

الجمهورية الجزائرية الديمقراطية الشعبية
République algérienne démocratique et populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique
جامعة عين تموشنت بلحاج بوشعيب
Université –Ain Temouchent- Belhadj Bouchaib
Faculté des Sciences et de Technologie
Département Électronique et Télécommunication



Projet de Fin d'Etudes
Pour l'obtention du diplôme de Master en :
Domaine : Sciences et Technologie
Filière : Électronique
Spécialité : Instrumentation
Thème

**Reconnaissance de l'Activité Humaine à partir des Données
Smartphone**

Présenté Par :

- 1) Melle MOULKRALOUA Nour El Houda
- 2) Melle ZENAGUI Fadela

Devant le jury composé de :

Mme BOUTKHIL Malika	MAA	UAT.B.B (Ain Temouchent)	Présidente
Dr SEKKAL Mansouria	MCB	UAT.B.B (Ain Temouchent)	Examinatrice
Dr BENTAIEB Samia	MCB	UAT.B.B (Ain Temouchent)	Encadrante

Année Universitaire 2020/2021

Je dédie ce mémoire aux êtres les plus chers : mon père, ma mère. À ma belle sœur et mon meilleur frère et à toute la famille. À la mémoire de ma grand-mère et mon grand-père. Je tiens tout particulièrement à exprimer ma pleine gratitude et mes sincères remerciements pour tout leur soutien moral et leur aide depuis mon enfance jusqu'à la l'achèvement de ce mémoire. A tous mes cousins et cousines. A tous mes professeurs qui me donnent l'espoir et la force de continuer. Vous avez tous les éloges et la reconnaissance pour ce que vous avez fait pour moi. Merci de tout mon cœur. A tous ceux qui, de près ou de loin loin, m'ont aidé à réaliser ce mémoire en gage de ma sincère amitié, mon partenaire Zenagui Fadela.

Nour El Houda

Avec l'expression de ma reconnaissance, je dédie ce modeste travail à ceux qui, quels que soient les termes embrassés, je n'arriverais jamais à leur exprimer mon amour sincère. à maman, qui m'a donné les outils et les valeurs nécessaires pour être là où je suis aujourd'hui. à mes chères soeur qui n'ont pas cessée de me conseiller, encourager et soutenir tout au long de mes études. Sans oublier ma partenaire houda pour son soutien moral, sa patience et sa compréhension tout au long de ce projet.

Fadéla

Remerciements

Nous tenons tout d'abord à remercier "ALLAH" le tout-puissant pour nous avoir donné la volonté, la force, la patience, le courage, la santé. . . pour mener à bien ce travail.

Nous tenons à remercier chaleureusement notre Dr BENTAIEB Samia pour le temps qu'elle a consacré et les conseils qu'elle nous a prodigués pour son encouragement sans son aide, ce travail n'aurait pas vu le jour.

Nous remercions également la présidente Mme BOUTKHIL Malika et l'examinatrice Dr SEKKAL Mansouria d'avoir accepté d'examiner notre travail. Veuillez trouver dans ce travail notre profonde gratitude à tous les enseignants qui ont collaboré à notre formation.

Nous tenons à exprimer nos vifs remerciements envers à Mr Sadra Rafatnia pour son aide.

Merci à tous ceux qui ont participé à la réalisation de ce modeste document.

Table des matières

Table des figures	vi
Liste des algorithmes	vii
Glossaire	ix
1 Le Machine Learning pour la Classification de l'Activité Humaine	7
1.1 Introduction	8
1.2 Machine learning	9
1.2.1 Taxonomie des algorithmes de ML	9
1.2.1.1 Apprentissage supervisé	9
1.2.1.2 Apprentissage non supervisé	10
1.2.1.3 Apprentissage semi-supervisé	11
1.2.2 Approches basés sur le Machine Learning	11
1.3 Aperçu sur la reconnaissance d'activités humaine	13
1.3.1 Activités humaine	15
1.3.2 La detection et la collecte de données	16
1.3.3 L'apprentissage automatique pour la reconnaissance de l'activité humaine	17
1.4 La reconnaissance de l'activité humaine : état de l'art	18
1.5 Conclusion	20
2 La Classification de l'Activité Humaine à partir de Données Smartphone	21
2.1 Introduction	22

2.2	Les smartphones comme capteurs portables	22
2.3	Les capteurs portables	23
2.3.1	L'accéléromètre	24
2.3.2	Le gyroscope	25
2.4	L'apprentissage supervisé pour la HAR	26
2.4.1	Introduction	26
2.4.2	K-Nearest Neighbors K-NN	27
2.4.3	L'algorithme Support Vector Machine SVM	28
2.5	Résultats expérimentaux	29
2.5.1	Un peu de traitement de signal	29
2.5.1.1	Le filtre médian	29
2.5.1.2	Le filtre de Butterworth	30
2.5.2	Base de données	31
2.5.3	Protocole d'évaluation	36
2.5.3.1	Split Dataset	36
2.5.3.2	Tableau de confusion	37
2.5.3.3	Taux de classification	37
2.5.4	Environnement de travail	37
2.5.5	Évaluation et discussions	39
2.6	Conclusion	40
3	Classification de l'Activité par le Deep Learning	41
3.1	Introduction	42
3.2	Deep Learning (DL)	42
3.2.1	Présentation de quelques types de DL	43
3.2.1.1	Fully Connected Neural Network (FCN)	43
3.2.1.2	Recurrent Neural Network (RNN)	43
3.2.1.3	Convolutional neural network (CNN)	43
3.2.1.4	Long Short-Term Memory (LSTM)	43
3.3	Les réseaux de neurones artificiels (ANN)	43
3.3.1	Neurone artificiel	44
3.3.2	Fonction d'activation	45
3.4	Convolutional Neural Network	46
3.4.1	Couche de convolution	46
3.4.2	Couche de Pooling	47

3.4.3	Dropout	48
3.4.4	Flatten	48
3.4.5	Couche entièrement connectée	48
3.5	La classification de l'activité humaine par le CNN sur Python	48
3.5.1	Apprentissage d'un réseau CNN	49
3.5.2	Évaluation d'un réseau CNN	51
3.6	Conclusion	51
4	Classification de l'Activité sur La Raspberry Pi4	52
4.1	Introduction	53
4.2	Réorganisation des données	53
4.3	Normalisation des données	53
4.4	Évaluation	54
4.5	Implémentation sur Raspberry Pi4	55
4.5.1	Présentation du Raspberry Pi4	55
4.5.2	Architecture de la Raspberry Pi4	56
4.5.3	Configuration de la Raspberry Pi4	57
4.6	L'application Sensorstream IMU+GPS	59
4.7	La classification de l'activité humaine dans la Raspberry	62
4.8	Conclusion	64
	Bibliographie	66

