent comme étant « l'ensemble des techniques et modèles statistiques ou quantitatifs permettant d'identifier avec une certaine précision l'état de l'ensemble des mouvements corporels», grâce à l'utilisation de différents dispositifs et capteurs [2][3].

La reconnaissance de l'activité physique joue un rôle important dans plusieurs domaines. En effet, la médecine, le bien être, le sport, l'énergie, le bâtiment, l'analyse du comportement et bien d'autres domaines s'intéressent de plus en plus à l'état de l'activité d'un sujet [4]. Dans le domaine de la santé, par exemple, on voit de plus en plus de dispositifs médicaux, développés par des entreprises privées, se basant sur un système de reconnaissance de l'activité physique pour calculer des indicateurs de santé, telle que la fréquence cardiaque, l'analyse du pouls, la pression artérielle, etc [5]. Dans le secteur de l'énergie et du bâtiment, la détection de l'activité physique peut renseigner les professionnels de l'énergie sur le comportement des occupants et d'agir en conséquence. En effet, en adaptant la consommation de l'énergie selon les comportements des usagers, ceci peut permettre aux consommateurs de réaliser des économies [6]. Dans le secteur du commerce et du marketing, les professionnels du secteur s'intéressent de plus en plus à l'étude des comportements de leurs clients afin de leur proposer des services adaptés et personnalisés [7].

L'arrivée des objets connectés a donné une nouvelle ère à ces différents domaines applicatifs. De nouveaux dispositifs portables et connectés intégrant différents types de capteurs ont vu le jour. Par exemple, les smartphones permettant de passer des appels téléphoniques, de se connecter à internet et d'envoyer des textos. Les smartwatches et les bracelets connectés, permettant l'affichage et le calcul du nombre de pas effectués dans la journée, la fréquence cardiaque, etc. D'autres dispositifs commencent à voir le jour comme les semelles connectées, des vêtements connectés, des smart TV et bien d'autres objets [8]. Ces nouveaux dispositifs génèrent ainsi de nouveaux usages (par exemple, allumer son téléviseur avec son smartphone, répondre à un appel téléphonique avec sa smartwatch,.. etc), c'est ce que nous désignons par le terme « nouveaux usages ou usage » [9].

L'ensemble de ces objets connectés portables sont, en général, portés par les usagers. Ainsi, beaucoup de travaux les ont explorés afin de reconnaître différents types d'activités. Les techniques ainsi utilisées reposent généralement sur l'utilisation d'approches à base d'algorithmes d'apprentissage automatique [10].

2. Problématique

Des études récentes ont montré l'importance de la reconnaissance de l'activité physique avec des objets connectés (smartphones, montres connectées..). Cependant, l'étude et le développement des systèmes de reconnaissance de l'activité physique font face à trois problématiques de taille. Tout d'abord les environnements des expériences réalisées ne permettent pas d'explorer efficacement les méthodes de traitement. La majorité des bases de données disponibles ont été collectées dans des environnements contrôlés et avec des dispositifs intrusifs (capteurs placés à différents endroits du corps, détection de l'attention à la TV avec un traqueur des yeux, etc.). D'autre part, les techniques de reconnaissances de l'activité physiques employées ne sont pas adaptées aux différentes sources de données disponibles (en termes d'incertitude, d'erreurs et plus généralement d'imperfection). Enfin, l'exploration des techniques d'apprentissage en profondeur reste très limitée. La majorité des modèles nécessitent d'être optimisés afin d'avoir des résultats interprétables.

3. Objectifs et hypothèses de recherche

L'objectif général de ce projet de recherche consiste à développer un système de classification automatique des activités physiques de l'utilisateur d'un appareil mobile. Ce système se basera sur les données enregistrées à partir des capteurs du terminal mobile pour identifier les classes d'activités. Pour y parvenir, nous avons défini les objectifs spécifiques suivants :

1. Collecter les données de deux capteurs (accéléromètre, gyroscope), sur plusieurs participants.
2. Prétraitement des données enregistrées.
3. Conception du système de classification des données prétraitées afin d'identifier les différentes classes d'activités physiques.
4. Implémenter ce classifieur dans une application mobile en temps réel.

Notre hypothèse de recherche est donc qu'il est possible de classer les activités physiques en utilisant les données provenant d'un terminal mobile.

4. Plan du mémoire

Ce mémoire sera organisé comme suit :

Après cette brève introduction, nous ferons dans le chapitre 1 une revue de la littérature qui couvrira les cinq aspects suivant:

- (1) Les environnements intelligents
- (2) Les capteurs utilisés dans les téléphones intelligents
- (3) La reconnaissance d'activité humaine et les différentes approches existantes
- (4) Les techniques de classification de formes en général
- (5) plus spécifiquement pour l'activité physique.

Dans le chapitre 2, nous nous intéresserons aux réseaux de neurones, qui, comme nous le verrons sont des alternatives très prometteuse pour l'élaboration d'un système de classification de l'activité physique.

Dans le chapitre 3, nous détaillerons les différentes étapes qui ont abouti à la validation du système de classification, à savoir, la méthodologie concernant la collecte de données, le prétraitement et la classification, puis nous résumons les contributions apportées dans ce mémoire ainsi que les résultats obtenus. Pour finir, nous clôturerons ce mémoire, avec les perspectives liées aux travaux futurs.

Chapitre I

*Un état de l'art sur la
classification du
comportement des Humains*

1. Introduction

Depuis quelques années, nous assistons à un développement et diffusion exceptionnel de la microélectronique produisant des capteurs et des appareils mobiles et portables avec une variété de caractéristiques (différents types de capteurs, de systèmes embarqués hautes technologies, etc.). Leur puissance de calculs élevés, leur petite taille et leur faible coût ont permis aux différents utilisateurs d'interagir avec ces appareils en permanence. Ceci a fait émergé le domaine de la détection omniprésente (Ubiquitous Sensing, en anglais), un domaine de recherche actif dont l'objectif principale est d'extraire des connaissances à partir des données acquises par des capteurs embarqués^[11]. En particulier, la reconnaissance des activités humaines est devenue une tâche d'un grand intérêt dans plusieurs domaines, principalement pour les applications médicales, militaires^{[17][18]} et de sécurité^[12].

L'arrivée des objets connectés sur le marché a permis de faire émerger ces applications de suivi en les rendant plus flexibles. En effet, la détection des activités, qui se faisait traditionnellement, avec des capteurs fixés à différents endroits du corps, est aujourd'hui réalisée grâce à un ou plusieurs objets connectés (smartphone, smartwatch, etc.) sans avoir à attacher les objets connectés dans différents endroits du corps^[19]. Ce qui rend la tâche de détection plus flexible et pratique. Ainsi, le caractère intrusif des capteurs est largement diminué^[21].

L'objectif général de ce projet de recherche consiste à développer un système de classification automatique des activités physiques de l'utilisateur d'un appareil mobile. Ce système se basera sur les données enregistrées à partir des capteurs du terminal mobile pour identifier les classes d'activités.

Afin d'aborder au mieux ce sujet, nous proposons dans ce chapitre une revue de la littérature qui couvrira les cinq aspects suivants. Le premier portera sur les environnements intelligents et leurs différentes formes (Section 2). Le deuxième traitera des différents capteurs intelligents (Section 3), et plus spécifiquement des capteurs destinés aux humains (Section 4). Ce qui nous amènera à aborder la notion de "reconnaissance d'activité humaine" (HAR) ainsi que toutes ses implications (Section 5), pour ensuite donner une liste non exhaustive des différentes approches utilisées à ce jour (Section 5.3). Enfin, le sixième et dernier aspect traitera, plus spécifiquement, des méthodes de classification de formes utilisées dans le cadre de la classification de l'activité physique (Section 9).

2. L'environnement intelligent

L'intelligence est une capacité très évoluée de percevoir, retenir, récupérer, répéter, comparer, transformer et générer un motif. Toutes ces capacités impliquent la construction d'une certaine représentation interne, d'une certaine forme de mise en miroir du monde extérieur cependant on définit un environnement intelligent par un environnement dans lequel les actions de nombreux de contrôleurs, détecteurs ont été intégrés, et sont orchestrés par des processus d'auto programmation préventive (par exemple, des agents logiciels intelligents) de manière à créer une fonctionnalité holistique interactive qui améliore l'expérience des occupants. Ces environnements peuvent se présenter sous plusieurs formes parmi elles on citera :

2.1 Ville intelligente:

Les villes intelligentes (smart cities) constituent aujourd'hui un enjeu prioritaire à la fois pour le développement économique des territoires et pour le bien-être et le cadre de vie de ses habitants.

Une ville intelligente est une ville qui profite à plein de la transformation numérique et des TIC ainsi que de divers dispositifs physiques (capteurs) connectés au réseau (l'Internet des objets ou IOT) pour améliorer les services publics et les rendre plus personnalisés et plus efficaces telle que la surveillance et la gestion des systèmes de circulation et de transport, les centrales électriques, la gestion des déchets...etc. Elle consiste principalement à collecter et diffuser des données et se caractérise par plusieurs caractéristiques (Comme le montre la figure 1).

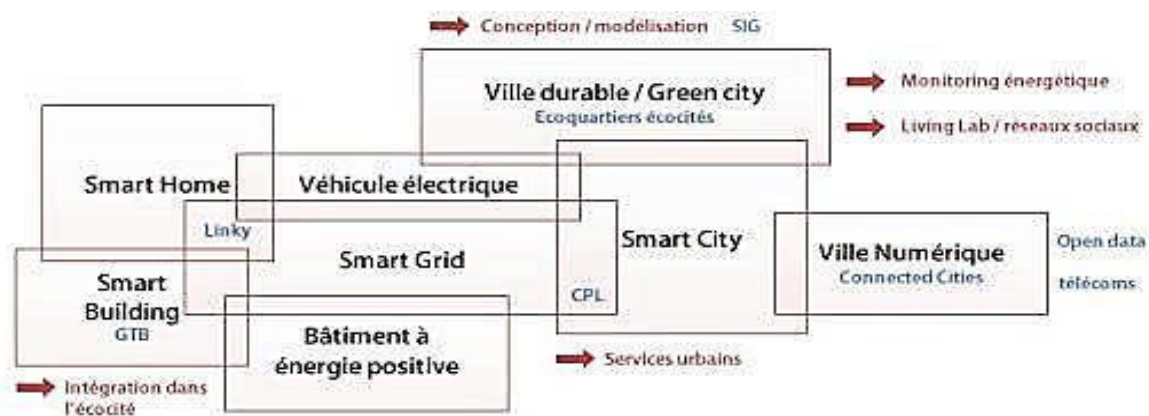


Figure 18 Les caractéristiques d'une ville intelligente

2.2 Maison intelligente :

Une maison intelligente est un lieu avec des systèmes hétérogènes à de nombreux dispositifs frontaux avec le support d'architectures d'informations et de communication intégrées. Gram-Hanssen, K., & Darby, S. S. J. (2018)^[22] la décrivent comme « une maison avec une intégration de dispositifs de détection et de communication numériques pour produire des services grâce à des communications transparentes ».

Un habitat intelligent s'appuie notamment sur la domotique, qui repose elle-même sur des appareils connectés capables de communiquer des informations, en réseau, à un terminal de centralisation (Tablette, Smartphone...), avec une intervention humaine réduite, produisant des services et des informations à partir d'un ensemble d'autres informations. Nous citons comme exemples : la possibilité de régler la lumière des pièces en fonction de la luminosité extérieure, ou encore augmenter le chauffage d'une pièce si l'un des habitants est malade

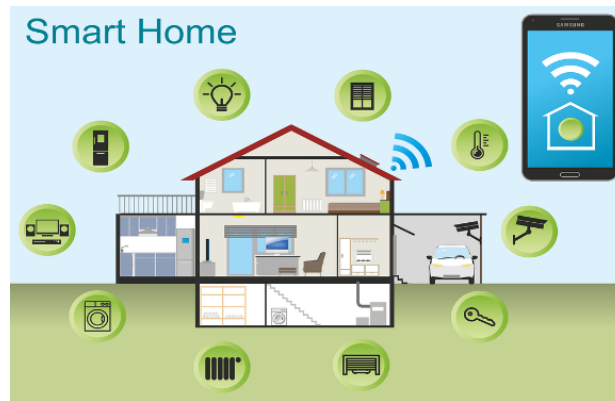


Figure 19 Maison intelligente

2.3 L'agriculture intelligente:

On définit l'agriculture intelligente par un concept de gestion agricole utilisant une technologie moderne pour augmenter la quantité et la qualité des produits agricoles.

Les agriculteurs du 21e siècle ont accès au GPS, à l'analyse des sols, à la gestion des données et à l'Internet des objets. En mesurant précisément les variations au sein d'un champ et en adaptant la stratégie en conséquence, les agriculteurs peuvent augmenter considérablement l'efficacité des pesticides et des engrais et les utiliser de manière plus sélective. De même, en utilisant des techniques d'agriculture intelligente, les agriculteurs peuvent mieux surveiller les besoins des animaux individuels et ajuster leur nutrition en conséquence, empêchant ainsi les maladies et améliorant la santé du troupeau.



Figure 20 Exemple de serre intelligente

- Tous les exemples cités au-dessus des environnements intelligents ont un point en commun qui est le « Smart Sensor » Ou « Capteur Intelligent » en Français, qui est un point important pour la réalisation de tels environnements, c'est pour cela que nous voulons réaliser un (Smart Sensor) à travers ce travail.

3. Capteur intelligent

Le « SmartSensor » est un élément crucial et intégral de l'Internet des objets (IoT), défini comme étant un appareil qui prend des entrées de l'environnement physique et utilise des ressources de calcul intégrées pour exécuter des fonctions prédéfinies lors de la détection d'une entrée spécifique, puis traiter les données avant de les transmettre.

Ces appareils sont utilisés pour la surveillance et le contrôle des mécanismes dans une grande variété d'environnements.

On trouve une grande variété de ces dispositifs qui peuvent être utilisés de différentes manières, ils peuvent être attachés à différents objets (voitures, humains, animaux...) ou peuvent être utilisés comme capteurs portables ou être déployés dans l'environnement.

3.1. Capteurs attachés aux animaux:

Pour le bien-être des animaux et pour surveiller en permanence leur santé un ensemble de moyens technologiques est mis en œuvre à base des capteurs attachés à ces premiers telle que :

- **Le collier connecté :**

Inventé pour l'analyse du comportement des animaux, en émettant des alertes s'il y a quelque chose qui ne tourne pas rond, grâce à un capteur, qui est capable de détecter si l'animal marche, dort, court, mange, se lèche, se gratte et aussi de les interpréter. Par exemple, un chien qui se gratte tout le temps, quand il est seul à la maison est un signe de stress.



Figure 21 Collier connecté pour chien

3.2 Capteurs présents dans les voitures:

En plus du confort d'un véhicule, une bonne conduite sans accidents de route est une question primordiale pour les constructeurs automobiles, chose qui les a poussés à utiliser la technologie des capteurs.

Des dizaines de capteurs embarqués informant en temps réel plusieurs calculateurs de bord. Qu'il s'agisse du détecteur de ceinture de sécurité, du capteur de pluie, de l'allumage automatique des phares ou du dosage de carburant nécessaire, tous ces capteurs ont pour objectif d'améliorer le confort, la sécurité et l'impact du véhicule sur l'environnement permettant ainsi d'effectuer plusieurs opérations (Par exemples : surveiller les lignes sur la route, un freinage actif qui peut ralentir la voiture à l'approche de la circulation debout...etc.),

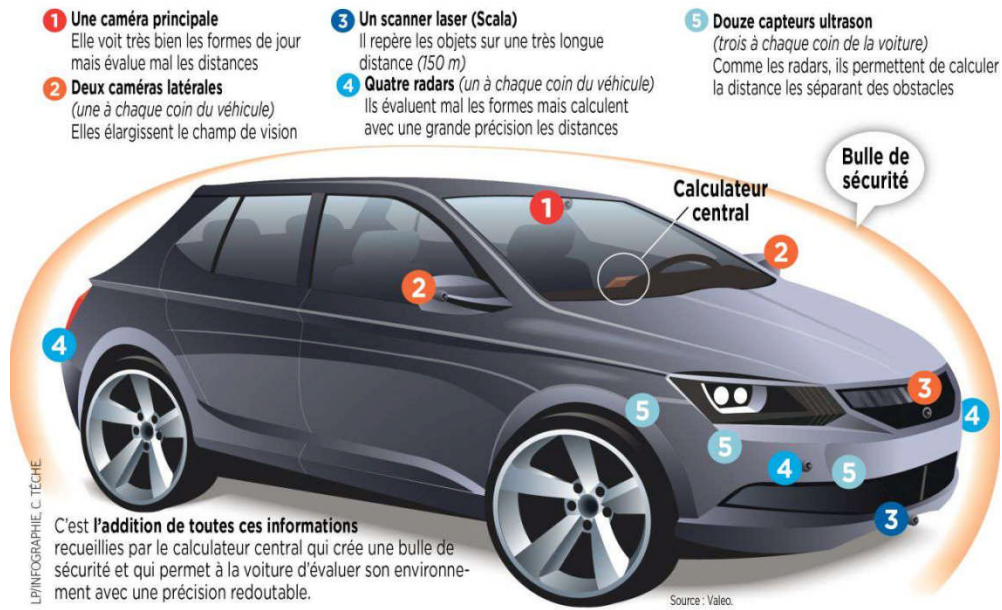


Figure 22. Capteurs présents dans une voiture autonome.

3.3 Capteurs attachés aux humains :

Au cours des dernières années, nous observons l'intérêt accru pour la technologie « Body Area Network (BAN) » ou ce qu'on appelle « réseau de zone corporelle » qui consiste à interconnecter sur, autour ou dans le corps humain différents capteurs afin de récolter des données pour traiter la surveillance et la détection de l'ensemble du système du corps humain.

Pour ce fait nous trouvons plusieurs types de capteurs, peuvent être attaché au différent endroit de corps humain (Figure.6), ayant chacun une fonction spécifique.

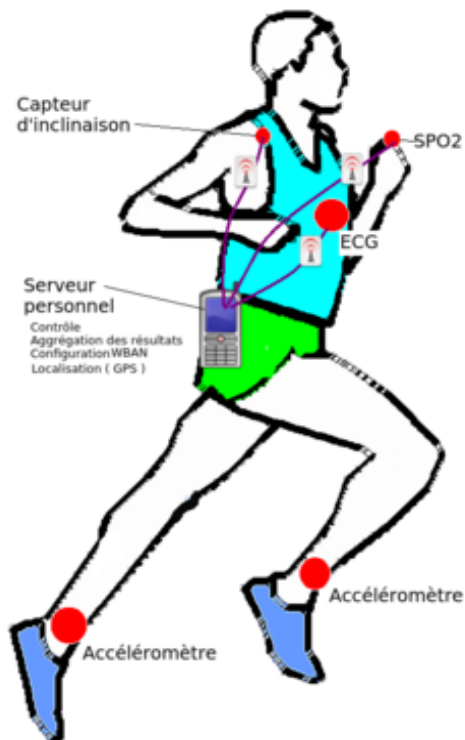


Figure 23 . Ensemble de capteurs attachés à des endroits différents du corps humain.

4. Capteurs des humains

Comme nous l'avons déjà mentionné précédemment, il y'a une grande attention aux capteurs des humains et cela dans de différents domaines tel que : la santé, le militaire, le divertissement, le sport, ou les Interactions homme-machine.

Voici quelques exemples de cas d'utilisations de ces capteurs :

4.1 Surveillance de température :

L'augmentation de la température du corps est le premier mécanisme de réponse à une agression virale, causant des fièvres qui peuvent être elles aussi origines d'autres maladies plus graves, ou même des déshydratations. C'est pour cela qu'il est très important de contrôler de la température de manière régulière.

Grâce au capteur intégré dans des objets, des montres par exemple, cette fonction de surveillance devint facile.

4.2 Localisation :

La géolocalisation de personnes est de plus en plus courante surtout pour les personnes âgées.

Ce système se fait en intégrant des capteurs dans des dispositifs mobiles (smartphones, montres ou bracelets connectés...) qui doivent accompagner l'utilisateur dans tous ses déplacements.

Cette localisation utilise un système GPS qui permet de connaître à tout moment la position de l'utilisateur permettant ainsi d'intervenir en cas d'urgence.

4.3 Surveillance d'activité :

La surveillance d'une activité est le processus continu d'une série d'observation des actions et les objectifs d'un ou plusieurs agents dans des différents environnements, en impliquant la collecte de données via les variétés de capteurs .

Les informations recueillies à la suite du suivi seront utilisées pour arriver à une analyse des progrès, afin d'aboutir à des résultats permettant la réalisation d'un objectif tel que la reconnaissance (de plan, d'objectif, d'intention, ou de comportement humain).

- **Dans ce mémoire, nous nous intéresserons principalement à la Surveillance d'activité.**

5. La reconnaissance des activités humaines (RAH) :

La reconnaissance d'activité (HAR) a été un sujet de recherche très actif au cours des deux dernières décennies notamment pour ses applications dans divers domaines tels que la santé, la télésurveillance, les jeux, la sécurité et la surveillance, et l'interaction homme-machine.

5.1 Notions et Objectifs des 'RAH' :

La reconnaissance d'activité peut être définie comme la capacité à reconnaître/détecter et d'interpréter les mouvements et les actions des personnes, en cours sur la base des informations reçues de différents capteurs donner une idée sur leurs comportements voire leurs intentions.

Ces capteurs peuvent être des caméras, capteurs portables ou des capteurs fixés aux objets du quotidien ou déployés dans l'environnement.