Table des figures

1.1	Type de Machine Learning	9
1.2	Pipeline du processus de reconnaissance des activités humaines avec ses quatre blocs principaux	14
2.1	(a) Accéléromètre (b) Accéléromètre dans smartphone	25
2.2	(a) Gyroscope (b)Gyroscope dans smartphone	26
2.3	Exemple de classification K-NN	28
2.4	Exemple de classification SVM	29
2.5	Exemple du filte median	30
2.6	Graphe du gain des filtres passe-bas de Butterworth d'ordres 1 à 5, avec fréquence de coupure.	31
2.7	Exemple des activités réalisées lors de la collecte des données expérimentales. De gauche à droite et de haut en bas, debout, assis, couché, marchant, descendant les escaliers et remontant les escaliers [Ortiz, 2015]	32
2.8	Exemple de signaux provenant de l'accéléromètre pendant la réalisation des activités marcher et monter les escaliers [Ortiz, 2015]	33
2.9	Répartition de la base de données UCI-HAR	36
2.10	Colaboratory (Exemple)	38
2.11	Tableau de confusion du SVM	39
2.12	Tableau de confusion du K-NN lorsque K=5	40
3.1	Deep neural network	42
3.2	Un réseau neuronal artificiel avec trois couches de base	44
3.3	Un seul neurone artificiel	45

3.4	Fonctions d'activation	46
3.5	Exemple du réseau de neurones convolutifs [Ahmed, 2018]	46
3.6	Opération de corrélation croisée unidimensionnelle. Les parties en bleu sont le premier élément d'entrée ainsi que l'élément de sortie du noyau utilisé dans lequel : $0 \times 1 + 1 \times 2 = 2$	47
3.7	1D MaxPooling avec une taille de fenêtre $l = 3$	47
3.8	Exemple des couches utilisé	50
3.9	L'historique de Colab durant la phase d'apprentissage	50
3.10	(a) Résultats de précision de l'apprentissage (b) Fonction de perte de l'apprentissage du modèle CNN pour 100 itérations	50
3.11	Tableau de confusion de l'architecture CNN	51
4.1	Les matrices de confusion de l'architecture CNN des deux modèles entraînés	54
4.2	Architecture du Raspberry Pi 4	56
4.3	Menu de configuration de Raspian	58
4.4	Illustration de la mise à jour du système	59
4.5	Le bureau du Raspberry Pi4	59
4.6	SensorstreamIMU+GPS	60
4.7	Choix des capteurs sur SensorstreamIMU+GPS	60
4.8	Configuration de flux	61
4.9	Gestion du flux	61
4.10	Envoie des données en ligne dans la Raspberry Pi4	62
4.11	Projet HAR dans la Raspberry Pi4	62
4.12	Quelque résultats du programmation Python avec la Raspberry Pi4	63
4.13	La classification par la Raspberry de deux activité (Walking, Stan- ding, Sitting)	64

Liste des tableaux

1.1	Classification des activités selon leur durée et leur complexité	15
1.2	Classification des activités par type	16
2.1	Capteurs pour smartphones pouvant être utilisés pour la détection de mouvement	22
2.2	Description des signaux bruts de l'expérience HAR	34
2.3	Description des variables dérivées des signaux bruts	35
2.4	Description des 561 caractéristiques	36
2.5	Tableau de confusion d'un problème à deux classes.	37
2.6	Taux de classification des algorithmes KNN et SVM	39
4.1	Comparaison entre différents modèles de Raspberry	56

Liste des algorithmes

1	Algorithme du plus proche voisins K-NN	28
---	--	----

Abréviations

AAL : **A**mbient **A**ssisted **L**iving
AI : **A**rtificial **I**ntelligence
AMI : **A**mbient **M**ue **C**ontroleur
ANN : **A**rtificial **N**eural **N**etworks
CNN : **C**onvolution **N**eural **N**etworks
DT : **D**ecision **T**ree
DL : **D**eep **L**earning
FCN : **F**ully **C**onected **N**etwork
FN : **F**alse **N**egative
FP : **F**alse **P**ositive
HAR : **H**uman **A**ctivity **R**ecognition
HMM : **H**idden **M**arkov **M**odels
KNN : **K**- **N**earst **N**eighbors
LSTM : **L**ong **S**hort **T**erm **M**emory
MVC : **M**odèle **V**ue **C**ontroleur
ML : **M**achine **L**earning
NB : **N**aive **B**ayes
RNN : **R**ecurrent **N**eural **N**etwork
RF : **R**andom **F**orest
ReLU : **R**ectified **L**inear **U**nits
SVM : **S**upport **V**ector **M**achine
TN : **T**rue **N**egative
TP : **T**rue **P**ositive
W : **W**alking

WU : **W**alking-upstairs

WD : **W**alking-down

Sit : **S**itting

Stan : **S**tanding

Layin : **L**aying

Résumé

La reconnaissance de l'activité humaine (HAR) classe l'activité d'une personne à l'aide de capteurs sensibles qui sont affectés par le mouvement humaine. Les utilisateurs et les capteurs des smartphones augmentent et les utilisateurs portent généralement leur smartphone avec eux. Ces faits rendent HAR plus important et plus populaire.

Ce travail se concentre sur la reconnaissance de l'activité humaine à l'aide de capteurs de smartphone en utilisant différentes approches de classification par apprentissage automatique et apprentissage profond. Les données extraites des capteurs d'accéléromètre et de gyroscope des téléphones intelligents sont classées afin de reconnaître l'activité humaine. Les résultats des approches utilisées sont comparés en termes d'efficacité et de précision. Pour l'évaluation de ces approches nous avons utilisé la base de donnée UCI HAR qui a plus de 800 citations depuis 2012.

Après avoir prouvé la fiabilité et la robustesse du modèle, la carte raspberry pi4 a été utilisée pour rendre le système embarqué.

Mots-clés : La reconnaissance de l'activité humaine (HAR), smartphone, Classification, K-NN, SVM, Deep learning DL, Convolutionnel Neurol Network (CNN), Raspberry.

ملخص

يصنف التعرف على النشاط البشري *HAR* نشاط الشخص باستخدام اجهزة استشعار حساسة تتاثر بحركة الانسان. يتزايد مستخدموا الهواتف الذكية و اجهزة الاستشعار و يحمل المستخدمون هواتفهم الذكية معهم بشكل عام. هذه الحقائق تجعل *HAR* اكبر و اكثر شعبية. يركز هذا العمل على النشاط البشري باستخدام مستشعرات الهواتف الذكية باستخدام مناهج تصنيف مختلفة عن طريق التعلم الآلي و التعلم العميق. يتم تصنيف البيانات المستخرجة من اجهزة استشعار التسارع و الجيروسكوب للهواتف الذكية من اجل التعرف على النشاط البشري تتم مقارنة نتائج الاساليب المستخدمة من حيث الكفاءة و الدقة. لتقييم هذه الاساليب، استخدمنا قاعدة بيانات *UCIHAR* التي تحتوي على اكثر من ٨٠٠ اقتباس منذ عام ٢٠١٢. بعد اثبات موثوقية و متانة النموذج، تم استخدام لوحة *RaspberryPi* لانشاء النظام المضمن.

الكلمات الرئيسية : التعرف على النشاط البشري *HAR* ، الهاتف الذكي ، التصنيف ،

HAR ، *CNN* ، *Raspberry* ، *DeepLearning* ، *Raspberry* ، *SVM* ، *KNN*

Abstract

Human Activity Recognition (HAR) classifies a person's activity using sensitive sensors that are affected by human movement. Smartphone users and sensors are increasing and users usually carry their smartphones with them. These facts make HAR more important and popular.

This work focuses on human activity recognition using smartphone sensors using different machine learning and deep learning classification approaches. Data extracted from accelerometer and gyroscope sensors of smartphones are classified to recognize human activity. The results of the approaches used are compared in terms of efficiency and accuracy. For the evaluation of these approaches we used the UCI HAR database which has more than 800 citations since 2012.

After proving the reliability and robustness of the model, the raspberry pi4 board was used to make the system embedded.

Keywords :Human activity recognition (HAR), smartphone, Classification, K-NN, SVM, Deep learning DL, Convolutional Neural Network (CNN), Raspberry.

Introduction générale

Contexte

Le numérique joue un rôle primordial dans toutes les sociétés humaines. Il est considéré comme une vitalité pour tous les domaines en l'occurrence : le commerce, l'industrie, l'éducation, l'économie, le tourisme, la culture, le patrimoine, le sport, la mode, les médias, etc. Les smartphones, qui font partie de cette technologie, sont devenus un outil indispensable au fonctionnement quotidien de notre monde d'aujourd'hui. Offrant un certain nombre de fonctionnalités qui vont au-delà des simples fonctions d'appel ou de messagerie. Selon les dernières statistiques, en Janvier 2020, on comptait 2.87 milliards d'utilisateurs de mobile dans le monde. Effectuant différentes tâches, ces appareils mobiles sont équipés de nombreux capteurs qui en font de véritables plateformes de détection capables d'extraire des informations pertinentes. L'un des scénarios les plus attrayants dans lequel l'information qui peut être exploitée, est la reconnaissance de l'activité humaine 'Human Activity Recognition' (HAR), où les données détectées par des capteurs de mouvement, par exemple un accéléromètre et un gyroscope, peuvent être analysés pour déduire l'activité physique actuelle de l'utilisateur.

Dans ce contexte, les activités humaines peuvent être intuitivement considérées comme des séquences de motifs récurrents dans les données brutes fournies par les capteurs du smartphone. De nombreux algorithmes HAR ont été présentés dans la littérature, mais leur application est souvent limitée à des scénarios d'application spécifiques, par exemple, la santé en ligne.

L'une des techniques les plus fiables de HAR est qui a été implémentée dans les API de Google pour Android agit malheureusement comme une boîte noire ne

fournissant ni un moyen de comprendre les mécanismes qui la sous-tendent, ni des informations intermédiaires pour superviser le déroulement de la reconnaissance processus.

Dans ce présent travail, nous présentons un cadre pour la reconnaissance de l'activité humaine à l'aide des données du téléphone intelligent.

Motivation

L'identification de l'action humaine et de son interaction avec l'environnement est un élément clé pour le développement du système intelligent susmentionné. C'est un domaine qui traite des problèmes générés dans l'intégration de la détection et du raisonnement, pour fournir des données contextuelles qui peuvent conférer un support personnalisé à travers une application. Par exemple, si l'on imagine une maison intelligente qui est équipée de différents capteurs et appareils et qui est capable de détecter le fonctionnement de tous les appareils et la présence de personnes dans la maison, il est possible de déduire les diverses activités effectuées par une personne à l'intérieur de la maison en fonction du signal des capteurs avec d'autres facteurs pertinents comme le domaine temporel et la date (comme une personne le matin est censée être dans la cuisine et la machine à café suggère que la personne prépare le petit-déjeuner). C'est pourquoi le HAR collecté peut être absorbé pour anticiper les demandes futures des personnes et peut répondre à leur objectif (attente) (par exemple, réglage automatique de la température, contrôle automatique de la lumière, etc.).

Dans le système de reconnaissance de l'activité humaine, il existe encore divers problèmes qui doivent être résolus, tels que la limitation de la batterie des capteurs portables, le problème de confidentialité concernant la surveillance continue des activités, la difficulté à exécuter HAR en temps et manque de systèmes entièrement ambiants capables d'atteindre les utilisateurs à tout moment.

Objective

Notre projet s'inscrit dans le cadre de la reconnaissance d'activité, un domaine qui offre de nombreux avantages et permet de nombreuses nouvelles applications telles que l'assistance aux personnes âgées.

L'objectif principal de notre projet est de reconnaître les activités humaines en analysant les données des capteurs du téléphone mobile. Plus précisément, nous devons créer un modèle capable de prédire ou de classer avec précision si une personne est en train de s'allonger, de marcher, de monter les escaliers, de descendre les escaliers, de s'asseoir ou de se tenir debout uniquement sur la base des données des capteurs du téléphone portable.

Structure

Notre travail est composé de cinq chapitres structurés comme suit le chapitre donne une introduction générale sur le sujet, décrivant le contexte ainsi que les principaux objectifs de cette recherche. Dans le chapitre [1](#), nous présentons d'abord une revue détaillée concernant la reconnaissance de l'activité humaine. Dans le chapitre [2](#), la classification de l'activité humaine à partir de données smartphone en utilisant quelques algorithmes de machine learning seront présentés et évalués sur la base de données UCI HAR. Dans le [3](#), nous proposons d'utiliser le deep learning pour la construction d'un modèle à partir de données d'apprentissage et l'évaluation sur des données de test tirés de la même base de données. Dans le chapitre [4](#) nous proposons un deuxième modèle de CNN, et on a utilisé Le nano-ordinateur Raspberry Pi est utilisé pour la classification la reconnaissance d'activité humaine.

Finalement, une conclusion générale est présentée.