5.2 Domaines d'applications :

La reconnaissance d'activités est présente dans plusieurs domaines, nous citons :

- **Le secteur du sport** : certains produits sportifs tels que les chaussures de course Philips Direct Life ou NIKE + intègrent des capteurs de mouvement et offrent aux athlètes amateurs et professionnels un retour d'informations sur leurs performances^[24]
- **Le divertissement**: à travers les consoles de jeux de la Nintendo Wii et le Microsoft Kinect qui s'appuient sur la reconnaissance des gestes ou des mouvements du corps entier pour changer totalement l'expérience de jeu^[25].
- **La santé**: la reconnaissance d'activités permet aux médecins de surveiller le comportement et les activités des patients (par exemple, personnes âgées et/ou atteintes d'Alzheimer) même à distance et d'agir rapidement en cas d'urgence ou de danger. Elle leur permet aussi de mieux connaître le patient, de diagnostiquer ses pathologies, et de développer de nouvelles stratégies d'intervention en matière de prévention et d'accompagnement. Par exemple, les patients atteints de diabète, d'obésité ou de maladie cardiaque doivent souvent suivre une série d'exercices bien définie dans le cadre de leur traitement^[13]. Par conséquent, reconnaître leurs activités quotidiennes telles que la marche, la course ou le repos devient très utile pour fournir des informations au personnel de santé sur le comportement du patient^[14]. De même, les personnes âgées pourraient être surveillés pour détecter des activités anormales et ainsi prévenir des conséquences indésirables (une chute, par exemple)^{[15][16]}.
- **Le militaire**: Dans les scénarios tactiques et militaires, des informations précises sur les activités des soldats ainsi que leurs emplacements et leurs conditions de santé sont très bénéfiques pour leur performance et leur sécurité^[17]. De telles informations sont également utiles pour soutenir la prise de décision dans les environnements de combat et d'entraînement.

5.3 Les différentes approches de RAH :

Pour saisir ces activités, différentes approches ont été adoptées.

Ces approches peuvent être classées en deux grandes catégories : celles basées sur la vision et celles basées sur les capteurs ^[26], comme le montre la figure 14.

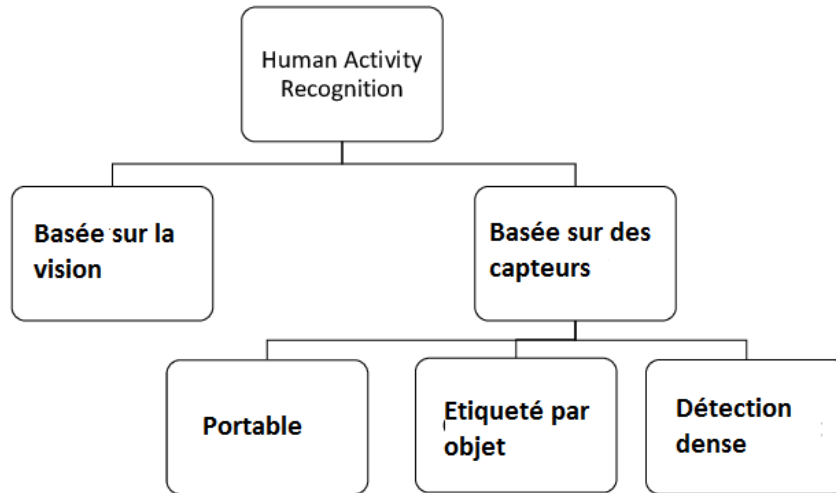


Figure 24 Classification des approches de reconnaissance de l'activité humaine

❖ Basée sur la vision:

Les approches pionnières dans ce domaine sont les approches fondées sur la vision, où une caméra est utilisée pour capturer les informations sur les activités de la personne.

Cette méthode se présente sous forme de détection des mouvements à partir d'images ou flux vidéo. En appliquant les techniques de vision par ordinateur sur ces données saisies, différentes activités peuvent être reconnues.

Bien que les techniques basées sur la vision par ordinateur soient faciles à utiliser et peuvent donner de bons résultats, il y a de nombreuses problématiques liées à cette approche.

Le respect de la vie privée est la principale préoccupation. Un autre problème avec cette approche est une dépendance à une source de lumière. En effet, les caméras traditionnelles ne parviennent pas à travailler s'il n'y a pas de lumière appropriée (par exemple, la nuit).

La reconnaissance des activités basée sur la vision a trouvé de nombreuses applications telles que l'interaction homme-machine, la conception d'interfaces utilisateur, l'apprentissage par robot et la surveillance, entre autres de nombreuses enquêtes ont été rédigés pour des approches basées sur la vision.

❖ Basée sur les capteurs:

En raison du faible coût et des progrès de la technologie des capteurs, la plupart des recherches dans le domaine du HAR ont évolué vers une approche basée sur les capteurs.

Les solutions basées sur des capteurs peuvent être divisé en trois grandes catégories sur la base des déploiements, qui sont:

- i) portables,
- ii) étiquetés aux objets (liés à l'appareil),
- iii) détection dense (étiquetés par l'environnement / sans périphérique).

Dans une approche portable, l'utilisateur doit porter les capteurs sur lui lorsqu'il effectue une activité. De nombreuses études ont été faites sur la reconnaissance des activités en utilisant capteurs portable, mais le problème majeur avec ce type d'approche est qu'il n'est parfois pas possible de porter un capteur sur soi. Par exemple, dans le cas des personnes âgées ou des patients, ils peuvent oublier de porter, ils résistent à porter les étiquettes du tout.

Pour des solutions qui utilisent une approche par objet, des capteurs sont attachés à des objets d'usage quotidien. Sur la base de l'interaction d'un utilisateur avec ces objets, différentes activités sont reconnues. Il s'agit d'une approche liée au périphérique, c'est-à-dire que les utilisateurs doivent utiliser des objets spécifiques (objets marqués) uniquement. Comme l'approche portable, cette approche peut également ne pas être réalisable tout le temps car elle limite les utilisateurs à utiliser seulement des objets marqués.

Au cours des dernières années, les chercheurs se sont concentrés sur une approche sans appareil (détection dense) dans laquelle les utilisateurs ne transportent pas d'appareil avec eux. L'idée est de déployer capteurs dans l'environnement (l'installation dans laquelle l'activité est en cours) et lorsqu'une personne effectue une activité, les données seront capturées par ces capteurs, qui peuvent ensuite être utilisés pour la reconnaissance d'activité. L'approche sans appareil est plus pratique car elle ne nécessite pas que l'utilisateur transporte avec lui un appareil spécifique lorsqu'il effectue ses activités. Le principal inconvénient de cette approche

réside dans les interférences provoquées par l'environnement car ce dernier peut perturber les données capturées par les capteurs ce qui peut induire du bruit.

○ Les technologies utilisées :

La technologie des capteurs a joué un rôle central dans l'évolution des HAR.

En effet, afin de récolter les différentes données nécessaires à la reconnaissance des activités, de nos jours, de nombreux types de puces bon marché et capteurs portable sont disponibles, avec la capacité de détecter et de communiquer les informations à l'aide de réseaux sans fil.

Dans cette section, nous fournissons les détails sur certaines des technologies qui sont utilisés pour la reconnaissance de l'activité humaine comme indiqué dans les figures suivantes.

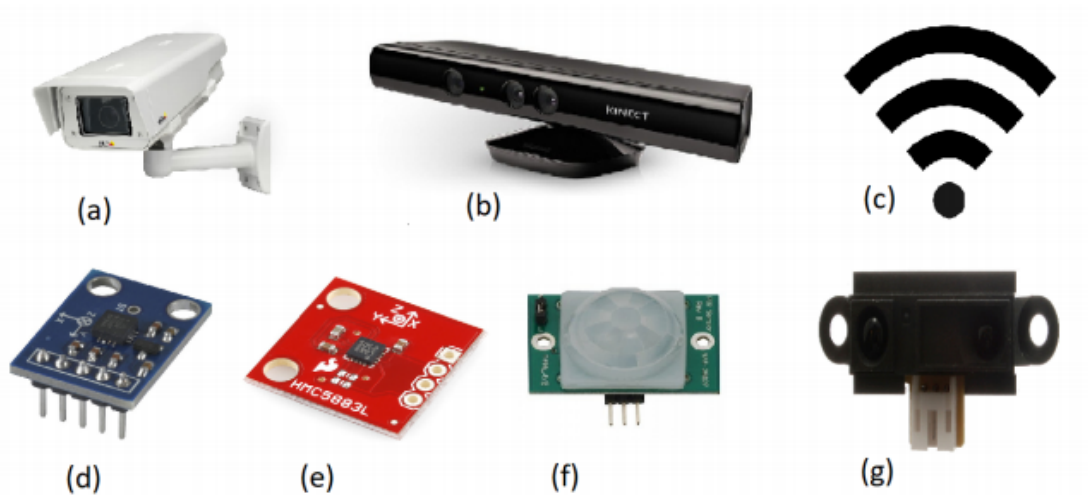


Figure 25 . Quelques technologies utilisées pour les HAR: (a) Caméra de surveillance (b) Caméra de profondeur (c) Wi-Fi (d) Accéléromètre (e) Magnétomètre (f) Capteur de mouvement (g) Capteur de proximité.

✓ **Caméra de surveillance :**

La manière la plus élémentaire et traditionnelle de reconnaître une activité est d'installer des caméras de surveillance dans l'environnement et de surveiller les activités humaines.

La surveillance peut être effectuée par un humain (une personne qui regarde les vidéos et les images provenant de caméras) ou de façon automatique. Différentes techniques de vision par ordinateur ont été développées afin de traiter et analyser les données (vidéos et images) de la caméra et ainsi reconnaître automatiquement les activités.

Les chercheurs ont tenté un certain nombre de méthodes telles que le flux optique, le filtrage de Kalman, les modèles de Markov cachés...etc., sous différentes modalités telles que la caméra unique, la stéréo et l'infrarouge. De plus, les chercheurs ont examiné plusieurs aspects sur ce sujet, notamment le suivi d'un seul piéton, le suivi de groupe et la détection d'objets tombés.

✓ ***Caméra de profondeur :***

Un des problèmes avec les caméras traditionnelles est la dépendance à la lumière, c'est-à-dire qu'ils ne peuvent pas travailler dans l'obscurité. Le développement des caméras de profondeur telles que Kinect a résolu ce problème parce qu'il peut fonctionner dans l'obscurité totale. Différents flux de données peuvent être obtenus à partir de Kinect tels que RVB, la profondeur et l'audio^[27].

Les caméras de profondeur ajoutent une dimension supplémentaire, c'est-à-dire la profondeur que la caméra 2d normale ne fournit pas. Les informations sensorielles de ces caméras de profondeur ont été utilisées pour générer un modèle squelette en temps réel d'humains avec différentes positions corporelles. Ces informations sur le squelette fournissent des informations significatives que les chercheurs ont utilisées pour modéliser les activités humaines qui sont entraînées et utilisées plus tard pour reconnaître des activités inconnues.

✓ ***WI-FI :***

Au cours de la dernière décennie, il y a eu un changement de paradigme dans la recherche sur les HAR, passant des approches liées aux appareils (device-bound) aux approches sans appareils (device-free). Les chercheurs ont étudié les propriétés des réseaux sans fil, tels que Channel State Information (CSI) et ont commencé à l'utiliser pour la reconnaissance d'activité ^[28]. De nombreuses solutions ont été proposées pour la localisation, le suivi, la détection des chutes...etc., en utilisant le Wi-Fi.

La reconnaissance d'activité via Wi-Fi se base sur le principe que lorsque le trajet du signal lorsqu'il traverse le corps humain pendant la transmission est modifié (réflexion,

diffraction et la diffusion du signal). En analysant ainsi les signaux résultants, nous pouvons alors déduire les mouvements de la personne.

Un avantage majeur du Wi-Fi est qu'il est omniprésent, discret et les utilisateurs ne sont pas tenus de transporter un appareil avec eux.

✓ *Capteur :*

Au XXI^e siècle, d'importante avancées ont été faites dans le domaine des capteurs, et de nombreux types de capteurs ont été produits. Certains de ces capteurs sont largement utilisés dans la recherche pour la reconnaissance des activités. Nous pouvons citer:

- **Accéléromètre :** Un accéléromètre est un appareil électromécanique utilisé pour mesurer l'accélération dans plusieurs directions. (x, y et z). L'accéléromètre est largement utilisé dans les solutions pour la reconnaissance des gestes, la reconnaissance de la posture, la chute détection, suivi, vie assistée ambiante, activités quotidiennes... etc.
- **Magnétomètre:** Un magnétomètre est un appareil qui sert à mesurer selon les cas l'intensité ou la direction d'un champ magnétique, ou l'aimantation d'un échantillon.

Ce capteur est utilisé dans divers domaines de reconnaissance d'activité (par exemple, la reconnaissance des gestes) en raison de sa capacité à détecter les changements dans le champ magnétique causés par l'activité humaine.

- **Capteur de mouvement:** les capteurs de mouvement sont utilisés pour détecter mouvement ou la présence d'un sujet dans une zone particulière.
- **Capteur de proximité:** il s'agit d'un capteur électronique qui peut détecter la présence d'objets à proximité sans nécessiter aucun contact physique.

✓ *RFID :*

La technologie d'identification par radiofréquence a connu une croissance exponentielle au cours de la dernière décennie. Développé à l'origine à des fins militaires pour permettre de différencier les avions ennemis des alliés. Elle est, désormais, utilisée dans de nombreux secteurs.

Les radio-étiquettes sont de petits objets, tels que des étiquettes autoadhésives, qui peuvent être collés ou incorporés dans des objets ou produits et même implantés dans des organismes vivants (animaux, corps humain 3), elles comprennent une antenne associée à une puce électronique contenant un identifiant et éventuellement des données complémentaires ; qui leur permet de recevoir et de répondre aux requêtes radio émises depuis l'émetteur-récepteur.

En raison de la nature passive de la RFID, à faible coût et discrète, cette technologie a été adoptée dans divers domaines. La RFID est maintenant largement utilisée dans la recherche sur la reconnaissance de l'activité humaine, les chercheurs l'utilisent pour la reconnaissance de la posture, la reconnaissance des gestes, le suivi, la localisation, le comportement ...etc.

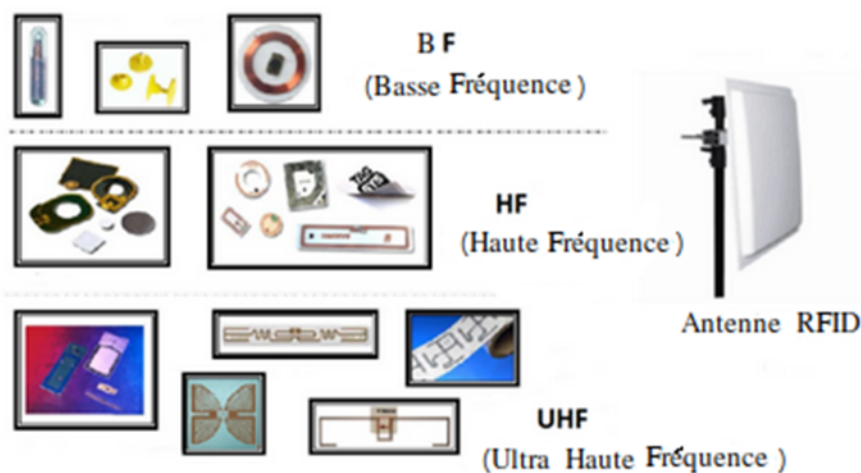


Figure 26 Technologie RFID (balise et antenne).

- **Pour notre étude, nous nous sommes intéressé plus précisément aux capteurs présent dans les smartphones (téléphones intelligents).**

6. Les capteurs utilisés dans les téléphones intelligents

La reconnaissance du mouvement humain à base de capteurs installés sur des terminaux mobiles a reçu beaucoup d'attention de la part des chercheurs ces dernières années. De ce fait, plusieurs capteurs ont été adaptés pour les mouvements à grande échelle pour améliorer la qualité de vie des personnes. Ces capteurs présentent deux avantages majeurs :

- La portabilité par rapport aux méthodes basées sur la vision par ordinateur et de télésurveillance.

- Leurs faibles coûts couplés au fait qu'ils soient non invasives pour les utilisateurs (car très léger et peu encombrant), ce qui permet de les appliquer dans divers domaines, tels que la réadaptation médicale et des jeux interactifs.

La plupart des appareils mobiles, téléphones intelligents et tablettes sont munis de capteurs qui permettent de convertir des valeurs analogiques en des valeurs numériques afin de pouvoir lire les données collectées par un appareil mobile. Il existe plusieurs types de capteurs.

Cependant, dans cette section nous nous sommes limités à ceux qui sont fréquemment intégrés dans les appareils mobiles à savoir, l'accéléromètre, le gyroscope. Ces deux capteurs permettent de détecter le mouvement et la rotation. Les données enregistrées par ces capteurs sont tridimensionnelles, c'est-à-dire selon les trois axes X, Y et Z telles qu'illustrées dans la figure suivante.

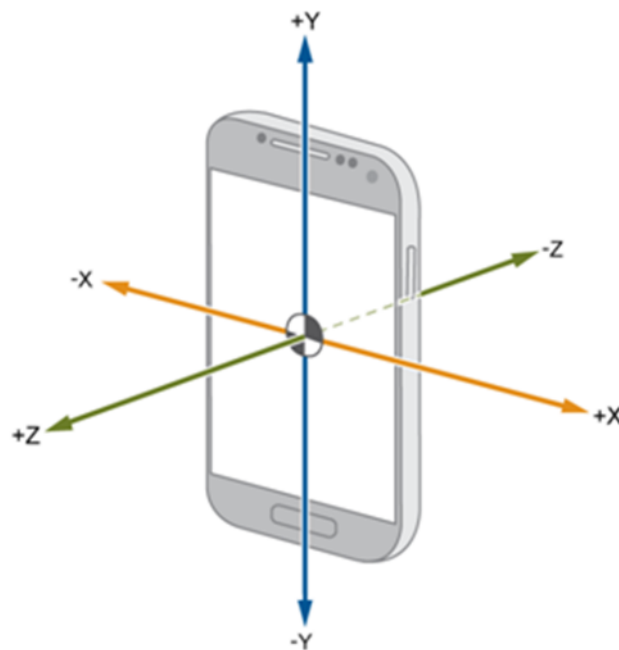


Figure 27 Référentiels utilisés dans les terminaux mobiles.

6.1 Accéléromètre

L'accéléromètre permet de mesurer l'accélération linéaire selon trois axes orthogonaux (X, Y et Z) ce qui permet de déduire les changements de vitesse et les changements de position.

L'accéléromètre permet aussi de déterminer l'orientation de l'écran d'un téléphone intelligent (paysage et portrait) grâce à la force de gravité générée par la terre.

L'accélération linéaire est calculée en utilisant la formule suivante :

$$A = \sqrt{A_x^2 + A_y^2 + A_z^2}$$

(1)

Où :

- A est la valeur de l'accélération linéaire; elle est exprimée en m/s^2 .
- Acc_x , Acc_y et Acc_z désignent les valeurs des composantes de l'accélération selon les trois axes X, Y et Z.

La figure 11 montre des valeurs enregistrées par un accéléromètre selon les trois axes (Composante X, composante Y, composante Z).



Figure 28 Signaux d'accélération collectés à partir de terminaux mobiles

6.2 Gyroscope

Le gyroscope permet de mesurer la position angulaire et donc la vitesse de rotation selon les trois axes (X, Y et Z). Dans le cas des appareils mobiles, le gyroscope est calibré pour donner une lecture de zéro lorsque le dispositif est maintenu sur une surface horizontale.

Tout changement dans l'orientation du dispositif est mesuré par le capteur gyroscopique. Un exemple d'utilisation du gyroscope est, lorsqu'on a une image visualisée dans un plan horizontal et que l'on change le plan vers celui correspondant à la verticale on remarque que l'image s'oriente et s'affiche dans le plan vertical. La figure 12 illustre un exemple de données enregistrées par le

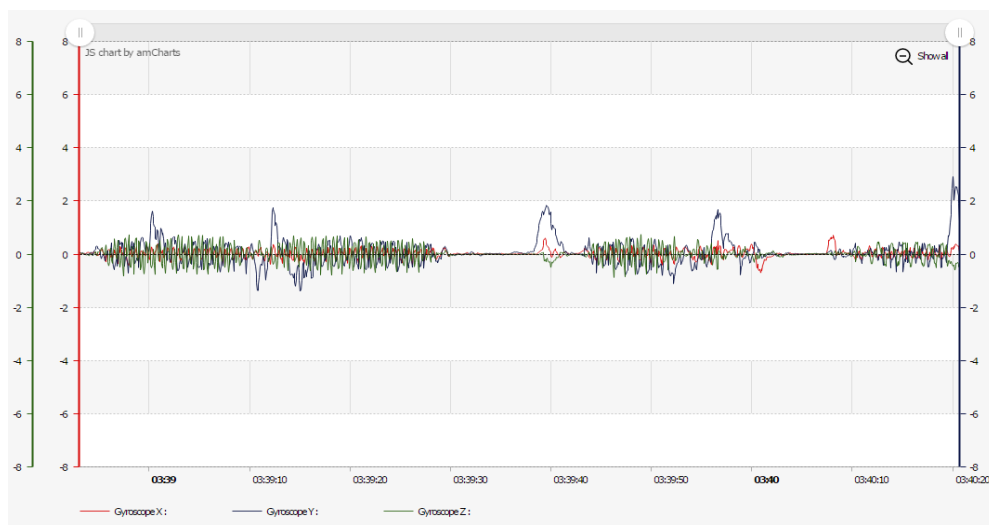


Figure 29 Données collectées à partir d'un gyroscope

6.3 Capteur de gravité

Le capteur de la gravité fonctionne selon le même principe que celui de l'accéléromètre sauf qu'il permet de mesurer, non pas l'accélération linéaire, mais la gravité par rapport à la terre.

La figure 13 montre un exemple des valeurs enregistrées par un capteur de gravité.