Chapitre 1

Le Machine Learning pour la Classification de l'Activité Humaine

Sommaire

1.1	Introduction	8
1.2	Machine learning	9
1.2.1	Taxonomie des algorithmes de ML	9
1.2.2	Approches basés sur le Machine Learning	11
1.3	Aperçu sur la reconnaissance d'activités humaine	13
1.3.1	Activités humaine	15
1.3.2	La detection et la collecte de données	16
1.3.3	L'apprentissage automatique pour la reconnaissance de l'activité humaine	17
1.4	La reconnaissance de l'activité humaine : état de l'art	18
1.5	Conclusion	20

1.1 Introduction

L'essor des systèmes informatiques omniprésents dans notre environnement engendre une forte nécessité d'adopter de nouvelles approches en matière d'interaction entre l'homme et l'ordinateur. Les systèmes HAR peuvent contribuer efficacement au développement de ces approches en raison du caractère discret et omniprésent des modes d'interaction qu'ils fournissent [Schmidt et al., 1999]. Une intervention humaine est nécessaire dans de nombreux systèmes de prise de décision, généralement par le biais d'une interaction avec des dispositifs traditionnels tels que les claviers, les télécommandes, les interrupteurs ou les écrans tactiles [Weiser, 1993]. Ces mécanismes d'interaction deviennent insolubles compte tenu de la quantité d'appareils auxquels nous sommes exposés chaque jour. En outre, nous sommes maintenant confrontés à un nouveau défi en raison de l'accès facile à de vastes quantités d'informations provenant de sources multiples. Ils peuvent contribuer à réduire notre interaction exigeante avec les machines à un minimum. Cela est possible notamment parce que nous sommes toujours (involontairement ou non) en train de fournir un retour d'information à l'environnement par notre comportement et nos actions. Les nouveaux systèmes pourraient donc devenir plus cognitifs et transformer fondamentalement nos modes d'interaction avec eux [Cook and Das, 2012]. Par exemple, dans le domaine médical, la surveillance des signaux physiologiques chez l'homme, comme une augmentation de notre rythme cardiaque moyen quotidien ou de notre température corporelle, peut indiquer un état de santé émergent. Cette détection peut être possible même sans que le patient soit conscient de la situation ou de la nécessité d'une visite de contrôle chez le médecin.

Le HAR est un domaine actif qui vise à identifier les actions effectuées par une personne à partir d'un ensemble d'observations sur elle-même et sur son environnement.

L'intelligence artificielle (AI) est un domaine d'étude dans le champ de l'informatique qui s'intéresse au développement d'ordinateurs capables de s'engager dans des processus de pensée semblables à ceux des humains, tels que l'apprentissage, le raisonnement et l'auto correction. L'apprentissage automatique (Machine Learning) est un champ d'étude de l'intelligence artificielle qui donne aux ordinateurs la capacité de prendre des décisions intelligentes.

Dans ce chapitre, nous abordons le sujet du HAR en nous appuyant sur ses travaux les plus pertinents dans la littérature.

1.2 Machine learning

La Machine Learning (ML) est le domaine d'étude qui concerne la conception, le développement et l'évaluation de systèmes capables d'apprendre à partir de données. Dans de nombreuses situations courantes où nous devons, par exemple, accomplir une tâche particulière, ou peut-être faire une prédiction concernant un problème donné, il est possible de trouver des solutions par l'inspection et l'analyse d'observations antérieures présentant des caractéristiques similaires au problème traité. En d'autres termes, les systèmes du ML sont capables de prédire les actions futures sur la base des expériences passées ([Bishop, 2006];[Murphy, 2012]) .

Les données sont utilisées comme entrants du processus d'apprentissage et leur représentation est fondamentale pour la performance des systèmes de ML. Ils doivent décrire tout

52 errors11 warnings pour mieux prévoir les données futures de manière significative. La propriété qui permet de prédire correctement les échantillons non appris est connue sous le nom de généralisation.

1.2.1 Taxonomie des algorithmes de ML

Les algorithmes de ML ont été classés en fonction du type d'entrée utilisé pour l'apprentissage (voir la Figure 1.1) et de son résultat attendu. Dans cette section, nous décrivons les catégories les plus pertinentes.

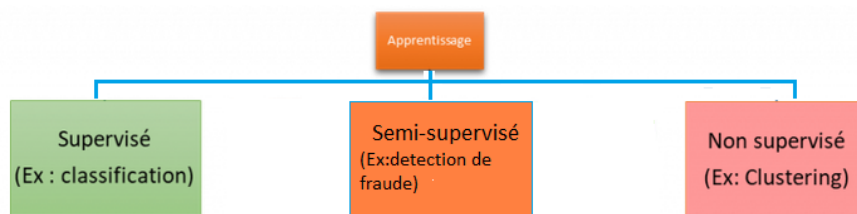


FIGURE 1.1 – Type de Machine Learning

1.2.1.1 Apprentissage supervisé

Dans ce type d'apprentissage, les données d'entrée sont généralement composées d'une paire d'éléments, à savoir le vecteur d'entrée (x) et sa cible (y) [Bishop, 2006].

Par exemple, décrivons un système qui apprend les chiffres manuscrits de 0 à 9. Les vecteurs d'entrée seraient l'ensemble des images de tous les nombres (généralement plusieurs échantillons pour chacun) et le vecteur cible sera des étiquettes réelles qui correspondent à chaque image. Si le résultat du système souhaité est catégorique (seul un ensemble de classes discrètes est pris en considération), il s'agit alors d'un problème de **classification**.

Si les données de sortie sont des variables continues, comme dans la prévision des températures ou des marchés boursiers, alors le système est considéré comme une **régression**. Ce type d'algorithme est le plus couramment utilisé pour le ML. Lorsque l'apprentissage est effectué progressivement, par exemple en ajoutant un nouvel échantillon et sa cible à la fois au modèle, on parle d'apprentissage automatique en ligne [Shalev-Shwartz et al., 2011]. Cette approche supervisée a l'avantage de rendre le modèle adaptable et flexible en fonction des nouveaux apports. Ce type d'apprentissage est requis dans les applications avec une grande variabilité de la production et où un flux de nouveaux échantillons est disponible et peut être progressivement ajouté au modèle d'apprentissage. C'est le cas des applications de classement en ligne et de prévision boursière.

1.2.1.2 Apprentissage non supervisé

Dans un problème d'apprentissage non supervisé, les données relatives à l'apprentissage se composent uniquement de vecteurs d'entrée (x) sans leurs cibles associées. Il vise à trouver certaines similitudes ou à découvrir une structure distincte dans les données d'entrée.

Des méthodes d'apprentissage non supervisées ont déjà été appliquées dans plusieurs domaines, notamment en imagerie médicale, où les scanners de Tomographie par Emission de Positrons (PET) 3D utilisent l'analyse des grappes pour trouver des dissimilitudes entre différents organes et types de tissus afin de pouvoir segmenter correctement la zone scannée [George et al., 2011]. Il a également été appliqué au regroupement automatique d'articles d'achat similaires, notamment dans les systèmes de recommandation pour les magasins en ligne qui visent à prédire les préférences de l'utilisateur en fonction des similitudes des produits et des achats précédents. Deux principales méthodes d'apprentissage non-supervisées existent à savoir le K-means et le regroupement hiérarchique. L'algorithme de K-means calcule les centroïdes et itère jusqu'à ce qu'il trouve le centroïde optimal. Dans cet algorithme, les

points de données sont assignés à un cluster de manière à ce que la somme de la distance au carré entre les points de données et le centroïde soit minimale.

1.2.1.3 Apprentissage semi-supervisé

Cette approche d'apprentissage combine des données étiquetées et non étiquetées pour l'apprentissage. Par conséquent, elle reprend des aspects des approches supervisées et non supervisées. En général, de petites quantités de données étiquetées sont intégrées à un grand nombre d'échantillons non étiquetés à des fins d'apprentissage. Par exemple, il est utile pour les ensembles de données où il n'est pas toujours possible d'avoir une étiquette pour chaque échantillon. Il a été prouvé que l'apprentissage semi-supervisé peut améliorer considérablement les performances d'apprentissage par rapport à l'apprentissage supervisé qui ne tient pas compte des données non étiquetées. Les vastes collections d'images numériques sur l'internet pour la recherche de contenu sont un exemple d'application où ce type d'apprentissage peut être exploité. Toutes les images n'ont pas de cible associée et il serait humainement impossible d'effectuer ce marquage manuellement [Fergus et al., 2009].

1.2.2 Approches basés sur le Machine Learning

Plusieurs approches de modélisation du ML ont été développées au fil des ans afin de résoudre différentes tâches telles que la classification, la régression et le regroupement [Bishop, 2006]. Certains d'entre elles sont basées sur des modèles déterministes qui visent à trouver des relations causales fixes entre les événements. D'autres approches, en revanche, sont probabilistes et supposent que les événements qui se produisent sont générés à partir d'une distribution de probabilité. Des combinaisons de ces approches ont également été explorées, comme dans [Franc et al., 2011]. Dans ce qui suit, une description des principaux algorithmes du ML est présentée.

- **Decision Tree (DT)** : est un modèle prédictif basé sur des arbres de décision qui fait des choix à partir d'un ensemble de règles hiérarchiques liées aux données d'entrée. Différentes versions ont été proposées, telles que ID3 et C5.4 [Quinlan, 1986][Quinlan, 1993]. Il s'agit d'une approche commune pour la classification, notamment parce que les modèles qui en résultent sont facilement interprétables par l'homme (en raison de sa structure arborescente).
- **Random Forest (RF)** : est un méta-classificateur ML qui est construit en

utilisant un ensemble de DT. La classe prédite est choisie comme la plus fréquente parmi les sorties de chaque DT [Breiman et al., 2001].

- **Naive Bayes (NB)** : est un classificateur probabiliste populaire basé sur le théorème de Bayes qui prédit la classe d'un échantillon donné en supposant un modèle de probabilité sous-jacent des données et en faisant de fortes hypothèses d'indépendance entre ses caractéristiques. Même si sa formulation est assez simple, il s'est avéré performant dans diverses applications [Bishop, 2006]. Par exemple, lorsque les données sont supposées être distribuées de manière Gaussienne, il est possible d'apprendre le modèle uniquement en calculant la moyenne et la variance des données d'entrée.
- **Artificial Neural Networks (ANN)** : est une approche de ML d'inspiration biologique. Elle simule la façon dont le cerveau et son système nerveux, composé de neurones inter connectés, sont capables d'apprendre par expérience et de saisir la structure sous-jacente des données. Les neurones sont placés dans une structure en couches et ont des poids associés qui sont capables de s'adapter en fonction des données d'entraînement et de la sortie du réseau grâce à une fonction de coût. Cette approche s'est avérée performante dans de nombreuses applications (par ex [LeCun et al., 1989]). Son principal inconvénient réside dans la nécessité de disposer d'un vaste ensemble de données pour sa phase d'apprentissage.
- **Support Vector Machines (SVM)** : est l'un des algorithmes de ML supervisé les plus couramment utilisés. Il a été initialement proposé par Vladimir Vapnik et ses collègues dans [Boser et al., 1992, Altman, 1992] dans le but de résoudre les problèmes de classification binaire linéaire et non linéaire. Le SVM est une approche déterministe qui vise à trouver les hyperplans qui séparent le mieux les données en classes. Ces sous-espaces sont ceux qui offrent la plus grande marge de séparation des classes de données d'apprentissage dans le but de fournir un modèle avec une faible erreur de généralisation pour son utilisation avec des échantillons de données invisibles.
- **K-Nearest Neighbors (K-NN)** : est une approche d'apprentissage déterministe exploite les mesures de similarité entre les données pour les tâches de classification et de régression. Dans le cas d'un nouvel échantillon, l'approche consiste à trouver les k échantillons les plus proches d'un ensemble d'apprentissage pour décider du résultat de la prédiction avec leurs valeurs (par exemple

en utilisant la règle de la majorité dans la classification ou la moyenne dans la régression)[Altman, 1992]. Son principal inconvénient réside dans la taille de son modèle, car il dépend des données et le rend inutilisable dans les grands ensembles de données. Il existe cependant des versions qui envisagent des techniques de réduction des données pour pallier ce problème.

1.3 Aperçu sur la reconnaissance d'activités humaine

Le HAR est un élément essentiel pour le développement de systèmes permettant des environnements cognitifs interactifs plus intelligents. A titre d'illustration, dans une vue simplifiée du pipeline du traitement de l'information humaine qui comprend quatre étapes (détection, analyse des données, prise de décision et action) ([Gandetto et al., 2003, Parasuraman et al., 2000]), le HAR appartient à la première et à la deuxième étape. Il contribue à l'acquisition des informations nécessaires concernant l'activité de l'utilisateur. Ces données sont ensuite combinées avec les données environnementales perçues afin d'obtenir une représentation complète de l'état du monde et de ses individus avant de passer aux étapes suivantes. Par exemple, un système de gestion des accidents chez les personnes âgées (par exemple, les chutes) nécessiterait initialement la détection d'un événement potentiel à partir des capteurs portés par l'utilisateur. Cette détection doit ensuite être fusionnée avec les informations environnementales qui peuvent aider à confirmer que ce qui s'est passé n'est pas un faux positif (par exemple, si le capteur a touché le sol). Ensuite, il est possible de prendre des décisions sur ce qu'il faut faire et, si nécessaire, d'agir en appelant les services d'urgence pour obtenir une aide immédiate. Dans cette section, nous présentons les principaux concepts qui sous-tendent le HAR et leur application aux problèmes du monde réel, avec un accent particulier sur les domaines de Ambient Assited Living (AAL) et de Ambient Mue Controleur (AMI). La Figure 1.2 présente une représentation générale des principales composantes d'une approche dédiée à la reconnaissance de l'activité humaine. Ces approches, trouvées dans la littérature, suivent une structure régulière avec de légères variations en fonction de leur application, des capteurs et des algorithmes ML sélectionnés. Le diagramme est valable pour les approches d'apprentissage supervisées, semi-supervisées ([Karantonis et al., 2006, Stikic et al., 2011, Wang et al., 2012]). Ils diffèrent selon le type d'entrée (avec