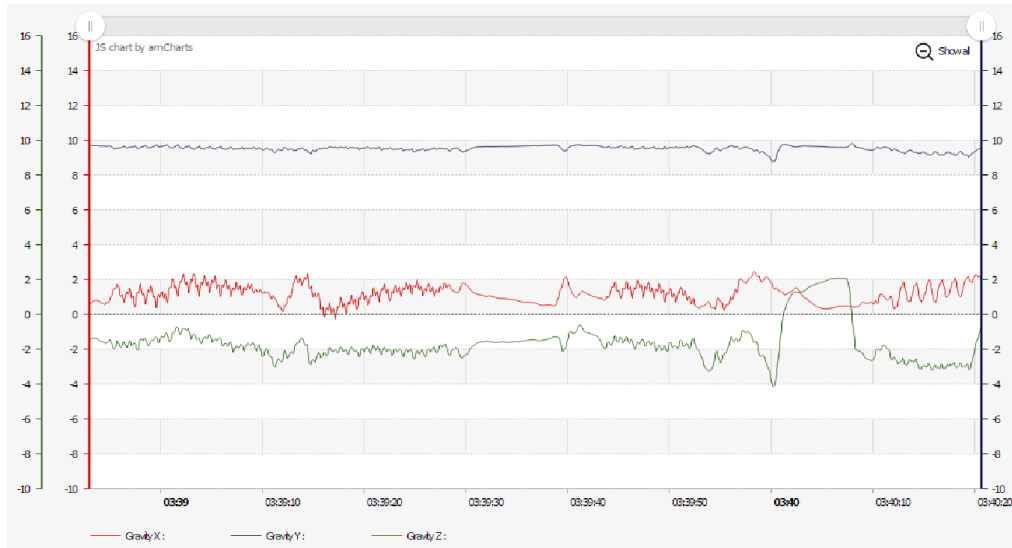


Figure 30 Données collectées par le capteur de gravité.

7. Méthodes de Classification :

7.1 Principe de la classification de formes

De nos jours, le volume des données collectées ne cesse d'augmenter exponentiellement et le critère du temps devient important. De ce fait, les techniques de classification automatique de données sont devenues très importantes pour faciliter l'exploitation de ces données ; plusieurs recherches et études ont été réalisées dans le domaine de classification de forme pour permettre de mieux comprendre et d'exploiter les données surtout dans le cas où ces données sont de volume important et que les critères de sélection sont moins visibles.

L'intégration de ces techniques dans la reconnaissance des activités physiques est devenue essentielle pour améliorer la qualité de vie humaine, et ce, en utilisant des technologies mobiles telles que les téléphones intelligents et les textiles intelligents qui contiennent des capteurs permettant de définir le type d'activité effectuée par une personne.

Le processus de la classification de formes commence par une première étape de prétraitement pour éliminer les différents bruits des données enregistrées. La deuxième étape consiste à définir les différentes caractéristiques à utiliser lors de la classification. Parmi ces caractéristiques, on trouve la moyenne, l'écart-type, la médiane (Yi He et Ye Li ^[29]; Oscar D et

al ^[30]). L'étape suivante consiste à développer le classificateur pour reconnaître les différentes classes et enfin la dernière étape permet d'évaluer et valider le système de classification.

Généralement, le diagramme en bloc d'un système de classification de formes suit la figure suivante qui illustre les étapes du processus de classification.

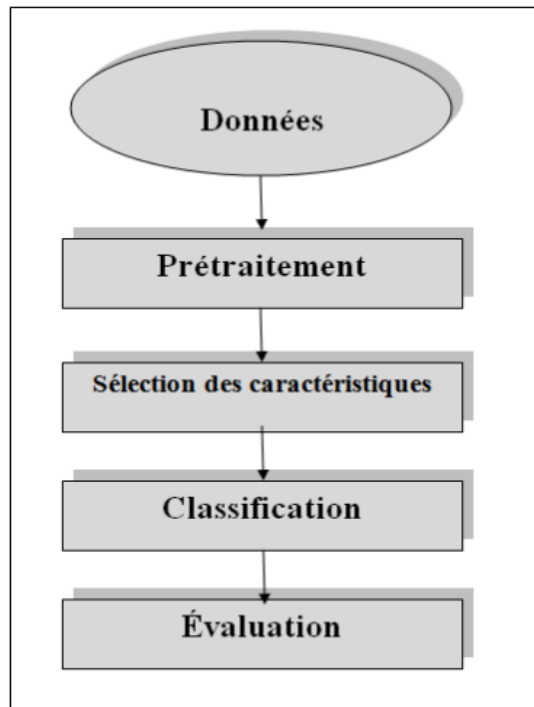


Figure 31 Processus de classification

A. Prétraitement :

Le processus de classification est précédé par une phase d'acquisition des données brutes par l'intermédiaire de différent capteur, puis par un prétraitement, qui consiste à réduire les bruits causés par les conditions d'acquisition des données et d'éliminer les distorsions et les données inutiles ^[31]. Le prétraitement consiste généralement à faire un filtrage pour éliminer le bruit et à normaliser les données afin d'uniformiser la dimension ou l'échelle.

Plusieurs techniques de filtrage existent dans la littérature parmi lesquels on retrouve, le filtre de Kalman, qui est un filtre à réponse impulsionnelle infinie. Ce filtre permet d'estimer l'état d'un processus tout en minimisant la moyenne de l'erreur quadratique. Le filtre de Kalman est très puissant parce qu'il prend en considération les estimations des états passés, actuels et

futurs. Ce filtre est un outil important dans l'étude des systèmes dynamiques parce qu'il offre un cadre fondamental à l'estimation d'états.

Cependant, il présente certaines limitations. En effet, il présente des instabilités dues à des écarts de modèle et à des problèmes de calcul en précision finie. De plus, est confronté à des problèmes de divergence. Ces problèmes peuvent être limités en utilisant un filtrage avec une équation qui propage la racine carrée de la matrice de covariance.

B. Extraction et sélection des caractéristiques :

Le but de cette étape est d'identifier les caractéristiques qui permettent de discriminer les différentes classes. Dans le cadre de l'analyse de l'activité physique, Oscar et al ^[30] ont recensé deux types de caractéristiques : les caractéristiques basées sur la fréquence (caractéristiques fréquentielles) et les caractéristiques basées sur le temps (caractéristiques temporelles) telles que la moyenne, la médiane, l'écart-type, l'interquartile ^[29].

D'autres types de caractéristiques qui permettent une représentation globale peuvent être utilisés. En l'occurrence, l'analyse en composantes principales (ACP) est une méthode d'analyse de données très connue en statistique et dans les sciences expérimentales. Elle consiste à rechercher les directions de l'espace qui représentent le mieux les corrélations dans un ensemble de données. Ceci a pour objectifs de réduire la dimension des caractéristiques et de les visualiser afin d'analyser les corrélations entre ces données.

C. Classification :

La classification de formes est l'opération qui consiste à grouper formellement des éléments de configuration, des incidents, des problèmes ou des changements, en fonction de leur type» (Office québécois de la langue française, 2005). En d'autres termes, la classification de formes consiste à affecter un objet à une classe correspondante en se basant sur leurs caractéristiques.

Les méthodes de classification de formes peuvent être réparties en deux types. La classification supervisée et la classification non supervisée. Ces techniques seront détaillées à la section 8 de ce chapitre.

D. Évaluation et validation d'un classificateur :

L'évaluation d'un système de classification se base généralement sur un partage de la base de données en trois domaines aussi appelés sous-bases :

- Une **sous-base d'apprentissage** pour la conception du système de classification ;
- Une **sous-base de test** pour sélectionner l'architecture du système de classification ;
- Une **sous-base de validation** utilisée pour valider la pertinence de l'architecture choisie.

La distinction entre les deux premières bases est nécessaire pour ne pas biaiser les tests de classification en testant sur des échantillons ayant servi pour la conception. La 3e sous-base permet d'obtenir une validation indépendante des critères de sélection. Dans certains cas, la taille de la base de données ne permet pas d'avoir un partage en trois sous-bases ; dans ce cas on utilise uniquement un partage en deux : une sous-base d'apprentissage et une sous-base de tests.

Il existe différents types d'algorithmes pour réaliser le partage des données en données d'apprentissage et données de test. Parmi ces méthodes on trouve la validation par Kfold.

- **Validation par K-fold:**

Le principe de cette méthode de validation consiste à diviser l'échantillon original en K échantillons de même taille. Puis on prend un échantillon pour procéder au test. On répète le processus jusqu'à atteindre les K échantillons. En d'autres termes, on divise l'échantillon original en K échantillons, puis on sélectionne un des K échantillons comme ensemble de tests et les (K-1) autres échantillons constitueront l'ensemble d'apprentissages pour la conception du système de classification. À la fin, on prend la moyenne des résultats de test pour avoir un seul résultat.

- **Critères de validation :**

L'évaluation d'un système de classification se base généralement sur la mesure du taux de classification. Pour avoir une meilleure interprétation des résultats, nous utilisons la matrice de confusion. Il s'agit d'une matrice carrée de taille $[W \times W]$ où W représente le nombre de classes. Chaque colonne de la matrice représente le nombre d'occurrences d'une classe prédite, alors que chaque ligne représente le nombre d'occurrences d'une classe réelle (ou de référence).

La matrice de confusion offre la possibilité d'analyse du taux de bonne classification (ou d'erreur) par classe.

Les valeurs de la diagonale de la matrice représentent le nombre de formes correctement classifiées. La somme des valeurs par ligne correspond au nombre d'échantillons de test par classe. Le taux de classification par classe est donné par la valeur à la diagonale divisée par la somme des valeurs par ligne.

- ❖ **Cas de deux classes** : L'évaluation d'un système de classification binaire se base sur le calcul du taux de classification, de la sensibilité et de la spécificité. Le taux de classification est calculé par l'équation :

$$T = \frac{N1}{N} \quad (2)$$

Où :

T désigne le taux de classification,

N1 est le nombre d'échantillons bien classifiés et N le nombre total des échantillons.

En utilisant cette relation, nous pouvons déduire le taux d'erreur, TE, par la relation suivante :

$$TE = 1 - T \quad (3)$$

Dans le cas de deux classes C1 et C2, la matrice de confusion est illustrée dans le tableau suivant :

Classe réelle	Classe prédite	
	C1	C2
C1	VP	FN
C2	FP	VN

Tableau 1 Matrice de confusion dans le cas binaire

Avec :

- VP (Vrai positif) : Éléments de la classe C1 correctement classifiés;
- VN (Vrai négatif) : Élément de la classe C2 correctement classifié;
- FP (Faux positif) : Éléments de la classe C2 classifiés comme étant de la classe C1;
- FN (Faux négatif) : Éléments de la classe C1 classifiée comme étant de la classe C2.

Cette matrice permet de déduire les paramètres suivants pour l'évaluation d'un classificateur binaire:

$$\begin{aligned}\text{Précision} &= \frac{(\text{VP} + \text{VN})}{N} \\ \text{Sensibilité} &= \frac{\text{VP}}{(\text{VP} + \text{FN})} \\ \text{Spécificité} &= \frac{\text{VN}}{(\text{VN} + \text{FP})}\end{aligned}\tag{4}$$

8. Techniques de classification :

Nous distinguons deux catégories de techniques de classification : les techniques supervisées et les techniques non supervisées

8.1 La classification supervisée :

Dans le cas de classification supervisée (en anglais supervised learning), le système observe des couples de types entrée-sortie et apprend une fonction qui permet d'aboutir à la sortie à partir de l'entrée. Cette phase est appelée phase d'apprentissage ou d'entraînement. Les données de sortie correspondantes sont appelées classes.

Il existe différents algorithmes pour la classification supervisée. Parmi ces algorithmes on cite, le K plus proche voisin (KNN : K-Nearest Neighbor), les arbres de décision et les machines à vecteur de support (SVM).

- **K plus proche (KNN) :**

Le principe de l'algorithme KNN est de classer les exemples dont la classe est inconnue en se basant sur leur similarité avec un ensemble d'échantillons de référence.

La similarité est déterminée par un calcul de distance. La distance euclidienne décrite dans l'équation suivante est celle la plus utilisée :

$$d(X, u) = \sqrt{\sum_{i=1}^n (x_i - u_i)^2} \quad (5)$$

Où

d est la distance euclidienne, $X = (x_1, x_2, \dots, x_n)$ est l'élément à classifier .

u est un élément de la base de référence.

n étant la dimension du vecteur de caractéristiques.

La figure 14 illustre le principe du KNN; pour déterminer la classe d'un échantillon X à l'aide du classificateur KNN. Nous cherchons les plus proches voisins par le calcul de la distance euclidienne entre X et tous les éléments des deux classes $Gr1$ et $Gr2$. L'élément X est affecté à la classe où le nombre des voisins trouvés est majoritaire.

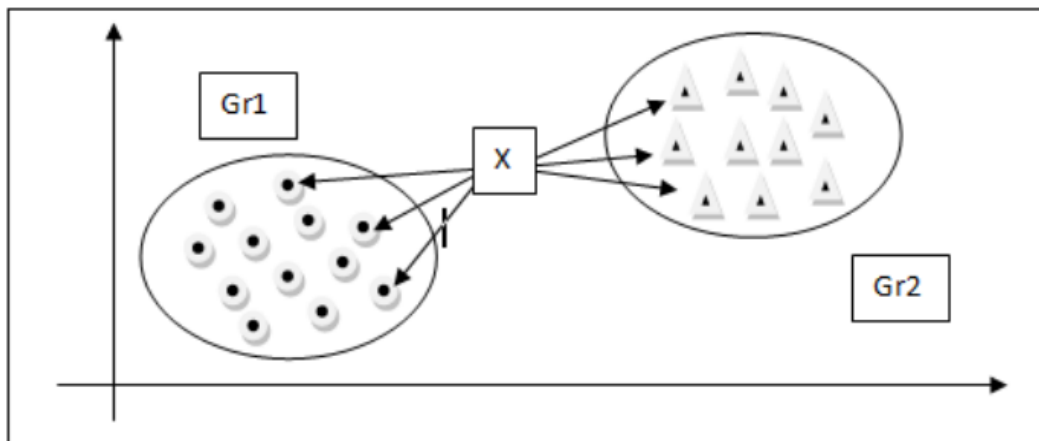


Figure 32 Distance euclidienne entre X et les éléments des deux classes $Gr1$ et $Gr2$

L'algorithme KNN présente plusieurs avantages tels que la facilité de son intégration et de sa mise en œuvre et sa robustesse. Cependant, il a des inconvénients tels que le temps de classification qui est important puisqu'il faut évaluer la distance entre l'élément à classer et les éléments de références de toutes les classes à chaque fois. L'algorithme KNN est, de plus, sensible aux attributs non pertinents et corrélés.

- **Les arbres de décision :**

L'apprentissage automatique par arbres de décisions permet de représenter graphiquement le système de classification de façon à rendre la compréhension de son fonctionnement plus facile pour l'utilisateur. Les arbres de décision permettent de classer un ensemble d'individus en se basant sur la pertinence des attributs par rapport à leurs classes. Les arbres de décision présentent des avantages, à savoir, la facilité de mise en œuvre, la facilité de prise de décision et la simplification des décisions complexes.

- **Les machines à vecteurs de support (SVM) :**

La classification basée sur les machines à vecteurs de support est relativement récente; elle a été créée par Vapnik Vladimir en 1995. Le principe de base des SVM consiste à maximiser la distance entre deux classes pour résoudre le problème de discrimination avec un séparateur linéaire ou non linéaire ^[32].

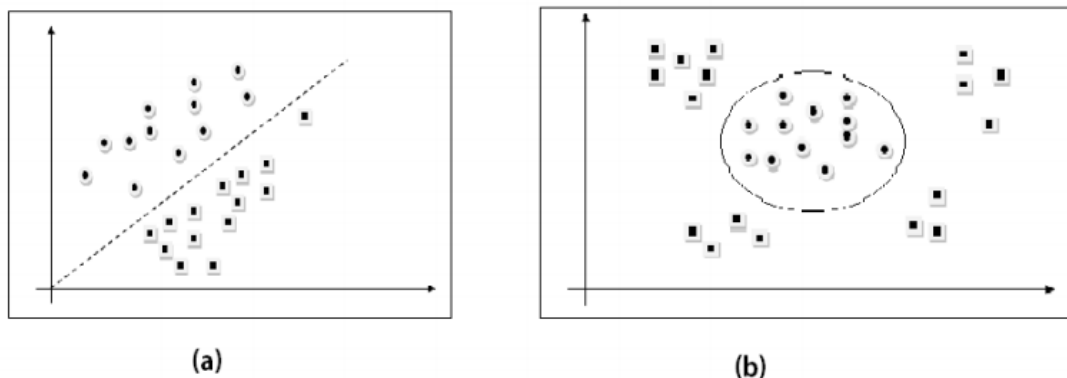


Figure 33 (a) : Discrimination à deux classes

-(a) : Séparateur linéaire - (b) : Séparateur non linéaire

- **Réseau de neurones artificiels:**

La conception des réseaux de neurones artificiels s'appuie sur la structure des neurones biologiques du cerveau humain. Les réseaux de neurones artificiels peuvent être décrits comme des systèmes composés d'au moins deux couches de neurones - une couche d'entrée et une couche de sortie - et comprenant généralement des couches intermédiaires (« *hiddenlayers* »).

Au sein d'un réseau de neurones artificiels, le traitement de l'information suit toujours la même séquence : les informations sont transmises sous la forme de signaux aux neurones de la couche d'entrée, où elles sont traitées. **À chaque neurone est attribué un « poids »** particulier, et donc une importance différente. Associé à la fonction dite de transfert, le poids permet de déterminer quelles informations peuvent entrer dans le système.

À l'étape suivante, une **fonction dite d'activation** associée à une **valeur seuil** calculent et pondèrent la valeur de sortie du neurone. En fonction de cette valeur, un nombre plus ou moins grand de neurones sont connectés et activés.

Cette connexion et cette pondération **dessinent un algorithme** qui fait correspondre un résultat à chaque entrée. Chaque nouvelle itération permet d'ajuster la pondération et donc l'algorithme de façon à ce que le réseau donne à chaque fois un résultat plus précis et fiable.

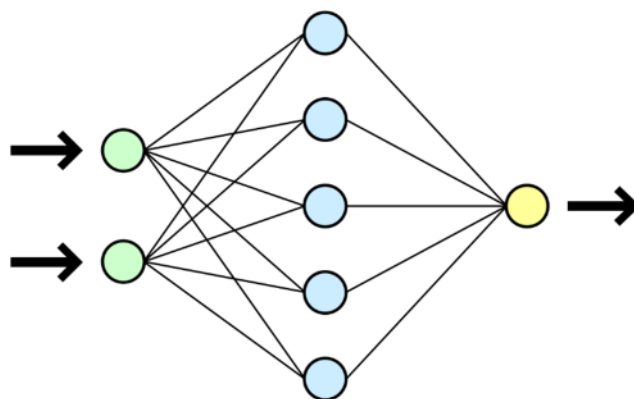


Figure 34. Vue simplifiée d'un réseau artificiel de neurones

Pour veiller à ce que les connexions au sein d'un réseau de neurones artificiels soient correctement établies, il faut au préalable procéder à son « entraînement ».

8.2 La classification non supervisée :

Contrairement à la classification supervisée, la classification non supervisée se base sur des algorithmes qui permettent de classifier en regroupant les individus similaires entre eux sans passer par la phase d'apprentissage. Parmi ces algorithmes on trouve le K moyennes (K-means) et celui de carte Kohonen (réseaux de neurones).

- **K-moyennes :**

Créé par Queen en 1997, l'algorithme K-moyennes permet la minimisation alternée en regroupant en K classes un ensemble d'individus sachant que K est fixé par l'utilisateur. Le choix de K n'est pas universel, plus on augmente la valeur de K plus la distorsion diminue^[33].

- **Carte de Kohonen :**

Il s'agit d'un algorithme original défini par TeuvoKohonen aux années 80 se basant sur la notion de voisinage entre les classes et l'appartenance des observations voisines, dans une dimension P, dans une même classe ou dans des classes voisines^[34].

9. Classification de l'activité physique :

Plusieurs études ont appréhendé la classification automatique de l'activité physique. La différence entre ces méthodes se situe au niveau des capteurs utilisés, des caractéristiques extraites et des techniques de classification utilisées. Le tableau de synthèse ci-dessous, dresse l'état de l'art des systèmes de classification de l'activité physique, se base sur l'étude de Robertas Damasevicius et Al^[35] dans laquelle il a résumé les différentes méthodes de classification ainsi que les caractéristiques utilisées.

D'après le tableau 2, nous constatons que, la majorité des études se sont basées sur l'accéléromètre afin de se servir des valeurs des trois composantes d'accélérations et/ou de l'accélération linéaire. Dans ces études, les taux de classification obtenus varient entre 64,5% et 99,1% selon le nombre d'activités physiques à classifier et le type de classificateur. Nous remarquons également une variété de caractéristiques extraites que ce soit dans le domaine temporel ou le domaine fréquentiel. Les meilleures performances en termes de taux de bonne classification obtenu sont ceux de l'étude de Mannini et Sabatini^[36]. Dans cette étude, les données d'accélération ont été utilisées pour classifier 21 activités physiques en utilisant le

modèle de Markov après l'extraction des caractéristiques basée sur l'énergie, l'entropie et les coefficients de corrélation.

D'autres études ont permis de classifier, avec un taux de classification important, les activités physiques en se basant sur les données de l'accéléromètre combinées avec d'autres capteurs. Il est à noter que les capteurs utilisés dans la majorité de ces études sont des capteurs adhésifs installés en contact direct avec les participants pour une meilleure qualité de collecte des données.

Cet aspect est contraignant pour les utilisateurs et rend l'applicabilité des méthodes développées contraignante. Ainsi, nous proposons d'investiguer des capteurs intégrés dans des téléphones intelligents installés dans la poche de l'utilisateur ou dans une ceinture dédiée aussi fréquemment utilisée par les utilisateurs.