ou sans étiquette) et selon que le modèle appris se met à jour lorsque de nouveaux échantillons sont ajoutés dans le système (notez la ligne pointillée Feedback sur le graphique). A partir des quatre blocs principaux d'un système HAR (voir la Figure 1.2), la détection est chargée de rassembler les données des capteurs à partir des sources disponibles et de les traiter. En général, le conditionnement du signal (par exemple, la réduction du bruit, la numérisation, l'amplification) est toujours nécessaire pour adapter les signaux détectés aux exigences de l'application. En second lieu, le processus d'extraction de caractéristiques est chargé d'obtenir des caractéristiques significatives qui décrivent les données et permettent une meilleure représentation et compréhension des phénomènes étudiés. Les caractéristiques extraites deviennent l'entrée de l'algorithme ML, soit pour l'apprentissage du modèle, soit pour la prédiction de l'activité de nouveaux échantillons lorsque le modèle existe déjà.

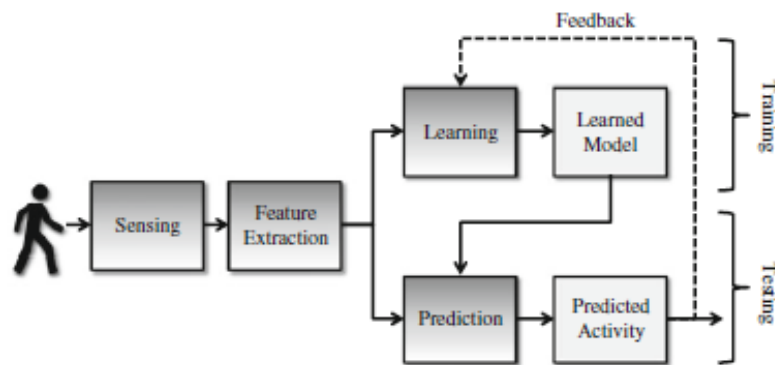


FIGURE 1.2 – Pipeline du processus de reconnaissance des activités humaines avec ses quatre blocs principaux

En outre, les systèmes HAR traditionnels fonctionnent généralement sur la base d'un feedback (retour), de sorte que l'apprentissage n'est effectué hors ligne qu'une seule fois et qu'il n'y a pas d'autre retour d'information dans le système. Cela est utile dans les cas où la distribution des données ne change pas au fil du temps ou lorsque le système est indépendant du sujet et résistant à une forte variabilité des entrées. Sinon, des méthodes adaptatives telles que l'apprentissage incrémental en ligne [Zheng et al., 2009] est conseillée mais conditionnées à une augmentation de la charge de calcul dans le processus.

1.3.1 Activités humaine

Un système HAR dépend de l'ensemble des activités à reconnaître car elles peuvent affecter directement la manière dont les systèmes sont conçus et mis en œuvre. Pour cette raison, la classification des activités en catégories simplifie la sélection des mécanismes appropriés pour les reconnaître. Ici, nous les classons en fonction de leur durée et de leur complexité, ainsi que du type d'activité. Le tableau 1.1 montre la catégorisation des activités en fonction de leur durée et de leur complexité. Il existe trois groupes principaux :

- Les événements de courte durée sont des activités simples d'une durée limitée définie. Ils se divisent en deux types. Premièrement, les gestes du corps qui sont des mouvements visibles utilisés principalement comme mécanisme de communication non verbale et deuxièmement, les transitions qui sont les événements qui relient l'exécution de deux activités différentes par exemple : lorsqu'une personne est assise puis se lève.

TABLE 1.1 – Classification des activités selon leur durée et leur complexité

Durée/ complexité	Type d'activité	Exemples
événement courts	geste transition	agiter les mains, hocher la tête , rire tenez-vous droit pour vous asseoir
activités de base	statique dynamique	debout, assis. marcher, courir, se déplacer à vélo
activité complexes	activité multiples utilisateur multiples	cuisiner, musculation parler, danser, jouer

Le deuxième groupe est composé des activités de base. Elles sont généralement caractérisées par une action continue ou cyclique. Leur durée est variable, mais en général, ils prennent beaucoup plus de temps que les événements courts (de l'ordre de quelques minutes). Deux types d'activités sont dérivés de cette classe : statiques qui impliquent des postures statiques telles que la position assise et debout et dynamiques qui sont des activités qui ont une périodicité comme la marche et la course (nous les appellerons ici activités d'ambulance).

- Le troisième groupe est celui des activités complexes. Il comprend le reste des activités qui sont généralement une séquence d'activités de base et des événements courts (activités multiples) ou qui impliquent plus d'un sujet (uti-

lisateur multiples) comme la danse de salon qui nécessite la participation et la coordination de deux personnes tout en exécutant plusieurs mouvements. Il existe actuellement de nombreuses applications d'intérêt public pour lesquelles le HAR est grandement nécessaire. Par exemple dans le domaine des soins de santé, pour l'adaptation des médicaments des patients handicapés ayant des problèmes moteurs grâce à la surveillance continue de la locomotion et des activités de la vie quotidienne à l'aide de capteurs inertiels, et dans le domaine de la sécurité, pour la surveillance des lieux publics et la prévention de la criminalité grâce à la détection des activités liées à la sécurité à l'aide de caméras vidéo([Avci et al., 2010, Lin et al., 2008]) . Il est ainsi possible de voir une catégorisation claire des activités en fonction de leur type. Ces groupes d'activités sont, en général, pertinents pour les diverses applications du HAR. Le tableau 1.2 présente les différents types d'activités qui ont été employées dans des études antérieures.