Auteur	Activités	Données des capteurs	Caractéristiques	Sélection de caractéristiques	Méthode de classification	Taux
Atallah et al. [26]	S'allonger, cuisiner, manger et boire, socialiser, lecture, s'habiller, marcher sur tapis roulant, marcher, passer l'aspirateur, essuyer les tables, courir, cyclisme, s'asseoir et se lever, se coucher et se lever.	Accélération	Entropie moyenne sur 3 axes, fréquence principale FFT (En moyenne) sur 3 axes, énergie en 0.2Hz fenêtre centrée autour de la fréquence principale sur totale énergie FFT (moyenne 3-axes), moyenne de covariance croisée entre tous les 2 axes	Relief, Simba et MRMR	KNN, classificateur Baiesien	90%
Bayat et al. [27]	Courir, marche lente, marche rapide, danse aérobique, monter les escaliers, descendre les escaliers	Triaxial Accéléromètre	Moyenne le long de l'axe z, MinMax, STD, et RMS Am, APF le long de l'axe x, l'axe y et l'axe z, VarAPF, STD le long de l'axe x, l'axe y, et l'axe des z, le long de RMS axe des x, l'axe y et l'axe z, corrélation entre l'axe z et l'axe y et min./max le long de l'axe x, l'axe y, et l'axe z	Regroupement des caractéristiques	Multilayer perceptron, SVM, Random Forest, and Logit Boost	81%–91%
Berchtold et al. [28]	Debout, assis, couché, marcher, monter les escaliers, le cyclisme, et étant stationnaire	Accéléromètre	Variance, moyenne	Aucune	Fuzzy inference	97.3%
Capela et al. [29]	Assis, debout, s'allonger, ramper vers le haut et le bas, monter et descendre les escaliers, transition entre les activités	Accélération linéaire, gravité, capteur de vitesse	Moyenne, écart-type, aplatissement, mouvement moyenne, matrice de covariance, dissymétrie, taux zéro croix, et le taux moyen croix	Aucune	Naive-Bayes, SVM, arbre de decision j48	97%
Gupta and	Sauter, courir, marcher, assis,	Triaxial	L'énergie, l'entropie, moyenne,	Relief, SFFS	kNN, Naive	98%

Dallas [30]	assis à debout, et debout à genoux	accéléromètre	variance, X-Z-énergie et différence max de l'accélération		Bayes	
Henpraserttae et al. [31]	Assis, coucher, debout, marcher	Accéléromètre	Moyenne, Écart-type	Aucune	Rules and threshold based classification	90%
Hoque and Stankovic [32]	Quitter la maison, utiliser les toilettes, prendre une douche, dormir, la préparation du petit-déjeuner, préparation du dîner, faire collation, boire, utiliser la machine à laver, utiliser la lave-vaisselle	Location sensors (open/closed)	Magnétomètre	Aucune	Custom clustering method	64.5%–89.9%
Iso and Yamazaki [33]	Marcher, courir, monter/descendre les escaliers, marche rapide	Accéléromètre	Composants ondelettes, Périodogrammes et entropie	Aucune	Probabilités bayésiennes	80%
Kose et al. [34]	Marcher, courir, vélo, assis, debout	Accéléromètre	Min., Max., Moyenne, variance, coefficients FFT, et autocorrélation	Aucune	Clustered kNN	95.2%–97.5%
Kwapisz et al [35]	Marche, jogging, monter/descendre les escaliers, assis, debout	Accéléromètre	Moyenne, std. dev., moyenne différence absolue, moyenne accélération résultante, le temps entre les pics et binned distribution	Aucune	Arbre de décision, logistic regression, and MNN	91.7%
Lane et al. [36]	Conduire, stationner, courir, marcher	GPS, Accéléromètre, et microphone	Moyenne, variance	Aucune	Naive-Bayes	85–98%
Lee and Cho [37]	Être debout, marcher, courir, monter/descendre les escaliers, shopping, et prendre le bus	Accéléromètre	valeurs accélération pour les axes x, y et z.	Aucune	Hierarchical HMM	70%–90%

Mannini and Sabatini [38]	Marcher, marcher en transportant des objets, assis & détente, travailler sur ordinateur, debout, manger ou boire, regarder la télévision, la lecture, la course, le vélo, étirage, la force la formation, le lavage, passer l'aspirateur, le pliage blanchisserie, couché et détente, se brosser les dents, monter les escaliers, prendre ascenseur, prendre escalier mécanique	Accélération	composante continue, l'énergie, domaine fréquentiel entropie, et les coefficients de corrélation	SFFS (Pudil algorithm)	Continuous emissions, Hidden Markov Model	99.1%
Mathie et al. [39]	les différents mouvements humains y compris repos, marcher et chutter	Triaxial Accélération sensor	zone intégrée sous la courbe	Aucune	Arbre de décision binaire	97.7% (sensitivity) 98.7% (specificity)
Maurer et al. [40]	Marcher, debout, assis, courir, monter/descendre les escaliers	Multiple sensors	Moyenne, moyenne quadratique, écart-type, variance, moyenne absolue écart, cumulative histogramme, percentiles, l'interquartile, ZCR, MCR	Correlation based Feature Selection (CFS)	Arbre de décision (C4.5 algorithm), k-Nearest Neighbor, Naive-Bayes	80%–92%
Miluzzo et al. [41]	Assis, debout, marcher, courir	Accéléromètre, GPS, audio	DFT, les caractéristiques de la FFT, moyenne, std et le nombre de pics par unité	Aucune	Arbre de décision	79%

Parkka et al. [42]	Allongé, vélo, assis, debout, courir, marcher	GPS, audio, altitude, EKG, accelerometer, compass, humidity, light, temperature, heart rate, pulse, respiratory effort, and skin resistance	Fréquence de pics de haut en bas de l'accélération, la médiane de haut en bas de l'accélération, puissance de pics de l'accélération, la variance l'accélération, la somme de variances de poignet 3D l'accélération et la puissance rapport des bandes de fréquences 1-1.5Hz et 0.2-5Hz mesurée à partir du magnétomètre	Heuristic	Arbre de décision	86%
Saponas et al. [43]	Marcher, jogging	Accéléromètre	124 caractéristiques: Nike + iPod Payload Packet, l'ampleur (Moyenne, std. Dev., Min., max., et min. moins max.), la fréquence (en énergie de chacune des 10 premières composantes de fréquence DFT, l'énergie de chaque bande de 10 fréquences composants, la plus grande composante de fréquence, et indice de la plus grande composante de fréquence)	Aucune	Naive-Bayesian Network	97.4% (with person), 99.48% (cross-person)
Siirtola and Roning [44]	Marcher, courir, vélo, conduire, assis, et debout	Accelerometer	Magnitude, std., moyenne, min., max., percentiles (10, 25, 50, 75 et 90), et la somme et somme des carrés de observations dessus / dessous percentile (5, 10, 25, 75, 90, et 95) d'une amplitude accélération et la somme carrée de X et Z	Aucune	Arbre de décision + kNN/QDA	95%

Sohn et al. [45]	Marcher, conduire	GPS	Corrélation de rang, variance, moyenne, La distance euclidienne au cours d'une Fenêtre de mesures	Aucune	Logistic regression	85%
Yang [46]	Assis, debout, marcher, courir, conduire, vélo	Accelerometer	Moyenne, std., ZCR, 75e percentile, interquartile, spectre barycentre, l'entropie, et corrélation croisée	Aucune	Arbre de décision, Naive-Bayes, kNN, and SVM	90%
Zhu and Sheng [47]	Assis, debout, coucher, marcher, assis à debout, debout à assis, coucher à assis, et assis à coucher	3D acceleration	Moyenne, variance	Aucune	Neural network ensemble	67%–98%

Tableau 2 *Résumé des travaux effectués dans le domaine de classification des activités physiques .Tiré de : Robertas Damasevicius et al [35]*

10. Conclusion

Au cours de ce chapitre, nous avons exposé l'état de l'art relatif à la reconnaissance d'activité humaine. Nous avons commencé, dans un premier temps par définir les différents environnements intelligents existants, leurs caractéristiques ainsi que leurs avantages. Par la suite, nous avons vu ce qu'était un capteur de manière générale, puis plus précisément ceux liées aux humains.

Après avoir introduit les environnements intelligents et les capteurs, nous nous sommes intéressés au domaine de la reconnaissance des activités humaines, où nous avons présenté les points importants, les multiples domaines d'utilisation ainsi que les différentes approches existantes, et cela en s'appuyant sur quelques travaux déjà existants.

Comme nous l'avons déjà abordé, notre projet a pour objectif de développer un système de classification automatique de l'activité physique de l'utilisateur d'un terminal mobile en utilisant les données captées par ce dernier, donc il est nécessaire de savoir quelle méthode est approprié à notre travail, c'est pour cela que nous avons abordé les techniques de classification de formes en général puis plus spécifiquement pour l'activité physique, en passant par la méthode d'élaboration de ces systèmes de classification ,commençant par l'étape d'acquisition des données ,suivie de leurs prétraitements et pour finir, de quelques algorithmes de classification.

Les chapitres suivants vont apporter des éléments de réponses aux différentes interrogations soulevées, notamment:

Dans le chapitre 2, nous nous pencherons sur les réseaux de neurones et leurs avantages dans les systèmes de reconnaissance de mouvement.

Dans le chapitre 3, nous présenterons les différentes étapes qui nous ont permis de construire notre système de classification, en commençons par l'acquisition des données, le prétraitement. Puis, nous discuterons des résultats finaux, et pour finir nous présenterons l'application Android qui implémente ce classifieur.

CHAPITRE II

Optimisation de l'architecture d'entraînement des données de l'activité physique par apprentissage en profondeur

1. Introduction:

Les méthodes d'apprentissage automatiques explorées dans les chapitres 1 ont permis d'atteindre de bonnes performances de reconnaissance des activités. Toutefois, ces algorithmes nécessitent une étape de calcul et sélection de descripteurs, très fastidieuse à mettre en place. L'exploration des réseaux de neurones profonds, dans différents domaines tels que le traitement d'images, le traitement de la parole, etc, a permis de répondre à ces limitations. En effet, ce type de modèles permet d'apprendre des formes sans avoir à sélectionner des descripteurs. Des résultats très satisfaisants ont été atteints. La mise en place d'un tel type d'approches, pour la reconnaissance de l'activité, nécessite une adaptation des modèles existants, tel que les modèles développés dans le traitement d'images, par exemple, afin qu'ils soient exploitables pour la reconnaissance des activités physiques. En particulier, les modèles de réseaux de neurones récurrents RNN et LSTM.

L'état de l'art a montré que les modèles de réseaux de neurones profonds utilisés pour la reconnaissance de l'activité physique étaient très vastes. En effet, suivant le nombre d'activités à reconnaître et le type de données à entraîner, l'architecture du modèle devient de plus en plus profonde (plusieurs couches). Ceci peut causer plusieurs problèmes tels que la disparition et/ou d'explosion de Gradients et la lenteur du processus d'entraînement. Enfin, un tel type de modèles contenant des milliers de paramètres risquera fort de conduire au sur ajustement du modèle.

Nous proposons dans ce chapitre de répondre à ces limitations à travers deux points :i) nous commencerons d'abord par une brève introduction sur les réseaux de neurone en générale afin de mieux cerner notre problématique. ii) par la suite nous présenterons les modèles de réseaux de neurones que nous avons choisi afin d'élaborer notre classifieur, à savoir les réseaux de neurones récurrents (RNN) et les LSTM (Long-Short-Term Memory units).

2. Les réseaux de neurones

2.1 Introduction

On peut dire que parmi les buts essentiels de la recherche scientifique est de développer des machines intelligentes qui peuvent exécuter toute tâche pénible et encombrante. Parmi les technologies qui sont consacrées à ce type de recherche: l'intelligence artificielle et les systèmes de neurones artificiels. Ces derniers sont basés essentiellement sur le mécanisme de transmission nerveuse d'un être humain. L'élément fonctionnel essentiel du système nerveux est la cellule nerveuse ou neurone qui a pour rôle d'élaborer l'information reçue et transmettre les résultats à d'autres neurones.

Le cerveau humain développe mieux les solutions intelligentes qu'un ordinateur, cependant ce dernier est rapide dans l'exécution des opérations. Les différences entre l'ordinateur et le cerveau humain sont dues à l'architecture de chacun et les méthodes du traitement correspondantes. En vue de traitement de l'information, l'ordinateur utilise des programmes basés sur des algorithmes. Ces derniers opèrent avec des séquences d'instructions contrôlées par une unité centrale complexe, afin d'aboutir à un résultat en fonction des données emmagasinées dans des mémoires. Tandis que le cerveau utilise la notion de transformation, des représentations distribuées et parallèles.

Ce dernier met en communication des milliards des neurones. Les réseaux de neurones sont des structures (la plupart de temps simulées par des algorithmes exécutés sur des ordinateurs d'usage générale, parfois sur des machines ou même des circuits spécialisés) qui prennent leur inspiration (souvent de façon assez lointaine) dans le fonctionnement des systèmes nerveux. Leur domaine d'application est essentiellement celui de résoudre les problèmes de classification, d'association, de reconnaissance de forme, d'extraction des caractéristiques et d'identification.

❖ Les origines de cette discipline sont très diversifiées :

- En 1943, Mc CULLOCH et PITIS ont proposé le premier modèle d'un système de neurones artificiels, qui est encore largement utilisé pour expliquer comment le cerveau peut réaliser les fonctions logiques.
- En 1949, DONALD HEBB décrit une règle sur l'apprentissage ^[37]. Après plusieurs développements dans les modèles des réseaux de neurones, WEBBOS a développé en 1974 un algorithme nommé algorithme de rétropropagation, ce

qui a encouragé d'autres chercheurs à reprendre la recherche dans ce domaine après longue période.

2.2. Définition

Un réseau de neurones est un ensemble de méthodes d'analyse et de traitements des données permettant de construire un modèle de comportement à partir de données qui sont des exemples de ce comportement. Un réseau de neurones est constitué d'un graphe pondéré orienté dont les nœuds symbolisent les neurones. Ces neurones possèdent une fonction d'activation qui permet d'influencer les autres neurones du réseau. Les connexions entre les neurones, que l'on nomme liens synaptiques, propagent l'activité des neurones avec une pondération caractéristique de la connexion. On appelle poids synaptique la pondération des liens synaptiques. Les neurones peuvent être organisés de différentes manières, c'est ce qui définit l'architecture et le modèle du réseau. L'architecture la plus courante est celle dite du perceptron multicouche ^[37]

2.3. Applications

Les réseaux de neurones sont essentiellement utilisés pour faire de la classification. Construit à partir d'exemples de chaque classe qu'il a appris, un réseau de neurones est normalement capable de déterminer à quelle classe appartient un nouvel élément qui lui est soumis.

2.4. Fonctionnement

- La construction de la structure du réseau (généralement empirique).
- La constitution d'une base de données de vecteurs représentant au mieux le domaine à modéliser. Celle-ci est scindée en deux parties: une partie servant à l'apprentissage du réseau (on parle de base d'apprentissage) et une autre partie aux tests de cet apprentissage (on parle de base de test).
- Le paramétrage du réseau par apprentissage. Au cours de l'apprentissage, les vecteurs de données de la base d'apprentissage sont présentés séquentiellement et plusieurs fois au réseau. Un algorithme d'apprentissage ajuste le poids du réseau afin que les vecteurs soient correctement appris. L'apprentissage se termine lorsque l'algorithme atteint un état stable.

- La phase de reconnaissance qui consiste à présenter au réseau chacun des vecteurs de la base de test. La sortie correspondante est calculée en propageant les vecteurs à travers le réseau. La réponse du réseau est lue directement sur les unités de sortie et comparée à la réponse attendue. Une fois que le réseau présente des performances acceptables, il peut être utilisé pour répondre au besoin qui a été à l'origine de sa construction.

3. Modèle biologique

3.1. Définition et structure

Le bloc principal du système nerveux est le neurone. Il transmet l'information reçue vers les diverses parties du corps. Il est constitué:

- D'un corps cellulaire nommé soma
- Des plusieurs épines semblables propagées dans le corps cellulaires nommées dendrites. Leur rôle est de capter les signaux qui proviennent du neurone.
- D'une seule fibre nerveuse nommé axone, qui sert à connecter le corps cellulaire aux autres neurones. L'axone est un moyen de transport pour les signaux émis par le neurone.
- Les connexions entre les neurones se font par l'intermédiaire du corps cellulaire ou les dendrites en jonctions nommées synapses. Les synapses servent à limiter plus ou moins l'amplitude des signaux qui passent d'un neurone à un autre, comme est illustré dans la figure 18.

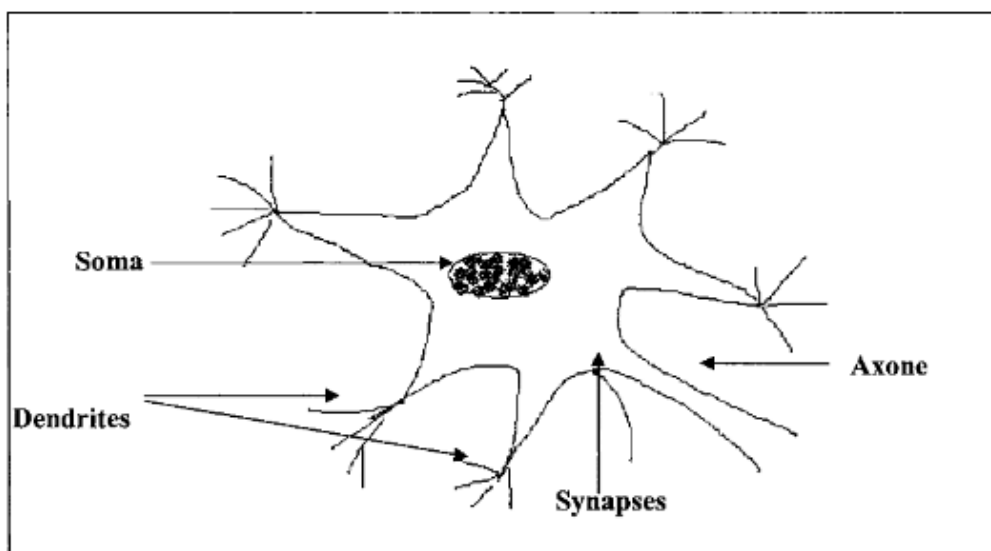


Figure 18 Représentation d'un neurone humain

3.2. Fonctionnement :

Le mécanisme de fonctionnement d'un neurone est de recevoir, grâce à ces nombreuses dendrites, les signaux émis par les autres neurones, puis de décider, à partir des données reçues, d'émettre ou non un signal à ses semblables le long de son axone. Plus précisément, le soma recueille l'ensemble des informations reçues par les dendrites et effectue la sommation dite spatio-temporelle. En raison de sa dimension, l'intégration somatique est aussi temporelle. Si le potentiel somatique dépasse un certain seuil, il y a émission d'un potentiel d'action ou "spike". Le signal, très bref (1 ms), est transmis sans atténuation le long de l'axone et réparti sur le neurone cible.

Le modèle neuronique dont la forme la plus simple est représenté par la figure 19 :

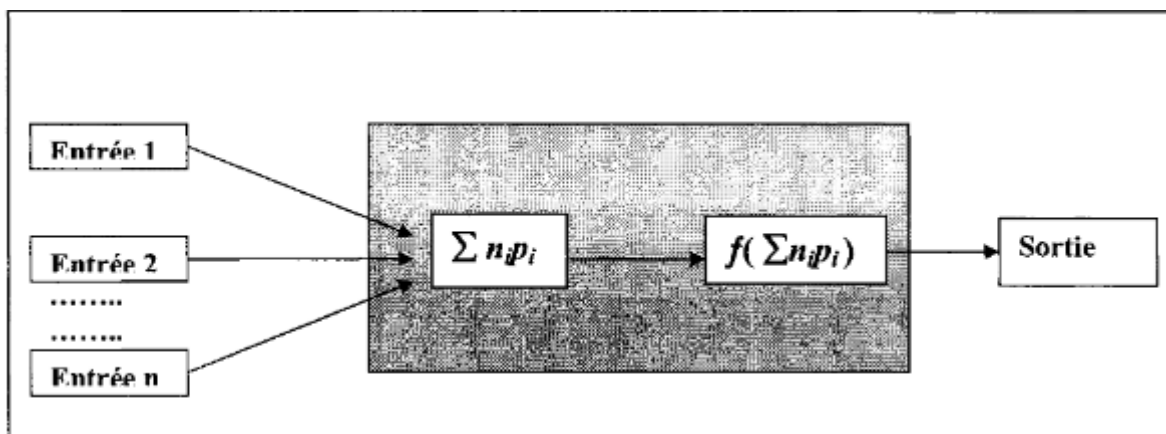


Figure 19. Modèle d'un neurone artificiel.

La figure 19 représente l'architecture générale d'un neurone artificiel, avec les différentes couches qui le constitue, la couche d'entrée, la couche cachée et la couche de sortie. Dans cette figure, on utilise les notations suivantes:

- n_i est la valeur de sortie du nœud i du niveau précédent (la sommation sur i correspond à l'ensemble des nœuds du niveau précédent connectés au nœud observé);
- P_i est le poids associé à la connexion entre le nœud i et le nœud observé;
- f est la fonction de transfert associée au nœud observé.

4. Étude et synthèse d'un réseau de neurone formel

La plus satisfaisante définition d'un réseau de neurone formel, est de celle de HIECHT NILSON:« un réseau de neurone est une structure de traitement parallèle et distribué d'informations comportant plusieurs éléments de traitement *Neurone*, qui peuvent posséder des mémoires locales et exécuter les opérations de traitements sur des informations locales. Ils sont interconnectés les uns aux autres avec des canaux des signaux unidirectionnels. »

La synthèse d'un réseau de neurone formel est basée sur des caractéristiques similaires à celle d'un réseau de neurone biologique ^[38]. Ces caractères sont:

- Il est composé d'un nombre très grand d'éléments de traitement simple.
- Chaque élément de traitement est connecté à plusieurs éléments voisins.
- Le fonctionnement d'un réseau est basé sur le mécanisme de modification de poids de connexion pendant la phase d'apprentissage.

a)-Neurone formel

Un neurone formel est un petit automate qui réalise la somme pondérée des poids W_1, W_2, \dots, W_n des entrées X_1, X_2, \dots, X_n qu'il reçoit du reste du réseau. Chaque nœud du réseau a un niveau d'activation numérique qui lui est associé au temps T . Ce niveau d'activation est modifié, à chaque période, par la quantité totale d'activation qu'il reçoit de ses voisins en entrée.