TABLE 1.2 – Classification des activités par type

Application	Exemples
la vie quotidienne	regarder la télévision, repasser, manger, nettoyer
locomotion	marcher, équitation, se tenir debout, s'allonger, tomber
sport/fitness	sauter, l'haltérophilie, l'escalade, nager
communication/connectivité	téléphoner, envoyer des sms, parler, signer
sécurité/surveillance	roder, poursuivre, surveiller, traquer

1.3.2 La détection et la collecte de données

La définition du dispositif expérimental pour l'acquisition de données est également un aspect important du HAR. En fonction de la façon dont le sujet est observé dans son habitat avec ou sans manipulation de l'observateur. Les environnements naturels sont idéaux pour l'expérimentation, mais dans de nombreux cas, il n'est pas possible de les exploiter. C'est pourquoi des expériences contrôlées peuvent être menées dans des conditions de laboratoire visant à simuler des environnements naturels (environnements semi-naturels). Sinon, les environnements entièrement contrôlés sont la dernière ressource pour l'acquisition de données, bien que les performances de la méthode/du système développé avec cette approche soient incertaines jusqu'à ce

qu'elles soient vérifiées dans des situations réelles. Les défaillances dans la conception des systèmes HAR peuvent être dues à l'absence de considérations réelles telles que les activités non comptabilisées ou les utilisateurs cibles, le bruit, l'étalonnage des capteurs et le positionnement, etc. Ces dernières sont par exemple fortement liées aux performances du système telles que présentées dans [Atallah et al., 2010, Maurer et al., 2006] où différents emplacements de capteurs ont été évalués pour déterminer les positions idéales pour effectuer la HAR grâce à l'utilisation d'accéléromètres portables. Une autre considération finale concernant le processus d'expérimentation est le nombre de personnes sélectionnées, car on préfère généralement un nombre plus important de personnes appartenant à des groupes d'âge et des conditions physiques différents. Cela est également directement lié à la performance et à la capacité de généralisation du système en présence de nouveaux utilisateurs.

1.3.3 L'apprentissage automatique pour la reconnaissance de l'activité humaine

Plusieurs approches de ML ont été développées au fil des années pour le HAR. Il a été principalement ciblé par des algorithmes d'apprentissage supervisé, bien que des méthodes semi-supervisées et non supervisées aient également été proposées ([Stikic et al., 2008, Altman, 1992]). Les modèles fréquentistes et Bayésiens ont été bien couverts dans toute la littérature du HAR. Ils impliquent des modèles basés sur des règles tels que DT et RF [Coley et al., 2005, Ermes et al., 2008], des approches géométriques incluant K-NN, ANN et SVM ([He and Jin, 2009, Khan et al., 2010, Maurer et al., 2006]), et des méthodes de classification probabilistes comme par exemple les classificateurs NB, et les modèles de Markov cachés (HMM) [Tapia et al., 2007, Zhu and Sheng, 2009]. Bon nombre de ces approches de ML ont démontré des performances comparables dans différents travaux (par exemple [Mannini and Sabatini, 2010]), ce qui suggère toutefois que l'efficacité et la bonne sélection des algorithmes peuvent être liées à d'autres aspects tels que la structure des données et l'application [Wolpert and Macready, 1997]. D'autres aspects pertinents pour la sélection d'algorithmes de ML comprennent : la consommation d'énergie, les besoins en mémoire, l'interprétabilité et la complexité de calcul, etc. À titre d'exemple, les arbres de décision pourraient être préférés lorsque l'interprétabilité du modèle est requise et les SVM pour les applications à haute performance.

1.4 La reconnaissance de l'activité humaine : état de l'art

Dans cette section, nous présentons quelques travaux de recherche dans le domaine de HAR. Les techniques de détection personnelles et omniprésentes telles que les smartphones ont permis de collecter des informations en continu de manière discrète [Garcia-Ceja et al., 2018]. Un nombre important de chercheurs ont effectué des recherches approfondies en explorant diverses techniques de détection et ont proposé un certain nombre pour modéliser et reconnaître les actions humaines.

L'utilisation des données d'accélération d'un smartphone est étudiée dans [Kwapisz et al., 2011], pour six activités humaines (marche, jogging, escaliers, descente d'escalier, assis et debout) en utilisant la régression logistique, et les algorithmes d'apprentissage multicouches du perceptron. Des techniques de reconnaissance d'activités utilisant le smartphone [Anjum and Ilyas, 2013, Antos et al., 2014] ont été proposées pour reconnaître les activités telles que marcher, courir, monter et descendre les escaliers, conduire, faire du vélo et être inactif. Les chercheurs ont pris différents emplacements du smartphone tels que la main, le sac à main, la poche du pantalon et la poche de la chemise. Ils ont évalué un certain nombre d'algorithmes de classification par apprentissage supervisé. Dans [Bayat et al., 2014], six schémas d'activité physique tels que la marche lente, la marche rapide, la course, l'ascension et la descente d'escaliers et la danse aérobique ont été évalués à l'aide du perceptron multicouche, SVM, RF, Logistic Model Tree (LMT), logistique simple et logitboost. Les données ont été recueillies à partir de capteurs de smartphones gardés en main ou dans la poche d'un pantalon. [Miao et al., 2015] ont collecté des données à partir de six positions telles que deux poches avant et deux poches arrière de pantalon et deux poches avant de manteau avec quatre orientations du smartphone pour cinq activités (statique, marche, course, montée et descend). Les auteurs ont utilisé des algorithmes de classification supervisés tels que DT, NV et l'optimisation séquentielle minimale (SMO) pour reconnaître les cinq activités humaines mentionnées. Un ensemble de méthodes de classification permettant de reconnaître les activités humaines quotidiennes telles que la marche, le jogging, la montée et la descente des escaliers, la position assise et debout a été proposé par [Catal et al., 2015]. [Tran and Phan, 2016] ont proposé un système permettant d'identifier six activités humaines telles que marcher, se tenir debout, s'asseoir, se coucher, monter et descendre les

escaliers à l'aide d'un smartphone. Dans ce travail, le modèle de classification du SVM a été utilisé. Cependant, les chercheurs n'ont pas mentionné la position du smartphone lors de la collecte des données. [Wang et al., 2016] ont classifié six activités humaines en utilisant le K-NN, où les données ont été collectées à partir des capteurs des smartphones. [Ravi et al., 2016] ont proposé une méthodologie d'apprentissage profond combinant les caractéristiques apprises à partir des données des capteurs inertiels avec des informations supplémentaires provenant d'un ensemble de caractéristiques peu profondes pour permettre une classification précise et en temps réel des activités telles que les transports publics, la station debout, le cyclisme, la marche, la course et les mouvements occasionnels. [Saha et al., 2017] ont proposé un cadre de reconnaissance d'activité en deux phases, en gardant le smartphone dans la poche du pantalon droit, la poche de la chemise, la ceinture et le sac, pour identifier certaines activités statiques comme s'asseoir, se tenir debout et certaines activités dynamiques comme marcher, monter les escaliers en utilisant le classificateur Logistic Regression (LR). Une méthode unidimensionnelle basée sur Convolutionnel Neural Network (CNN) pour reconnaître l'activité humaine en utilisant les données d'un accéléromètre triaxial collectées à partir d'un smartphone est proposée dans [Lee et al., 2017]. Les auteurs ont été amenés à faire trois activités (marcher, courir, rester immobile) avec la position du smartphone dans la main, le sac et la poche. En utilisant un ensemble de données étiquetées et non étiquetées, collectées à partir de smartphones et de smartwatches. Une méthode de reconnaissance de l'activité humaine a été proposée par [Mejia-Ricart et al., 2017] pour reconnaître les activités humaines (s'asseoir, se lever, courir, marcher, sauter, se coucher) en utilisant une méthode d'apprentissage non supervisée. [Nurhanim et al., 2017] ont proposé une étude comparative entre les différents noyaux d'une méthode de classification SVM multiclasse et toutes les autres pour six activités humaines (marcher, s'asseoir, se tenir debout, s'allonger, monter et descendre les escaliers) en utilisant les données des smartphones. [Cvetković et al., 2018] ont passé en revue les travaux de recherche récents sur les systèmes de surveillance de la santé mentale et ont examiné les activités de surveillance en temps réel (repos, position debout, couchée, assise, debout, mouvement, tâches, marche, nordique, course, vélo) avec les capteurs du smartphone et le bracelet riche en capteurs. [Bulbul et al., 2018] se sont concentrés sur la reconnaissance de six activités humaines telles que marcher, monter les escaliers, descendre les escaliers, s'asseoir, se lever et se coucher en utilisant les capteurs des smartphones. Les auteurs ont utilisé diverses méthodes de classification