La figure 20 suivante montre la structure d'un neurone formel:

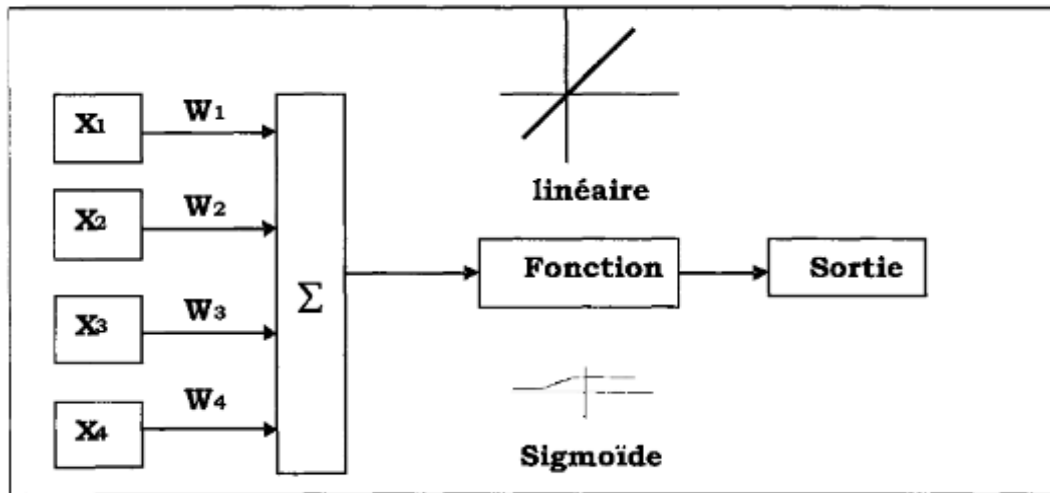


Figure 20 Structure du neurone formel.

b) Fonction d'activation

Afin de déterminer une valeur en sortie, une fonction appelée fonction d'activation (ou de transfert), est appliquée à cette valeur.

La fonction d'activation la plus généralement rencontrée est une fonction sigmoïde telle que « si la somme des entrées est supérieure à un seuil, alors le neurone de sortie est activé; sinon, rien ». La majorité des modèles utilisés aujourd'hui préfèrent employer des fonctions d'activations continues, qui permettent de communiquer et de traiter plus d'informations à la fois dans un seul neurone. Ceci a pour conséquence d'augmenter la puissance de calcul des réseaux. Les étapes dans la mise en œuvre d'un réseau de neurones pour la prédiction ou la classification sont:

1. L'identification des données en entrée et en sortie;
2. La normalisation de ces données;
3. La constitution d'un réseau avec une structure adaptée;
4. L'apprentissage du réseau ;
5. Le test du réseau;
6. L'application du modèle généré par l'apprentissage;

7. La dénormalisation des données en sortie.

4.1. Structure des réseaux de neurone

La structure du réseau de neurones, encore appelée « architecture » ou « topologie » du réseau de neurones, est le nombre de couches et de nœuds, la façon dont sont interconnectés les différents nœuds (choix des fonctions de combinaison et de transfert) et le mécanisme d'ajustement des poids.

4.1.1. Réseau mono-couche et réseau multi-couches

On sait que l'organisation d'un réseau de neurone est constituée de couches, c'est à dire un tel réseau peut contenir une ou plusieurs couches.

❖ Réseau mono-couches

Dans ce type de réseau, il y a une seule couche cachée, qui relie les cellules d'association (couche d'entrée) aux cellules de décision (couche de sortie). C'est la seule couche de connexion modifiable.

Les neurones de la couche d'entrée d'un réseau mono-couche (perceptron) effectuent seulement un prétraitement et la classification effective est effectuée par les neurones de la couche de sortie.

Ce réseau offre une grande convergence vers la solution du problème, malheureusement sa stratégie d'apprentissage n'offre que des séparations linéaires, limitées à la seule classe de problèmes linéairement séparables.

❖ Réseau multi-couches

Pour surmonter les limitations d'un réseau mono-couche, on utilise un réseau multicouche, où la sortie n'est connectée à l'entrée qu'après quelques couches de neurones intermédiaires apportant une richesse à la structure pour accroître la capacité de réseau. Notons que les couches internes n'ont aucune connexion prédéfinie, elles servent seulement à contribuer à l'obtention de résultats souhaités à la sortie. Le problème de séparation linéaire est donc résolu. Pour obtenir une séparation linéaire, on doit tenir compte de ce qui a été dit plus haut d'une part, et le bon dimensionnement en utilisant le modèle de rétro propagation d'autre part.

4.1.2. Réseaux récurrents et réseaux non récurrents

Les réseaux de neurones sont répartis en deux grandes classes :

❖ Les réseaux non récurrents (statiques)

Dans ce type de réseau, on utilise une structure à couche. Pour ce type de réseau les neurones de la même couche ne sont pas connectés, chaque couche reçoit des signaux de la couche précédente et transmet le résultat de ce traitement à la couche suivante. En conséquence le signal d'entrée prend un sens unique de l'entrée vers la sortie tout en ayant traversé des couches cachées.

Un réseau de neurones statique est généralement organisé en plusieurs couches de neurones appelées réseaux multicouches comme il est illustré dans la figure 21.

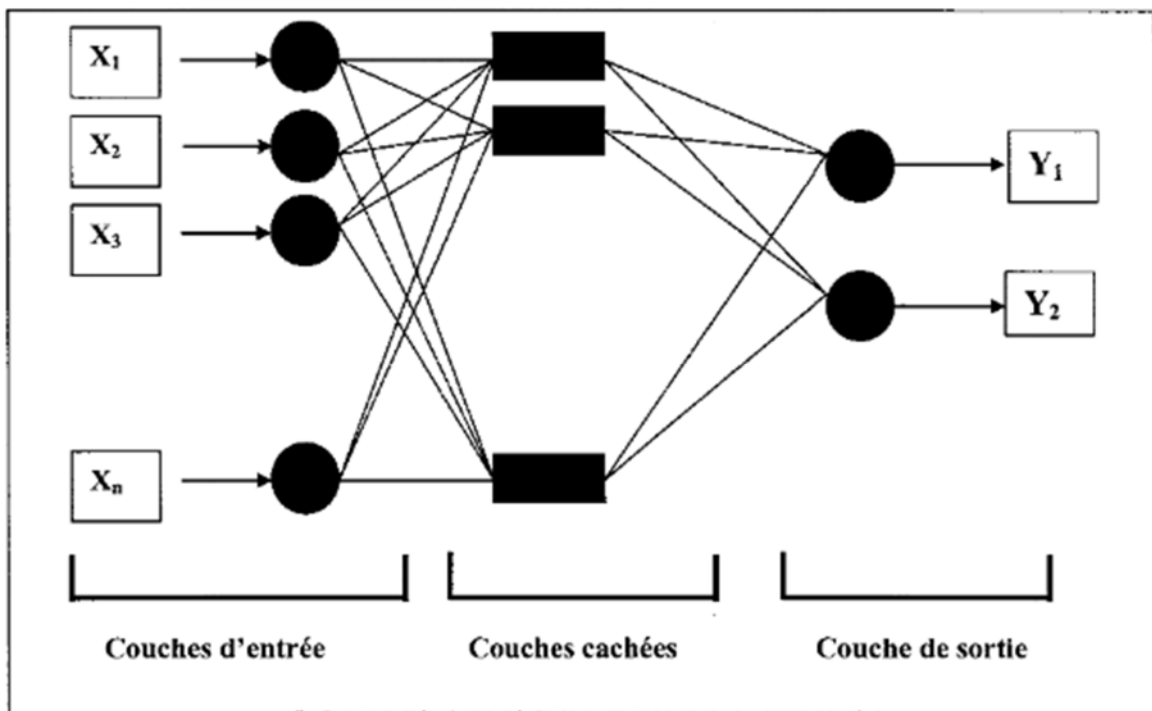


Figure 21. Architecture d'un réseau statique

Un réseau statique est constitué par:

- Une couche d'entrée qui reçoit les signaux d'entrée du milieu externe.
- Une ou plusieurs couches cachées (intermédiaire)
- Une couche de sortie qui fournit les résultats de traitement du réseau

❖ Réseaux récurrents (dynamiques)

Dans ce type de réseaux, les neurones sont entièrement connectés et les sorties des neurones de la couche sortie sont réinjectées sur les entrées des neurones précédents d'où l'existence d'une boucle de retour, cette dernière a pour rôle d'équilibrer le système lorsqu'il est soumis à un stimulus extérieur. Dans le réseau récurrent les décisions ne sont pas présentes instantanément, mais par étapes successives.

Plus précisément, Un réseau de neurones récurrents est constitué d'unités (neurones) interconnectées interagissant non-linéairement et pour lequel il existe au moins un cycle dans la structure. Les unités sont reliées par des arcs (synapses) qui possèdent un poids. La sortie d'un neurone est une combinaison non linéaire de ses entrées^[39].

Ils sont utilisés en reconnaissance automatique de la parole ou de l'écriture manuscrite - plus en général en reconnaissance de formes - ou encore en traduction automatique. «Dépliés», ils sont comparables à des réseaux de neurones classiques avec des contraintes d'égalité entre les poids du réseau (voir schéma ci-dessous).

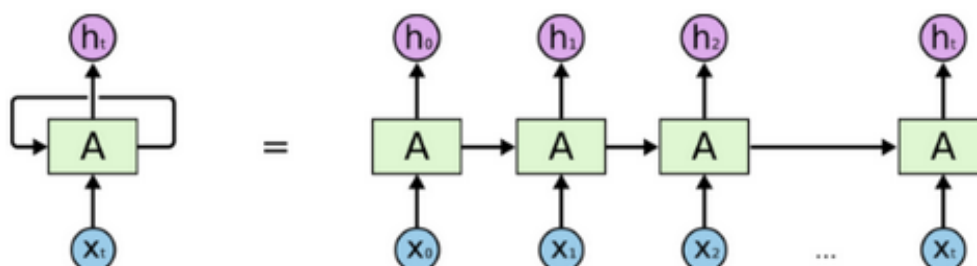


Figure 22. Schéma d'un réseau de neurones récurrents à une unité reliant l'entrée et la sortie du réseau. A droite la version « dépliée » de la structure.

L'état courant peut-être défini comme suit :

$$h_t = f(h_{t-1}, x_t) \quad (6)$$

En appliquant la fonction d'activation :

$$h_t = \tanh (W_{hh}h_{t-1} + W_{xh}x_t) \quad (7)$$

W :le poids associé à un neurone

h: l'unique vecteur caché

W_{hh}: le poids à la précédente couche cachée t

W_{hx} : le poids à l'état d'entrée actuel

tanh : La fonction d'activation, qui implémente une Non-linéarité, ce qui restreint les activations dans l'intervalle [-1.1]

$$y_t = W_{hy}h_t \quad (8)$$

En sortie:

Y_t: l'état de sortie.

W_{hy}: Lepoid à l'état de sortie.

Les techniques d'entraînement du réseau sont les mêmes que pour les réseaux classiques (rétropropagation du gradient^[40]), néanmoins les réseaux de neurones récurrents se heurtent au problème de disparition du gradient pour apprendre à mémoriser des évènements passés. Des architectures particulières répondent à ce dernier problème, on peut citer en particulier les réseaux Long short-termmemory.

❖ Les réseaux LSTM:

Un réseau **Long short-termmemory (LSTM)**^[41], en français réseau récurrent à mémoire court et long terme ou plus explicitement réseau de neurones récurrents à mémoire court-terme et long terme, est l'architecture de réseau de neurones récurrents la plus utilisée en pratique qui permet de répondre au problème de disparition de gradient.

L'idée associée au LSTM est que chaque unité computationnelle est liée non seulement à un état caché \mathbf{h} mais également à un état \mathbf{c} de la cellule qui joue le rôle de mémoire. Le passage de c_{t-1} à c_t se fait par transfert à gain constant et égal à 1. De cette façon les erreurs se propagent aux pas antérieurs (jusqu'à 1 000 étapes dans le passé) sans phénomène de disparition de gradient. L'état de la cellule peut être modifié à travers une porte qui autorise ou bloque la mise à jour (*input gate*). De même une porte contrôle si l'état de cellule est communiqué en sortie de l'unité LSTM (*output gate*). La version la plus répandue des LSTM utilise aussi une porte permettant la remise à zéro de l'état de la cellule (*forgetgate*)^[42].

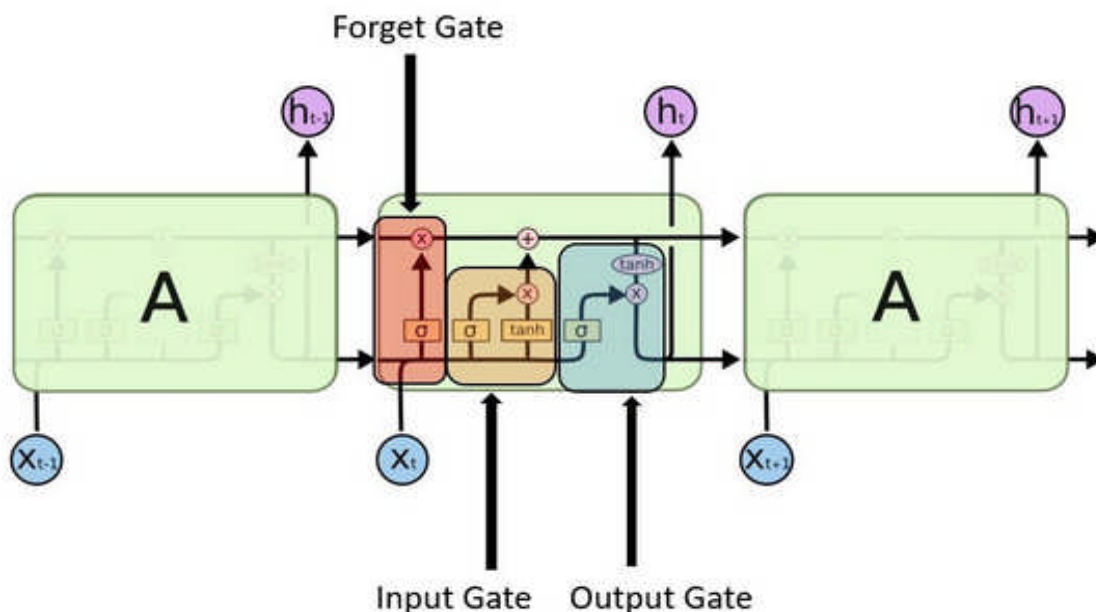


Figure 23. Schéma d'un réseau LSTM à une unité. Le graphe des opérations est détaillé pour l'étape t . Les poids ne sont pas indiqués.

- a) **Input gâte:** découvre quelle valeur d'entrée doit être utilisée pour modifier la mémoire. La fonction *sigmoïde* décide des valeurs à laisser passer, elle permet de réduire la valeur d'entrée à un nombre compris entre 0 et 1. Et la fonction *tanh* donne un poids aux valeurs qui sont passées en décidant leur niveau d'importance allant de -1 à 1.

$$\begin{aligned} i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ \tilde{C}_t &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \end{aligned} \tag{9}$$

Input gate

b) **Forget gâte:** découvre les détails à éliminer du bloc. Ceci est décidée par la fonction *sigmoïde*. On examine l'état précédent (h_{t-1}) et l'entrée de contenu (X_t) et génère un nombre compris entre]0;1] pour chaque nombre dans la cellule d'état C_{t-1} .

$$f_t = \sigma (W_f \cdot [h_{t-1}, x_t] + b_f) \quad (10)$$

Forget gate

c) **Output gâte:** L'entrée et la mémoire du bloc sont utilisées pour décider de la sortie. La fonction *sigmoïde* décide des valeurs à laisser passer (un nombre entre 0 et 1). Et la fonction *tanh* donne un poids aux valeurs qui sont passées en décidant leur niveau d'importance allant de -1 à 1 et le tout est multiplié par la sortie de *Sigmoid*

$$o_t = \sigma (W_o [h_{t-1}, x_t] + b_o) \quad (11)$$

$$h_t = o_t * \tanh (C_t)$$

Output gate

❖ Nb: L'opérateur \circ symbolise le produit matriciel de Hadamard (produit terme à terme). Les symboles σ et \tanh représentent respectivement la fonction sigmoïde et la fonction tangente hyperbolique, bien que d'autres fonctions d'activation soient possibles.

4.1.3. Fonctionnement d'un réseau

Un réseau de neurone peut fonctionner en deux modes, parallèle ou séquentiel. Dans le mode parallèle tous les neurones calculent leurs nouvelles activations et leurs sorties, et les transmettent aux neurones auxquels ils sont connectés, à chaque top d'horloge. Contrairement, au mode séquentiel, un seul neurone calcule sa nouvelle activation et sa sortie puis les transmet aux neurones auxquels il est connecté, à chaque top d'horloge. Donc, le calcul est fait en fonction des entrées des neurones au top d'horloge précédent. On peut obtenir d'autres modes dits mixtes en combinant les deux modes précédents.

4.1.4. Apprentissage

On peut définir l'apprentissage par la modification des interactions entre neurones, l'apprentissage consiste donc à ajouter les poids synaptiques de telle façon que le réseau présente

un certain comportement désiré. Les procédures d'apprentissage peuvent se subdiviser, en deux grandes catégories: Apprentissage supervisé ou apprentissage non supervisé.

a)-Apprentissage supervisé

L'apprentissage supervisé implique l'existence d'un professeur qui a pour rôle d'évaluer le succès ou l'échec du réseau quand on lui présente un stimulus connu. Cette supervision consiste à renvoyer au réseau une information lui permettant de faire évoluer ses connections afin de faire diminuer son taux d'échec. C'est à dire que ce professeur présente au réseau de neurones une entrée et la sortie désirée correspondante, pour faire la comparaison avec les sorties actuelles des vecteurs d'entrées. A partir de l'erreur calculée, les poids sont ajustés pour avoir des sorties correspondantes aux réponses désirées. Ce calcul se répète jusqu'à ce que l'erreur soit minimale par rapport à un critère préalable, et par conséquent les coefficients synaptiques prennent les valeurs optimales.

b)-Apprentissage non supervisé

Les réseaux, utilisant l'apprentissage non supervisé, sont souvent appelés auto-organiseurs, ou encore à apprentissage compétitif. Dans ce type d'apprentissage la connaissance de la sortie désirée n'est pas nécessaire c'est à dire que le réseau s'auto-organise et organise les entrées qui sont présentées de façons à optimiser un critère de coût donné.

4.1.5. Choix de l'échantillon d'apprentissage

L'apprentissage du réseau de neurones sera d'autant meilleur qu'il s'effectuera sur un échantillon suffisamment riche pour représenter toutes les valeurs possibles de nœuds de toutes les couches du réseau, c'est-à-dire en particulier toutes les modalités possibles de chaque variable, en entrée ou en sortie.

Il faut aussi veiller à ce que les enregistrements analysés ne soient pas triés selon un ordre significatif.

5. Développement d'un réseau de neurones

La procédure de développement d'un réseau de neurones suit le cycle classique suivant :

1. La collecte des données,

2. L'analyse des données,
3. La séparation des bases de données,
4. Le choix d'un réseau de neurones,
5. La mise en forme des données,
6. L'apprentissage,
7. La validation.

5.1. Collecte des données

L'objectif de cette étape est de recueillir des données, à la fois pour développer le réseau de neurones et pour le tester. Dans le cas d'applications sur des données réelles, l'objectif est de rassembler un nombre de données suffisant pour constituer une base représentative des données susceptibles d'intervenir en phase d'utilisation du système neuronal.

La fonction réalisée résultant d'un calcul statistique, le modèle qu'il constitue n'a de validité que dans le domaine où on l'a ajusté. En d'autres termes, la présentation de données très différentes de celles qui ont été utilisées lors de l'apprentissage peut entraîner une sortie totalement imprévisible.

5.2. Analyse des données

Il est souvent préférable d'effectuer une analyse des données de manière à déterminer les caractéristiques discriminantes pour détecter ou différencier ces données. Ces caractéristiques constituent l'entrée du réseau de neurones. Notons que cette étude n'est pas spécifique aux réseaux de neurones, quel que soit la méthode de détection ou de classification utilisée, il est généralement nécessaire de présenter des caractéristiques représentatives.

Cette détermination des caractéristiques a des conséquences à la fois sur la taille du réseau (et donc le temps de simulation), sur les performances du système (pouvoir de séparation, taux de détection), et sur le temps de développement (temps d'apprentissage).

Une étude statistique sur les données peut permettre d'écartier celles qui sont aberrantes et redondantes.

Dans le cas d'un problème de classification, il appartient à l'expérimentateur de déterminer le nombre de classes auxquelles ses données appartiennent et de déterminer pour chaque donnée la classe à laquelle elle appartient.

5.3. Séparation des bases de données

Afin de développer une application à base de réseaux de neurones, il est nécessaire de disposer de deux bases de données : une pour l'apprentissage et une autre pour tester le réseau obtenu et déterminer ses performances. Afin de contrôler la phase d'apprentissage, il est généralement préférable de posséder une troisième base de données appelée « base de validation croisée ».

Les avantages liés à l'utilisation de cette troisième base de données seront exposés dans les sections suivantes. Il n'y a pas de règle pour déterminer ce partage de manière quantitative. Il résulte souvent d'un compromis tenant compte du nombre de données dont on dispose et du temps imparti pour effectuer l'apprentissage. Chaque base doit cependant satisfaire aux contraintes de représentativité de chaque classe de données et doit généralement refléter la distribution réelle, c'est à dire la probabilité d'occurrence des diverses classes.

5.4. Choix d'un réseau de neurones

Il existe un grand nombre de types de réseaux de neurones, avec pour chacun des avantages et des inconvénients. Le choix d'un réseau peut dépendre :

- De la tâche à effectuer (classification, association, contrôle de processus, séparation aveugle de sources...),
- De la nature des données,
- D'éventuelles contraintes d'utilisation temps-réel (certains types de réseaux de neurones, tels que la 'machine de Boltzmann', nécessitant des tirages aléatoires et un nombre de cycles de calculs indéfini avant stabilisation du résultat en sortie, présentent plus de contraintes que d'autres réseaux pour une utilisation temps-réel),
- Des différents types de réseaux de neurones disponibles dans le logiciel de simulation que l'on compte utiliser.

Ce choix est aussi fonction de la maîtrise ou de la connaissance que l'on a de certains réseaux, ou encore du temps dont on dispose pour tester une architecture prétendue plus performante.

5.5. Mise en forme des données pour un réseau de neurones

De manière générale, les bases de données doivent subir un prétraitement afin d'être adaptées aux entrées et sorties du réseau de neurones. Un pré traitement courant consiste à effectuer une normalisation appropriée, qui tient compte de l'amplitude des valeurs acceptées par le réseau.

5.6. Apprentissage du réseau de neurones

Tous les modèles de réseaux de neurones requièrent un apprentissage. Plusieurs types d'apprentissages peuvent être adaptés à un même type de réseau de neurones. Les critères de choix sont souvent la rapidité de convergence ou les performances de généralisation.

Le critère d'arrêt de l'apprentissage est souvent calculé à partir d'une fonction de coût, caractérisant l'écart entre les valeurs de sortie obtenues et les valeurs de références (réponses souhaitées pour chaque exemple présenté).

La technique de validation croisée, qui sera précisée par la suite, permet un arrêt adéquat de l'apprentissage pour obtenir de bonnes performances de généralisation.

Certains algorithmes d'apprentissage se chargent de la détermination des paramètres architecturaux du réseau de neurones. Si on n'utilise pas ces techniques, l'obtention des paramètres architecturaux optimaux se fera par comparaison des performances obtenues pour différentes architectures de réseaux de neurones.

Des contraintes dues à l'éventuelle réalisation matérielle du réseau peuvent être introduites lors de l'apprentissage.

5.7. Validation

Une fois le réseau de neurones entraîné (après apprentissage), il doit être testé sur une base de données différente de celle utilisée pour l'apprentissage ou la validation croisée. Ce test permet d'évaluer les performances du système neuronal et de détecter des types de données problématiques. Si les performances ne sont pas satisfaisantes, il est nécessaire de modifier l'architecture du réseau ou de modifier la base d'apprentissage (caractéristiques discriminantes ou représentativité des données de chaque classe).

6. Architecture proposée

Notre choix s'est porté sur l'utilisation d'un classifieur basée sur les réseaux neuronaux récurrents (RNN) et les réseaux de longue mémoire à court terme (LSTM).

En effet, les réseaux de neurones récurrents "RNN" sont particulièrement adaptés pour des données d'entrée de taille variable. Ils conviennent en particulier pour l'analyse de séries temporelles, comme c'est le cas dans notre étude (voir page 46).

Compte tenu de la littérature dans le domaine, nous avons également fait le choix d'intégrer dans notre modèle un deuxième réseau neuronal, mais cette fois ci de type LSTM (Long Short Term Memory) (voir page 49) qui répond très bien au contrainte de notre étude.

Ainsi, notre modèle contient 2 couches entièrement connectées et 2 couches LSTM (empilées l'une sur l'autre) avec 64 unités chacune.

6.1 Fonction de perte et optimisation:

Une « fonction de perte » (*lossfunction*)^[43], quantifie l'écart entre les prévisions du modèle et les observations réelles du jeu de donnée utilisé pendant l'apprentissage. Tout le jeu de la phase d'apprentissage consistera à ajuster les paramètres du réseau de neurones de façon à minimiser cette fonction de perte.

Pour calculer la perte, nous avons utilisé la fonction de perte "L2-norm", également connue sous le nom de somme des moindres carrés (Least SquaredError).

Cette fonction minimise la **somme des carrés des écarts** des valeurs observés (Y_i) et des valeurs estimées ($f(x_i)$).

La fonction "L2-norm" est définie comme suit:

$$S = \sum_{i=1}^n (y_i - f(x_i))^2 \quad (12)$$

Et pour l'optimisation du modèle (étape qui consiste à apprendre les poids synaptiques et les biais de chaque neurone), nous avons utilisé l'algorithme d'optimisation Adam, cet algorithme est une extension descente de gradient stochastique. Le principe de cette méthode, est de faire que le taux d'apprentissage s'adapte aux paramètres, faisant de sorte qu'il s'ajuste automatiquement, en fonction de "l'éparse " des paramètres. Il peut donc être vu comme une généralisation de

l'algorithme Adagrad (diminutif de adaptive gradient, anglais pour «gradient adaptatif »), qui est une méthode de descente de gradient (itérative) utilisée pour la minimisation d'une fonction objectif qui est écrite comme une somme de fonctions différentiables.

7. Conclusion:

Dans ce chapitre nous avons abordé les différentes facettes des réseaux de neurones, et avons montré pour quelles raisons, ils représentent une bonne alternative pour la construction de notre classifieur. Et plus spécifiquement, les réseaux de neurone récurrent (RNN) et les réseaux LSTM (Long Short Term Memory), qui sont tout particulièrement adapté à notre sujet d'étude.

On peut donc dire, que le grand avantage des réseaux de neurones réside dans leur capacité d'apprentissage automatique, ce qui permet de résoudre des problèmes sans nécessiter l'écriture de règles complexes, tout en étant tolérant aux erreurs. Cependant, ce sont de véritables boîtes noires qui ne permettent pas d'interpréter les modèles construits. En cas, d'erreurs du système, il est quasiment impossible d'en déterminer la cause.

Dans le chapitre suivant, nous allons détailler les différentes étapes qui nous ont permis de construire notre classifieur, par la suite nous discuterons des résultats expérimentaux, et pour finir nous présenteront l'application Android qui implémente ce classifieur.

CHAPITRE III

Conception de la solution et résultats expérimentaux

1. Introduction

Nous avons proposé, pour la reconnaissance des activités, une approche fondée sur un modèle de classification comprenant deux réseaux de neurones distincts, prenant comme entrée les signaux bruts des deux capteurs présents dans le smartphone à savoir, l'accéléromètre et le gyroscope.

Dans ce chapitre, nous aborderons la conception et la réalisation de ce système de reconnaissance, ainsi que son implémentation sur une application Android.

Pour ce faire, nous détaillerons dans une première partie notre solution de façon conceptuelle, et cela en décrivant les différentes étapes de sa construction. En commençant par la phase d'acquisition des données et de construction du dataset, suivi par la phase de prétraitement, et enfin par l'élaboration du modèle.

Par la suite, nous évoquerons les outils utilisés durant cette étude (environnement de développement, langage de programmation, ressource ...).

Pour finir, nous présenterons les résultats obtenus et évaluerons les performances de la méthode de classification étudiée à travers les métriques d'évaluation choisies dans la phase conception, suivi des étapes d'implémentation du système et de réalisation de notre application mobile.

2. Conception architecturale

La figure ci-dessous représente un schéma de l'architecture de la solution proposée pour la classification de données d'activités humaines.

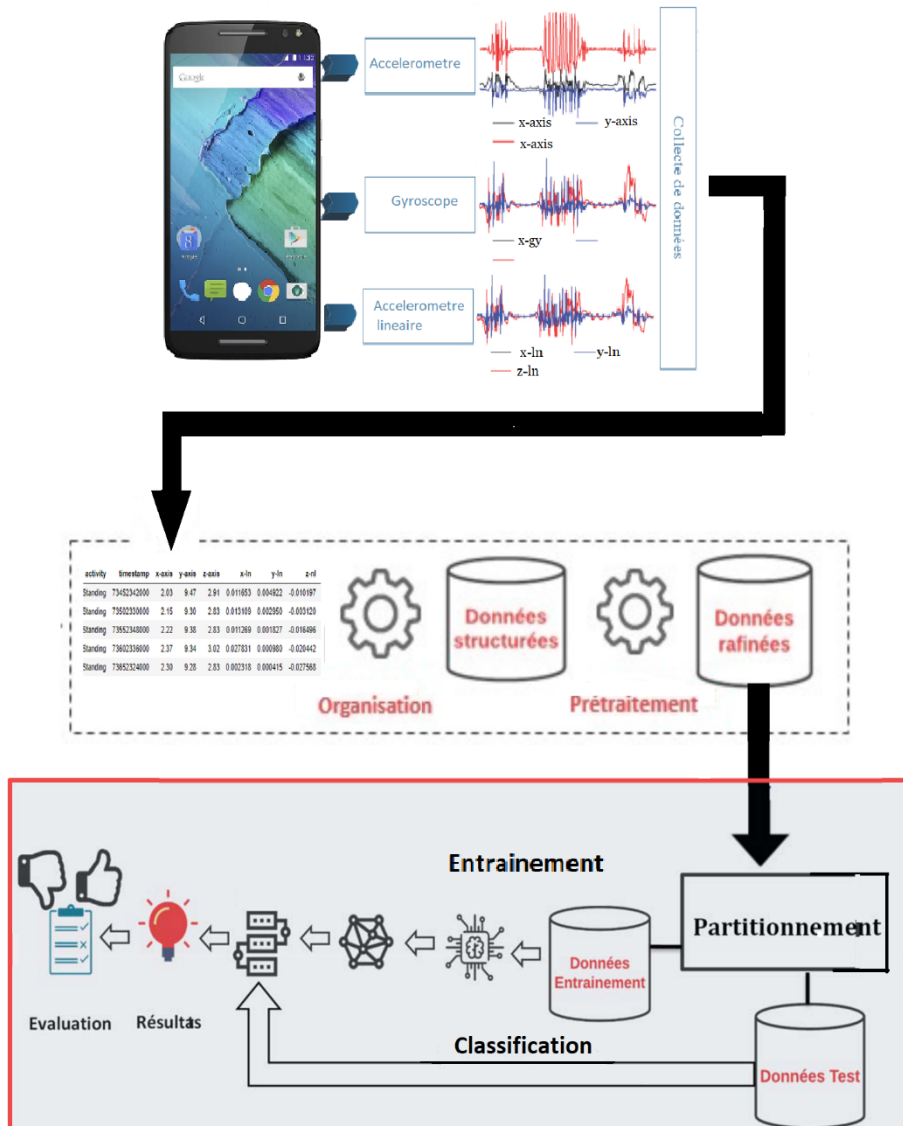


Figure 24. Procédure de développement du système de classification comportementale

- Notre système est composé principalement de trois modules dont les fonctionnalités sont les suivantes :
 - Acquisition et préparation des données.
 - Classification des données.

- Evaluation des résultats obtenus de la classification.

2.1. Acquisition de données

La première étape consistait à collecter les données provenant des capteurs présents dans les smartphones.

La collecte de données a eu lieu principalement au niveau du stade polyvalent de Tizi-Ouzou et dans les locaux du département informatique de l'UMMTO (Université Mouloud Mammeri de Tizi-Ouzou) avec la participation de 20 personnes âgées de 13 à 82 ans appartenant à différentes catégories (sportif, âgé, ...).

Un smartphone est placé dans la poche des participants (figure 25), et chacun d'eux est appelé à faire, deux fois, un cycle contenant dans l'ordre les 6 activités physiques suivantes (voir figure 26): monter et descendre l'escalier, marcher, courir, s'asseoir, et rester debout.

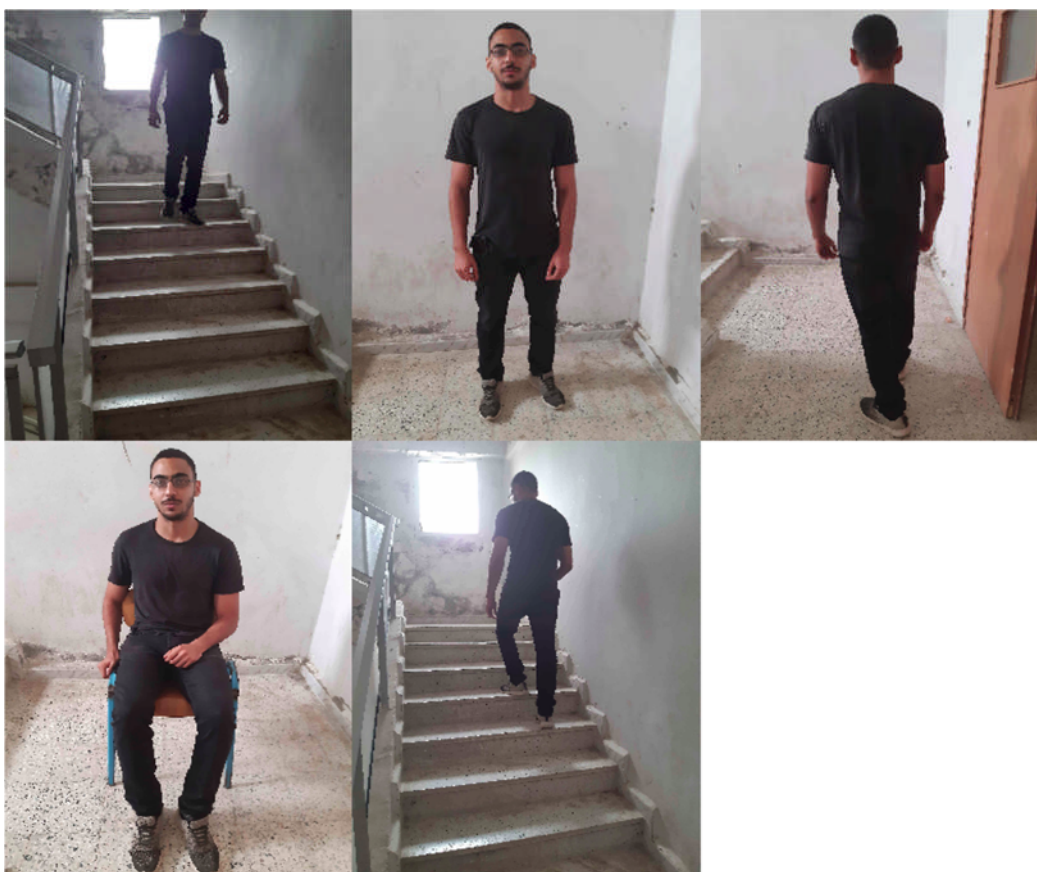


Figure 25. Exemple de la récolte des données d'un participant

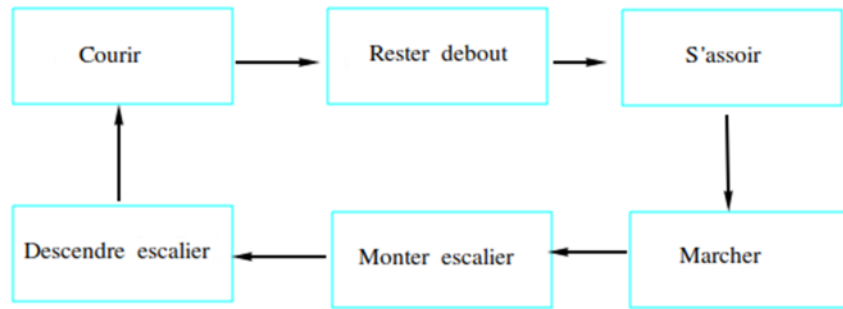


Figure 26. Cycle d'activité physique

Durant le cycle d'activité, les téléphones intelligents enregistrent les données brutes des capteurs accéléromètre et gyroscope avec une fréquence d'échantillonnage de 50 Hz (1 échantillon toutes les 20 ms).

Les Participants ont été filmé afin d'étiqueter manuellement les différentes classes d'activités physiques (marcher, debout, courir, s'asseoir, monter et descendre les escaliers). Ces étiquettes ont servi de références pour la validation des méthodes d'analyse et de classification de données.

- **Structure de notre DataSet :**

La phase d'acquisition mentionné précédemment, nous a permis de constituer un dataset (jeu de données) comportant les valeurs des signaux des deux capteurs : accéléromètre sur 3 axes et gyroscope sur 3 axes, à savoir que, les signaux de l'accéléromètre sont divisés en deux composantes :

- L'accélération linéaire (accélération du corps en mouvement sans la composante gravitationnel)
- L'accélération gravitationnelle (accélération du corps en mouvement avec la composante gravitationnel)

L'ensemble de ces données ont été rassemblé dans une base de donnée contenant 697 964 enregistrement de 13 champs chacun, soit une table de (697 964 X 13). Le tableau ci-dessous en montre un aperçu.

user	activity	timestamp	x-axis	y-axis	z-axis	x-ln	y-ln	z-ln	x-gy	y-gy	z-gy	
0	35	Standing	73452342000	2.03	9.47	2.91	0.011653	0.004922	-0.010197	-0.015542	0.004922	-0.023481
1	35	Standing	73502330000	2.15	9.30	2.83	0.013109	0.002950	-0.003120	-0.016666	0.002950	-0.017628
2	35	Standing	73552348000	2.22	9.38	2.83	0.011269	0.001827	-0.016496	-0.006630	0.001827	-0.021150
3	35	Standing	73602336000	2.37	9.34	3.02	0.027831	0.000980	-0.020442	-0.023481	0.000980	-0.032659
4	35	Standing	73652324000	2.30	9.28	2.83	0.002318	0.000415	-0.027568	-0.017628	0.000415	-0.022478

Tableau 3. Aperçu de la base de données

Une fois les données récoltées et stockées, il est primordial de les transformer sous un format adéquat afin de les utiliser comme entrée dans notre classificateur. C'est pour cela que nous avons procédé à leur nettoyage et prétraitement afin d'aboutir à une structure utilisable.

2.2 Prétraitement des données

Comme mentionné dans le chapitre 1, le processus de classification passe en premier temps par la phase de prétraitement, car la qualité des données et les informations utiles qui peuvent en être dérivées affectent directement la capacité d'apprentissage de notre modèle; il est donc extrêmement important de prétraiter les données avant leur intégration dans le modèle.

Nous avons donc nettoyé notre dataset en y supprimant les différentes erreurs inhérentes au processus d'acquisition de large quantité de données par des capteurs à savoir : les erreurs de saisie, les enregistrements redondants, invalides et comportant des valeurs nulles.

Gestion des valeurs nulles et aberrantes : La première étape consiste à éliminer tous les champs comportant des valeurs nulles (des zéros) ou aberrantes (infini, caractères alphanumériques, invalides ...), non utilisables pour la suite.

Gestion des valeurs manquantes: La deuxième étape vise à gérer les données manquantes. En effet, l'absence des valeurs n'est pas compatible avec tous les outils de Data Mining.

Il faut gérer ces valeurs manquantes selon l'une des méthodes suivantes:

- **Exclure les enregistrements incomplets:** Cette première méthode, très restrictive, consiste à exclure tous les enregistrements dont une valeur manque.
- **Remplacer les données manquantes:** La deuxième méthode, supportée par certains logiciels, remplace la donnée absente par une valeur qui est soit choisie par l'utilisateur (remplacée par la moyenne ou la médiane, par exemple), soit calculée (remplacée par le résultat d'une formule de score), soit héritée.
- **Gérer les valeurs manquantes:** Lorsque l'absence de données est acceptable du point de vue de la performance du modèle, les algorithmes offrent généralement la possibilité de gérer à part la valeur manquante en la distinguant des valeurs renseignées, ou celle considérer la valeur manquante comme un facteur d'indécision.

Une fois le nettoyage effectué, nous pouvons passer à la visualisation de nos données. (Voir figure 27 et 28).

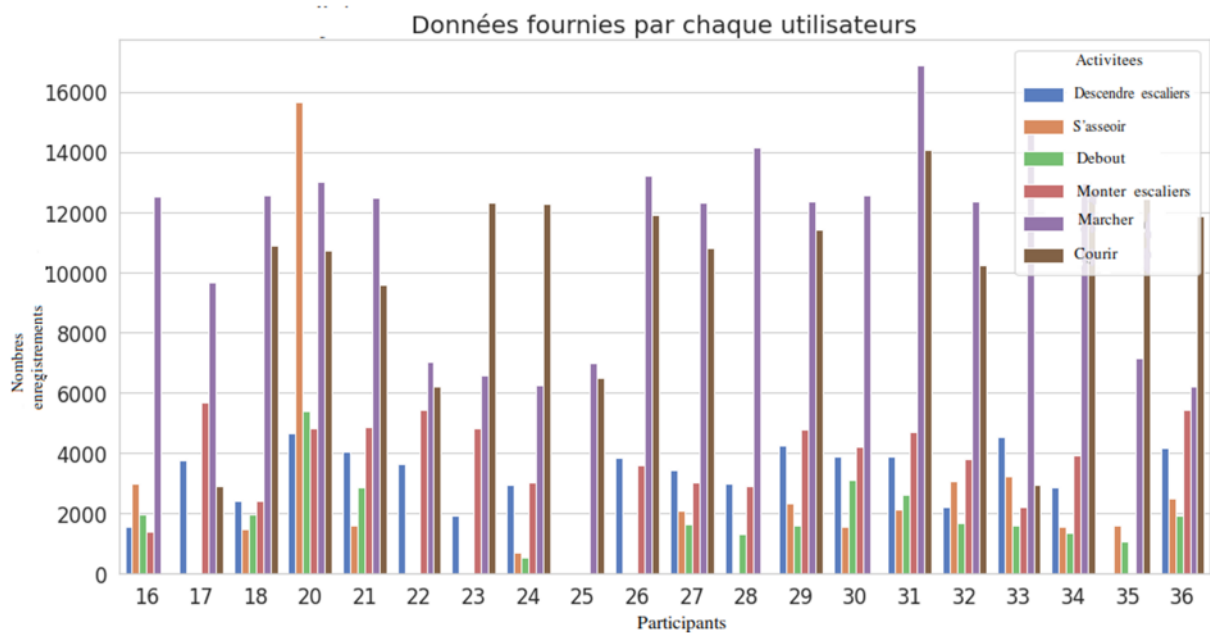


Figure 27 Distribution des activités par participants

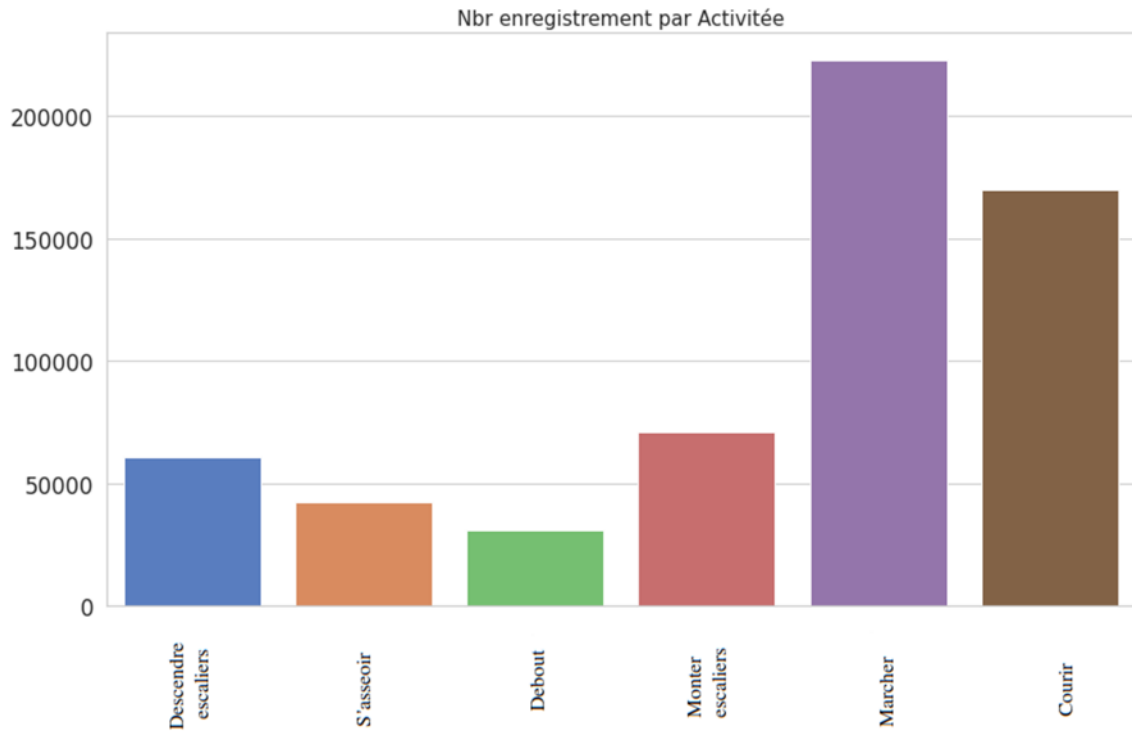


Figure 28. Distribution du nombre d'enregistrement en fonction des activités

La figure 27 nous montre le nombre d'enregistrement par activité et par participant. On peut voir que la distribution des activités est assez homogène entre les participants, ils ont approximativement tous le même nombre d'enregistrement, à part le participant numéros 25 qui ne semble avoir fourni que deux activités (surement dû à une erreur lors de l'acquisition). De plus, on remarque une asymétrie entre les différentes activités.

Les deux activités *marcher* et *courir* ont été enregistrées de façon beaucoup plus significative que les autres. Cette tendance est confirmée avec la figure 28, où l'on peut voir clairement que les quatre activités : *descendre escalier*, *s'asseoir*, *rester debout* et *monter l'escalier* ont été enregistré près de 50 000 fois, alors que les deux autres : *Marcher* et *courir* dépassent les 150 000 enregistrements chacun. Ce phénomène peut être expliqué par les conditions dont les participants ont été évalués (voir section 2.1).

Ce déséquilibre entre activités n'est pas optimal pour la conception du classifieur, mais cela reste tout de même acceptable.

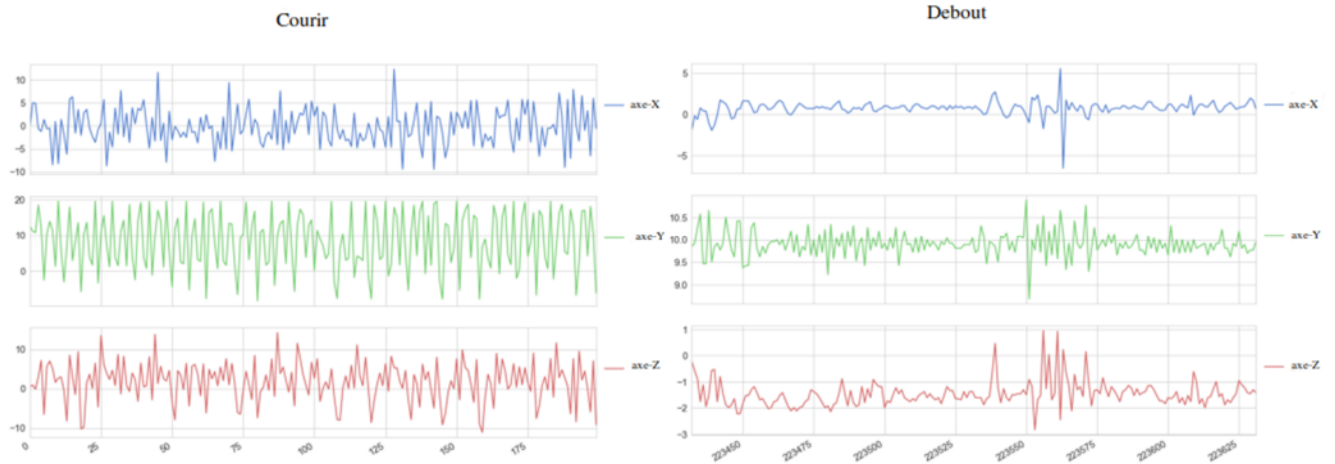
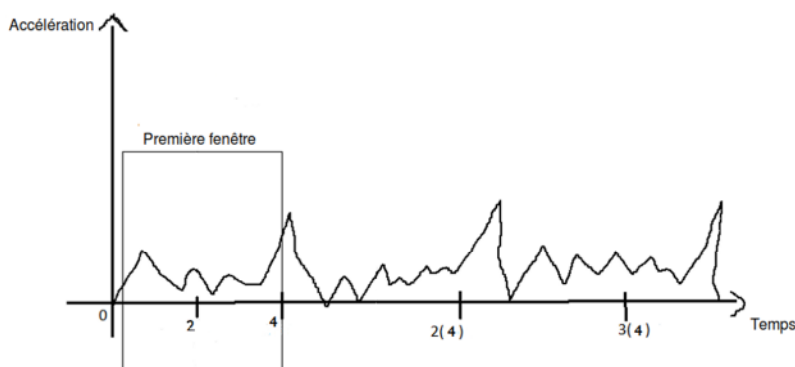


Figure 29 Signaux accéléromètre durant différentes activités

La figure 29, montre les signaux d'accéléromètre de différentes activités pour leurs 200 premières entrées (ici *Courir* et *rester Debout*). Il semble alors raisonnable de supposer que ces données pourraient être utilisées pour former un modèle capable de distinguer les différents types d'activités.

- **Organisation des données**

Une autre étape importante du prétraitement des données concerne la segmentation, où les signaux ont été **échantillonnés** dans des fenêtres de largeur fixe de 4 secondes chacune (chaque fenêtre a 200 enregistrements), comme mentionné sur les figure 30 et 31.



Dividing data into windows of size 4 sec

Figure 30 Découpage des données en séquence (1)

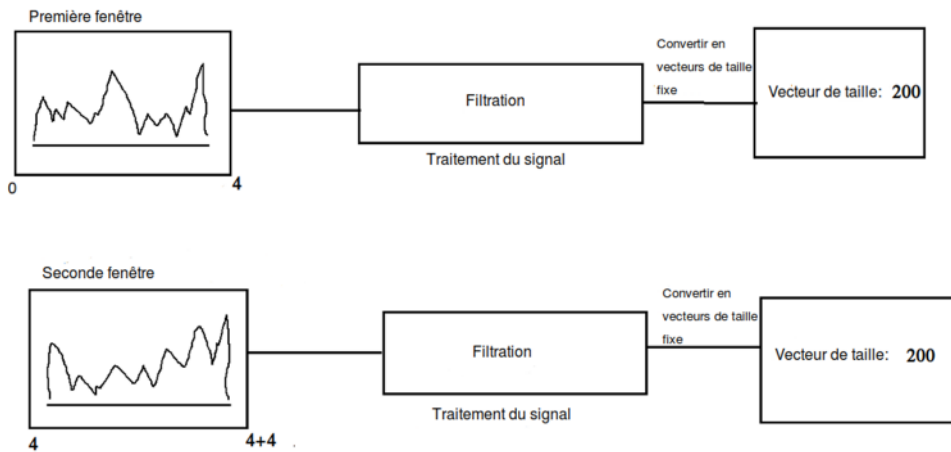


Figure 31. Découpage des données en séquence (2)

Par la suite, l'ensemble des données obtenues a été divisé au hasard en deux ensembles, où 70% (14 personnes) des volontaires ont été sélectionnés pour générer les données de formation destinée à l'entraînement du modèle de classification et 30%(6 personnes) pour former les données de test qui permettront d'effectuer le test et évaluer les performances du classifieur.

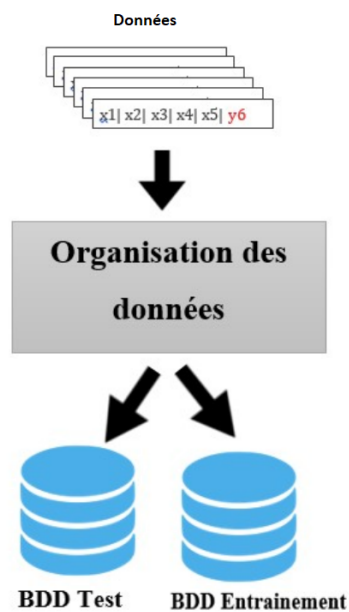


Figure 32 Segmentation du DataSet .

2.3 Classification des données

Après la collecte et la préparation de données, vient la phase qui consiste à entraîner le classificateur afin de construire et générer des règles sous forme de modèle pour une classification ultérieure de nouveaux échantillons.

Comme nous l'avons déjà mentionné précédemment, dans cette étude nous avons utilisé un RNN (Recurrent Neural Nets) et un LSTM (Long-Short-Term Memory units) pour la construction du modèle (Voir chapitre 2). Pour faire cette construction, nous avons défini et initialisé le classifieur en utilisant des fonctions prédéfinies, puis entraîné le modèle avec la fonction "**Fit**", qui accepte en entrée le DataSet d'entraînement, et retourne un modèle entraîné et opérationnel (voir figure 31).

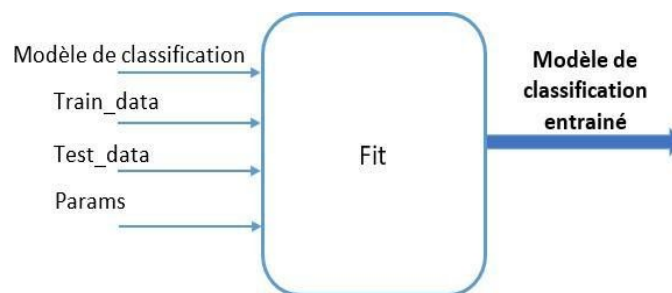


Figure 33 : La Fonction Fit.

Après ce traitement, le modèle de classification entraîné est prêt pour être évalué et déployé dans un environnement réel pour reconnaître et classifier des activités humaines.

Pour la phase d'entraînement (Processus consistant à déterminer les paramètres idéaux du modèle), nous avons utilisé 50 *epochs* (itérations, où toute les données d'entraînement passe à travers le réseau) et avons défini la *taille d'un lot* (batch size, c.à.d. le nombre d'exemple qui traverse le réseau en une seul fois) à 1024.

3. Environnement et outils de développement :

Pour la réalisation de cette application, nous avons utilisé une plateforme de développement sous le système d'exploitation Windows 10.

3.1 Acquisition des données

Pour la collecte de données avons utilisé deux applications Android :

- **SensorRecorder** : Application installée sur le smartphone destiné aux participants, permettant de récolter les signaux bruts des capteurs, puis à les sauvegarder dans un fichier csv.

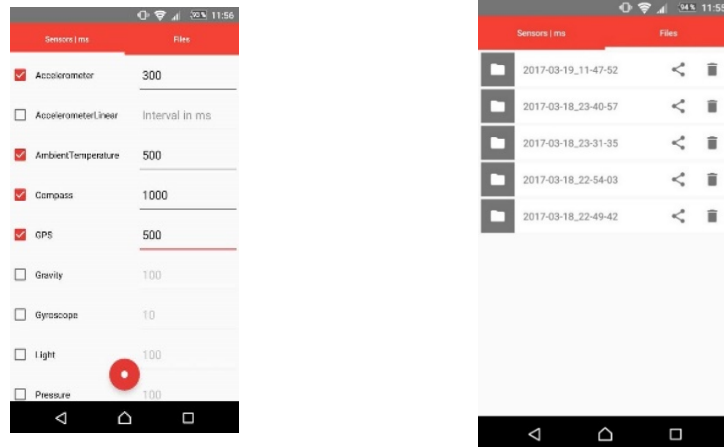


Figure 34 Capture d'écran de l'application

TimeStamp Camera Free : Application permettant de filmer tout en ajoutant un horodatage à la vidéo. Utilisée pour l'étiquetage.

À noter que les applications Android installées dans les deux téléphones doivent être lancées et arrêtées en même temps.

3.2 Développement du modèle

Le modèle a été construit en utilisant :

3.2.1 Google Colab

Google Colab est un service en nuage (cloud) offert par Google (gratuit), basé sur Jupyter Notebook, permettant d'améliorer les compétences de codage en langage de programmation Python. De développer des applications d'apprentissage automatique notamment approfondi à l'aide de bibliothèques populaires telles que Keras, TensorFlow, et Open CV.

Cette plateforme permet d'entraîner des modèles de Machine Learning directement dans le cloud.

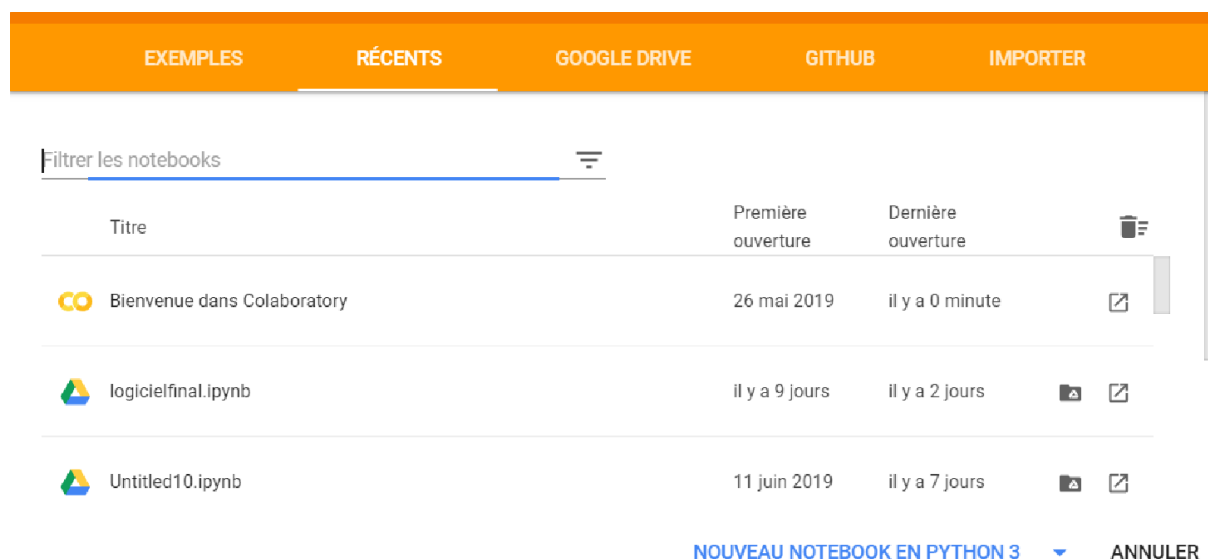


Figure 35. Environnement de Google Colab

3.2.2. Python 3.6

Python est un langage de programmation de haut niveau utilisé pour la programmation générale. Il fournit des constructions qui permettent une programmation claire à petite et à grande échelle. Python dispose d'un système de type dynamique et d'une gestion automatique de la mémoire.

Il prend en charge de multi-paradigme de programmation, y compris orientée objet, fonctionnelle, et dispose d'une bibliothèque standard vaste et complète. Les interpréteurs de Python sont disponibles pour de nombreux systèmes d'exploitation.



Figure 36 Logo Python

3.2.3 Tensorflow 1.13.1:

TensorFlow est une bibliothèque open source d'apprentissage automatique, créée par Google, permettant de développer et d'exécuter des applications de Machine Learning et de Deep Learning et regroupe un grand nombre de modèles et d'algorithmes de ces applications. Nous avons utilisé cette bibliothèque pour la reconnaissance d'activités humaines à partir des données de capteurs de smartphone récolté en temps-réelle.



Figure 37 Logo Tensorflow

- **Bibliothèques utilisées**

1. **Numpy** : Est une bibliothèque permettant d'effectuer des calculs numériques avec Python. Elle introduit une gestion facilitée des tableaux de nombres, des fonctions sophistiquées (diffusion).
2. **Matplotlib** : est une bibliothèque de traçage pour le langage de programmation Python et son extension mathématique numérique NumPy. Il fournit une API orientée objet permettant d'incorporer des graphiques dans des applications à l'aide de kits d'outils d'interface graphique à usage général tels que Tkinter, wxPython, Qt ou GTK +.
3. **Keras** : est une bibliothèque open source écrite en python et permettant d'interagir avec les algorithmes de réseaux de neurones profonds et de machine Learning, notamment Tensorflow.
4. **Pandas** : pandas est une bibliothèque open source sous licence BSD fournissant des structures de données hautes performances et faciles à utiliser, ainsi que des outils d'analyse des données pour le langage de programmation Python. Pandas est un projet sponsorisé par NumFOCUS. Cela contribuera au succès du développement de pandas en tant que projet open source de classe mondiale.

3. Développement de l'application :

3.1 Le langage de programmation Java :

Java est un langage de programmation orienté objet ,a été épuré des concepts les plus subtils du C++ et à la fois les plus déroutants, tels que les pointeurs et références, ou l'héritage multiple contourné par l'implémentation des interfaces. Une particularité de Java est que les logiciels écrits dans ce langage sont compilés vers une représentation binaire intermédiaire qui peut être exécutée dans une machine virtuelle Java (JVM) en faisant abstraction du système d'exploitation.



Figure 38. Logo Java

3.2 Android Studio:

Android studio est un environnement de développement pour développer des applications mobiles Android. Il permet principalement d'éditer les fichiers Java/Kotlin et les fichiers de configuration XML d'une application Android.

Il propose entre autres des outils pour gérer le développement d'applications multilingues et permet de visualiser rapidement la mise en page des écrans sur des écrans de résolutions variées simultanément. Il intègre par ailleurs un émulateur permettant de faire tourner un système Android virtuel sur un ordinateur.



Figure 39. Logo Android Studio

- l'API **SensorEventListener** : qui nous permet d'écouter les différents capteurs présents dans le périphérique.
- l'API **text-to-speech** : qui permet de convertir une chaîne de caractère en voix.
- l'API **tensorflow_inference** : qui nous permet d'exécuter des modèles tensorflow sur des appareils en temps réel, dans le but de faire des prédictions sur les données en entrées.

4. Test et évaluation

4.1 Critères d'évaluation

Afin d'évaluer notre système, nous avons fait appel aux différents critères de performances d'apprentissage cités dans le chapitre 2, donc la matrice de confusion serait interprétée comme suit :

- **Les vrais positifs (VP)** : Nombre d'activités bien prédites dans la classe à juste titre.
- **Les vrais négatifs (VN)** : Nombre d'activités prédites comme n'étant pas dans la classe à juste titre.
- **Les faux positifs (FP)** : Nombre d'activités prédites dans la classe alors qu'ils ne devraient pas en faire partie.
- **Les faux négatifs (FN)** : Nombre d'activités prédites comme étant de la classe alors qu'ils ne le sont pas en vrai.

4.2. Présentation des tests

Cette section décrit les résultats expérimentaux de notre système sur la reconnaissance des activités humaines.

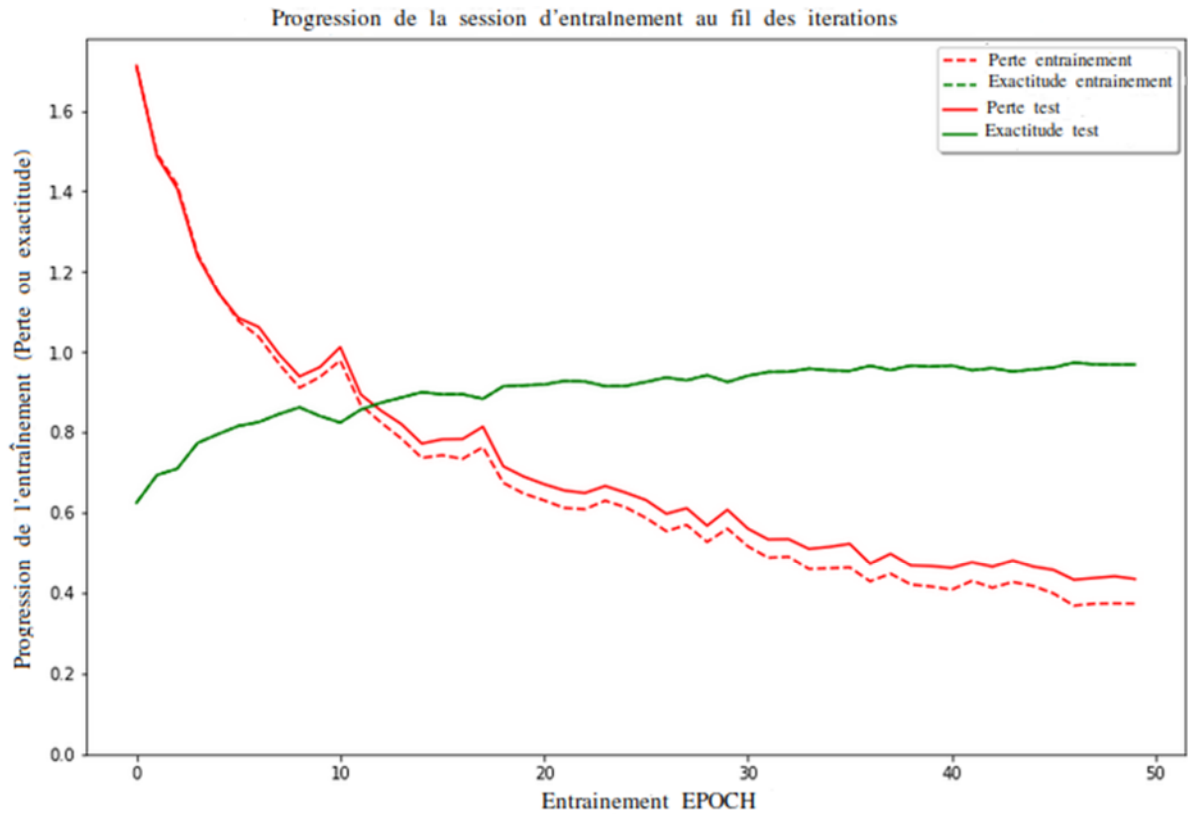


Figure 40 Graphe représentant le taux d'exactitude

Le système de classification développé a été évalué en termes de taux de bonne classification.

Ainsi, après avoir entraîné notre modèle, nous avons réussi à obtenir une exactitude de 95% avec une perte avoisinant les 0.4. Ce résultat est très encourageant et se compare favorablement à la littérature (voir figure X).

Pour une meilleure investigation des résultats, nous avons représenté la matrice de confusion du classifieur. La dernière colonne du tableau correspond au taux de classification par classe.



Tableau 4. Matrice de confusion du classificateur. La colonne de droite correspond aux pourcentages de classification correcte de chaque classe.

4.3. Discussions :

On peut remarquer que notre modèle se comporte plutôt bien dans la majorité des situations (5 activités sur 6 ont plus de 91% de justesse), à part quelques exceptions notables, on observe ainsi certaines erreurs de classification entre les activités : monter escaliers et descendre escaliers et vice versa. Cela est dû au fait que ces deux activités sont très ressemblantes.

De plus, vu l'exactitude élevée (95%), nous concluons que notre modèle ne souffre pas de problème d'under-fitting, ni d'over-fitting (deux phénomènes que nous expliquerons juste en dessous) car nous remarquons dans le graphe 38 que le training et test accuracy sont proches (de même pour training et test loss), ce qui prouve que l'apprentissage s'est bien déroulé.

- **Overfitting** : C'est ce qu'on appelle le sur-ajustement en Français, et c'est un problème courant en apprentissage automatique et en science des données. Il se produit lorsque le modèle de classification ne se *généralise* pas bien des données d'entraînement aux nouvelles données, autrement dit lors d'exécution de modèle sur un nouvel ensemble de données le taux d'exactitude sera petit.
- **Underfitting** : ou le sous-ajustement en Français, il se produit lorsqu'un modèle est trop simple et informé par trop peu de caractéristiques ou trop régularisé, ce qui le rend inflexible dans l'apprentissage de l'ensemble de données.

5. Implémentation du modèle sur une application Android

Dans cette partie, nous avons créé deux applications Android distinctes, la première utilise le modèle généré plus tôt, afin de collecter les données de l'utilisateur en temps réel, et de prédire l'activité de ce dernier (inférence). Et, la deuxième, sert plutôt de support à la première, en permettant d'afficher les graphes des différents capteurs utilisés. De ce fait, les utilisateurs pourront consulter, en temps réels, les valeurs des accéléromètres et du gyroscope.

Les deux applications ne nécessitent pas d'accès à internet, et marchent entièrement hors ligne, ce qui préserve la confidentialité des utilisateurs.

Concernant l'application principale, nous avons, dans un premier temps, sauvegardé notre modèle de reconnaissance d'activité, et cela en générant un dossier "Point de contrôle », contenant tous les paramètres utilisés par le modèle (valeurs des poids, prédiction, métadonnées.), et cela sous le format **protobuf** (format de sérialisation, permettant de convertir des informations complexes en une séquence d'octets).

Par la suite, en utilisant l'IDE Android Studio, nous avons développé une application en Java qui implémente notre modèle pré-entraîné. Ce dernier est stocké dans le dossier des ressources principales, avec toutes les bibliothèques 'Tensorflowinference interface', et est utilisé pour permettre l'inférence des activités (le réseau de neurones n'est utilisé que pour faire la prédiction) à partir des signaux bruts des capteurs du smartphone, à savoir l'accéléromètre, l'accéléromètre linéaire et le gyroscope, que nous avons récupéré à l'aide de la fonction **getSystemService**.

Le résultat final consiste en, l'affichage de la séquence de probabilité de chaque activité en temps-réel.

A noter, que nous avons rajouté une option de permettant d'épeler le résultat final, en utilisant l'API "text-to-speech" afin de faciliter l'utilisation de l'application. En effet, vu que le téléphone est censé se trouver dans la poche de l'utilisateur, il est préférable d'avoir le résultat sous forme sonore afin de ne pas sortir le smartphone de la poche pour voir l'affichage (pour ne pas biaiser les résultats).

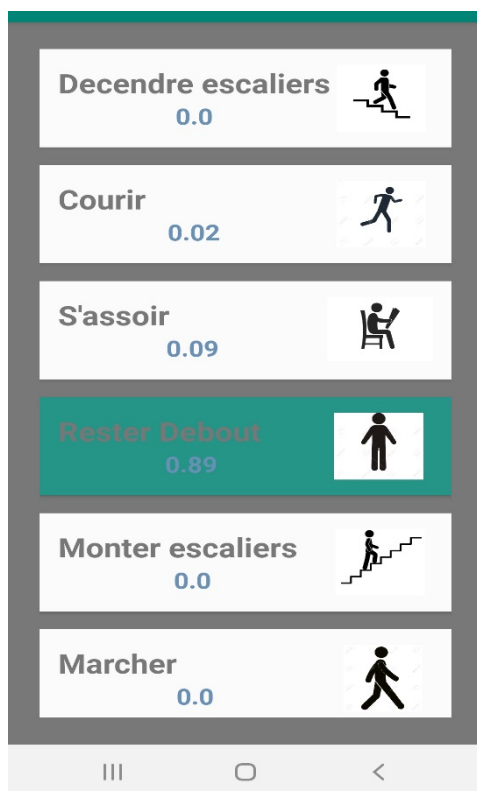


Figure 41 Capture d'écran de l'application (Activité: Rester debout)

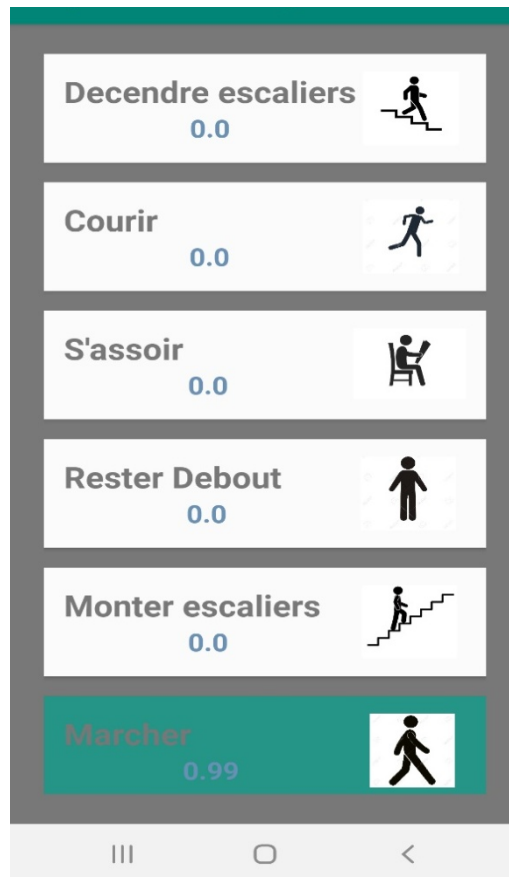


Figure 42. Capture d'écran de l'application (Activité: Marcher)

La deuxième application, quant à elle, a pour but de recueillir les signaux des trois (3) capteurs utilisés dans notre étude, à savoir l'accéléromètre, l'accéléromètre linéaire et le gyroscope, et d'afficher leurs valeurs en temps-réel sous forme de graphes.

De ce fait, l'utilisateur sera capable de vérifier en toute simplicité si les capteurs présents dans son smartphone sont opérationnels et que la capture se déroule bien (peu de bruits, graphe sans coupure...). De plus, il pourra visualiser les données afin de les étudier pour une utilisation future.



Figure 43 Capture d'écran de l'interface principale de l'application de visualisation

L'interface principale de cette seconde application (voir figure 43) permet à l'utilisateur de choisir quels capteurs il souhaite visualiser. Un exemple des interfaces secondaires affichant les graphes représentant les signaux des capteurs est donné à la figure 43.

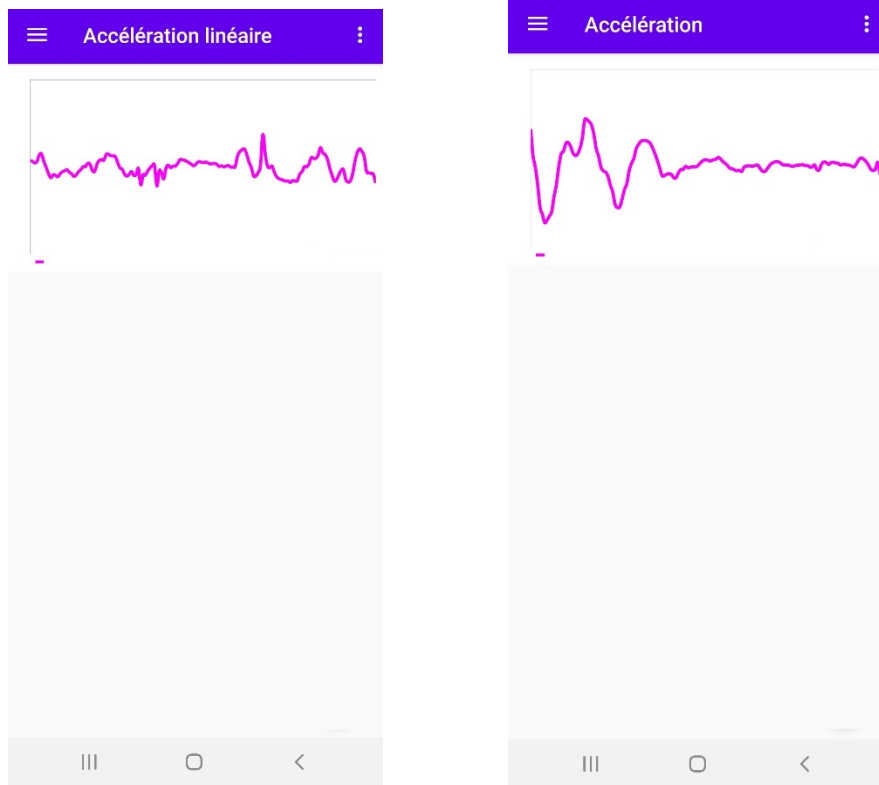


Figure 44. Capture d'écran des interfaces secondaires de l'application de visualisation

6. Conclusion

À travers ce dernier chapitre, nous avons exposé la conception de la solution que nous proposons face au problème posé à savoir : réaliser un système de classification automatique des activités humaines basé sur les capteurs d'un smartphone. Nous avons ainsi réorganisé notre DataSet pour qu'il corresponde à nos besoins, et qu'il serve de base d'entraînement et de test pour le classifieur. Puis nous avons présenté la réalisation de notre système de classification, et synthétiser l'ensemble des résultats des tests d'évaluation de notre méthode pour enfin l'implémenter dans une application Android.

Cette présentation inclut la structure modulaire du système, les outils utilisés pour le développement et l'implémentation de module d'acquisition des données ainsi que le module de génération du modèle de classification.

Conclusion Générale

Ce projet a permis de développer un système de classification automatique des activités physiques en utilisant un téléphone intelligent. Pour ce faire, nous avons collecté les données de capteur présent dans le smartphone, sur plusieurs participants.

La collecte de données a eu lieu principalement au niveau du stade polyvalent de Tizi-Ouzou et dans les locaux du département informatique de l'UMMTO (Université Mouloud Mammeri de Tizi-Ouzou) avec la participation de 20 personnes âgées de 13 à 82 ans. Chaque participant est appelé à faire, deux fois, un cycle contenant dans l'ordre les 6 activités physiques suivantes: monter et descendre l'escalier, marcher, courir, s'asseoir, et rester debout.

Nous avons par la suite développé un modèle de classification basé sur le deep learning, plus précisément, sur les réseaux neuronaux récurrents (RNN) et les réseaux de longue mémoire à court terme (LSTM). Le système de classification développé a été évalué en termes de taux de bonne classification.

Les résultats obtenus avoisinent les 95% d'exactitude. Ce qui représente, une très bonne performance comparée aux autres techniques de classification.

Enfin, nous avons exporté notre modèle, sur une application Android, permettant la détection des activités d'une personne en temps réel et ne nécessitant pas d'accès à internet.

Nous pouvons améliorer cette réalisation en élargissant notre base de donnée, qui est relativement petite pour le moment, puisque le nombre de sujets est restreint à 20 participants. Ainsi, en augmentant le nombre d'enregistrement, nous pourrions améliorer la précision de notre modèle, et augmenter le nombre d'activité prédite.

De plus, l'approche d'une application utilisant un modèle pré-entraîné, présente quelque défaut. Premièrement, le modèle de classification n'est pas facile à mettre à jour, car celui-ci est intégré dans l'application. De ce fait, si on souhaite améliorer le modèle existant, en le ré-entraînant avec des données supplémentaires, il nous faudrait télécharger et installer une nouvelle version. Ce qui peut poser un problème en termes de temps et de ressources pour l'utilisateur.

Deuxièmement, cette approche est très gourmande en terme d'espace mémoire (notre application fait plus de 80 Mb). Pour cela il serait judicieux de quantifier (en quelque sorte, arrondir) les poids de chaque couche afin d'avoir un nombre restreint de valeurs à sauvegarder. Ce qui permettrait de réduire le poids du modèle sans trop dégrader ses performances.

Cette étude démontre qu'il est possible d'utiliser les appareils mobiles pour discriminer les différentes activités faites par un sujet ce qui permettra d'améliorer la qualité de vie des personnes surtout les personnes âgées en terme de sécurité et de télésurveillance de ces personnes. Il suffit pour cela d'améliorer l'application Android pour répondre aux besoins attendus, que ce soit au niveau sécuritaire ou santé.

D'autres axes de recherches pourraient intéresser des chercheurs dans le domaine de classification automatique en utilisant d'autres dispositifs, à savoir, les montres intelligentes ou le textile intelligent, puisqu'ils contiennent, eux aussi, des capteurs tels que les accéléromètres. L'application Android utilisée lors de cette étude pourra aussi être compatible pour d'autres systèmes (iOS) et il est possible de rendre cette application plus compatible et plus adaptée aux besoins des futurs utilisateurs selon les activités physiques souhaitées.

Références bibliographiques

- [1] K. E. Powell, P. D. Thompson, C. J. Caspersen, et J. S. Kendrick, «Physical activity and the incidence of coronary heart disease», *Annual review of public health*, vol. 8, no1, p. 253–287, 1987
- [2] M. Chen, Y. Miao, Y. Hao, et K. Hwang, «Narrow band internet of things», *IEEE Access*, vol. 5, p. 20557–20577, 2017.
- [3] D. Riboni et C. Bettini, «Context-aware activity recognition through a combination of ontological and statistical reasoning», in *International Conference on Ubiquitous Intelligence and Computing*, 2009, p. 39–53.
- [4] R. M. Palmer, A. G. Ferrige, et S. Moncada, «Nitric oxide release accounts for the biological activity of endothelium-derived relaxing factor», *Nature*, vol. 327, no 6122, p. 524, 1987.
- [5] O. D. Lara et M. A. Labrador, «A survey on human activity recognition using wearable sensors», *IEEE Communications Surveys and Tutorials*, vol. 15, no 3, p. 1192–1209, 2013.
- [6] S. C. Mukhopadhyay, «Wearable sensors for human activity monitoring: A review», *IEEE sensors journal*, vol. 15, no 3, p. 1321–1330, 2015.
- [7] D. Preuveneers et Y. Berbers, «Mobile phones assisting with health self-care: a diabetes case study», in *Proceedings of the 10th international conference on Human computer interaction with mobile devices and services*, 2008, p. 177–186.
- [8] E. Kim, S. Helal, et D. Cook, «Human activity recognition and pattern discovery», *IEEE Pervasive Computing*, vol. 9, no 1, 2010.
- [9] M. Tambe et al., «Intelligent agents for interactive simulation environments», *AI magazine*, vol. 16, no 1, p. 15, 1995.
- [10] C. E. Zsombok et G. Klein, *Naturalistic decision making*. Psychology Press, 2014.
- [11] R. M. Palmer, A. G. Ferrige, et S. Moncada, «Nitric oxide release accounts for the biological activity of endothelium-derived relaxing factor», *Nature*, vol. 327, no 6122, p. 524, 1987.
- [12] O. D. Lara et M. A. Labrador, «A survey on human activity recognition using wearable sensors», *IEEE Communications Surveys and Tutorials*, vol. 15, no 3, p. 1192–1209, 2013.
- [13] S. C. Mukhopadhyay, «Wearable sensors for human activity monitoring: A review», *IEEE sensors journal*, vol. 15, no 3, p. 1321–1330, 2015.
- [14] D. Preuveneers et Y. Berbers, «Mobile phones assisting with health self-care: a diabetes case study», in *Proceedings of the 10th international conference on Human computer interaction with mobile devices and services*, 2008, p. 177–186.

- [15] E. Kim, S. Helal, et D. Cook, «Human activity recognition and pattern discovery», IEEE Pervasive Computing, vol. 9, no 1, 2010.
- [16] M. Tambe et al., «Intelligent agents for interactive simulation environments», AI magazine, vol. 16, no 1, p. 15, 1995.
- [17] C. E. Zsomboket G. Klein, Naturalistic decision making. Psychology Press, 2014.
- [18] J. K. Aggarwalet Q. Cai, «Human motion analysis: A review», Computer vision and image understanding, vol. 73, no 3, p. 428–440, 1999.
- [19] G. M. Weiss, J. L. Timko, C. M. Gallagher, K. Yoneda, et A. J. Schreiber, «Smartwatch-based activity recognition: A machine learning approach», in Biomedical and Health Informatics (BHI), 2016 IEEE-EMBS International Conference on, 2016, p. 426–429.
- [20] T. V. Duong, H. H. Bui, D. Q. Phung, et S. Venkatesh, «Activity recognition and abnormality detection with the switching hidden semi-markov model», in Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, 2005, vol. 1, p. 838–845
- [21] G. Kortuem, F. Kawsar, V. Sundramoorthy, et D. Fitton, «Smart objects as building blocks for the internet of things», IEEE Internet Computing, vol. 14, no 1, p. 44–51, 2010.
- [22] “Home is where the smart is”? Evaluating smart home research and approaches against the concept of home.
- Kirsten Gram-Hanssen, Sarah S.J. Darby
- [23] HAL Id:tel-01161610
- <https://tel.archives-ouvertes.fr/tel-01161610>
- [24] <https://www.planete-domotique.com/blog/2018/12/10/detecteur-mouvement-hue/>
- [25] HajarHiyadi. Reconnaissance 3D de gestes pour l’interaction homme-système. Traitement du signal et de l’image [eess.SP]. Université Paris-Saclay; Université d’Evry-Val-d’Essonne; Université Mohammed V de Rabat, 2016. Français. NNT: 2016SACLE052. tel-01804336
- [26] M. Humayun, Kabir M., RobiulHoque, KeshavThapa, Sung-Hyun YangTwo-Layer Hidden Markov Model for Human Activity Recognition in Home Environments.
- International Journal of Distributed Sensor Networks*, 12 (1) (2016)
- [27] L. Mo, F. Li, Y. Zhu and A. Huang, "Human physical activity recognition based on computer vision with deep learning model," 2016 IEEE International Instrumentation and

Measurement Technology Conference Proceedings, Taipei, 2016, pp. 1-6, doi: 10.1109/I2MTC.2016.7520541.

[28] Florian Adamsky ,Tatiana Retunskaja.WLAN Device Fingerprinting using Channel State Information (CSI)

[29] Yi He and Ye Li, "Physical activity recognition utilizing the built-in kinematic sensors of a smartphone", Hindawi Publishing Corporation, Volume 2013, Article ID 481580, 10 pages

[30] Oscar. D. Lara and M. A. Labrador, "A Survey on Human Activity Recognition using Wearable Sensors," inIEEE Communications Surveys & Tutorials, vol. 15, no. 3, pp. 1192-1209, Third Quarter 2013.

[31] Youssef Ouakrim. " Classification de sujets asymptomatiques et gonarthrosique en fonction des données cinématiques : Comparaison de l’approche globale et de l’approche locale“, thèse de maîtrise ÉTS 2011

[32] Dominique F(2010), Machine à vecteurs de support-Une introduction, CA MUS1, 7-25.

E. Lebarbier, T. Mary-Huard, Classification non supervisée, AgroParisTech 2008

[33] Francis B, K-means, EM, Mélanges de Gaussiennes, Théorie des graphes. Octobre 2010

[34] Marie C et Patrick L,Algorithme de Kohonen : classification et analyse exploratoire des données. Université de Paris-CNRS UMR 8595

[35] Robertas D, Mindaugas V, Justas S, Marcin W," Human activityrecognition in AAL environements using random projections",Hindawi Publishing Corporation, Volume 2016, Article ID 4073584, 17 pages.

[36] A.Mannini and A.M. Sabatini, “Machine learning methods for classifying human physical activity from on-body accelerometers,” Sensors, vol. 10, no. 2, pp. 1154–1175, 2010.

[37] <http://www.si.fr.atosorigin.comldatawarehouse/>

[38] Le Data Mining, René Lefébure et Gilles Venturi, Editions Eyrolles,Mars 2001, ISBN: 2-212-09176-1.

[39] https://fr.wikipedia.org/wiki/R%C3%A9seau_de_neurones_r%C3%A9currents

[40]https://fr.wikipedia.org/wiki/R%C3%A9tropropagation_du_gradient

[41]https://en.wikipedia.org/wiki/Long_short-term_memory

[42] (en) Gers, Felix, Schraudolph, Nicol et Schmidhuber, Jürgen, « Learning Precise Timing with LSTM Recurrent Networks », *Journal of Machine Learning Research*, vol. 3,p. ,2002 143-115

[43]https://rfiap2018.ign.fr/sites/default/files/ARTICLES/RFIAP_2018/RFIAP_2018_Gillot_Algorithmes.pdf