telles que DT, SVM, K-NN et les méthodes de classification d'ensemble (boosting, bagging, stacking). Une structure bidirectionnelle de mémoire à long Short-Term Memory (LSTM) pour la reconnaissance de l'activité humaine utilisant des séries temporelles pour identifier six activités pendant qu'un mobile est porté à la taille d'un humain[Yu and Qin, 2018]. Une approche Ensemble Extreme Learning Machine (ELM) pour la reconnaissance de l'activité humaine utilisant des capteurs de smartphone pour les activités humaines telles que la marche, la marche rapide, la montée et la descente des escaliers, la course et la statique ont été classées. La position du smartphone étant dans la poche de pantalon, la poche de chemise et le sac à dos [Chen et al., 2019]. Dix activités humaines (s'asseoir, marcher, courir, se coucher, monter et descendre les escaliers, faire du vélo, se tenir debout, s'accroupir dans les toilettes et tomber) ont été classées à l'aide de quatre classifieurs différents tels que RF, SVM , K-NN dans [Barna et al., 2019]. Un classifieur multicouche basé sur le perceptron utilisant un accéléromètre, un gyroscope et les capteurs de gravité d'un smartphone pour reconnaître les activités physiques humaines (marcher, courir, s'asseoir, se tenir debout, monter et descendre des escaliers) a été proposé par[Voicu et al., 2019].

1.5 Conclusion

Dans ce chapitre, nous avons présenté l'état de l'art de la reconnaissance d'activité humaine. Nous avons commencé par une introduction sur le domaine, ensuite le machine learning et ces types d'apprentissage, puis un aperçu sur la reconnaissance d'activités humaine.

Chapitre 2

La Classification de l'Activité Humaine à partir de Données Smartphone

Sommaire

2.1	Introduction	22
2.2	Les smartphones comme capteurs portables	22
2.3	Les capteurs portables	23
2.3.1	L'accéléromètre	24
2.3.2	Le gyroscope	25
2.4	L'apprentissage supervisé pour la HAR	26
2.4.1	Introduction	26
2.4.2	K-Nearest Neighbors K-NN	27
2.4.3	L'algorithme Support Vector Machine SVM	28
2.5	Résultats expérimentaux	29
2.5.1	Un peu de traitement de signal	29
2.5.2	Base de données	31
2.5.3	Protocole d'évaluation	36
2.5.4	Environnement de travail	37
2.5.5	Évaluation et discussions	39
2.6	Conclusion	40

2.1 Introduction

Les technologies portables comprennent tous les dispositifs qui se portent sur le corps et permettent de recueillir et de traiter les informations des utilisateurs et leur interaction avec l'environnement.

2.2 Les smartphones comme capteurs portables

Dans le cadre de ce travail, nous utilisons les smartphones en tant que dispositifs portables, étant donné qu'ils sont désormais dotés de nombreux capteurs internes, dont certains peuvent être exploités pour la détection de mouvements et sont donc appropriés pour l'identification des activités humaines. Le tableau 2.1 présente ces capteurs. dont deux à savoir l'accéléromètre et le gyroscope. sont sélectionnés. Ils fournissent des informations sur l'accélération linéaire et la vitesse angulaire de l'utilisateur respectivement lorsqu'ils sont utilisés comme capteur portable, et ne sont pas très affectés par des facteurs externes tels que la mauvaise réception des signaux intérieurs dans les GPS ou le bruit électromagnétique dans les boussoles.

Cependant, les mesures des accéléromètres sont toujours influencées par la composante de gravité dans la détection de l'accélération des mouvements du corps (voir section 2.3.1). De même, nous travaillons avec les signaux d'accélération et de vitesse angulaire qui sont directement lus par ces capteurs et nous évitons leur intégration pour obtenir des informations de position ou d'orientation étant donné la dérive connue due au bruit que l'on trouve habituellement dans ce type de capteurs inertiels (Sect. 2.3.2).

TABLE 2.1 – Capteurs pour smartphones pouvant être utilisés pour la détection de mouvement

Capteur	Mesures/captures	Avantages/Inconvénients
accéléromètre	accélération linéaire	la composante de la gravité est toujours présente dans les mesures. l'estimation de la position 3D nécessite une double intégration,est donc elle susceptible.
gyroscope	vitesse angulaire	l'orientation de l'angle est estimée par l'intégration de la vitesse angulaire et est sujette à la dérive due au bruit du signal.

Les accéléromètres sont inclus dans les smartphones depuis 2007 [Lane et al., 2010], tandis que les gyroscopes ont été introduits plus récemment (2010) dans les appareils de milieu et haut de gamme et ils ont démontré qu'ils amélioraient les performances de reconnaissance des systèmes HAR lorsqu'ils sont utilisés en combinaison avec des accéléromètres comme dans [Anguita et al., 2013],[Wu et al., 2012]

2.3 Les capteurs portables

Les capteurs portables sont utilisés pour recueillir les signaux directement des utilisateurs. Ils sont généralement fixés à différentes parties du corps telles que la taille, les poignets, la poitrine, les jambes et la tête [Ravi et al., 2005], mais peuvent également être fixés aux vêtements et intégrés à d'autres accessoires d'usage courant tels que les montres, les lunettes ou des téléphones portables [Brezmes et al., 2009]. Ils contiennent une unité de batteries qui fournit l'énergie nécessaire à un fonctionnement continu et, pour certains d'entre eux, une unité sans fil pour la transmission des données des capteurs lorsqu'elle est nécessaire en externe ou pour les interfacer avec d'autres appareils portés sur le corps. Les signaux physiologiques et de mouvement obtenus à partir de capteurs portables sont très instructifs pour le HAR. La température de la peau, le rythme cardiaque, le flux de chaleur, la conductivité, la position du système de positionnement mondial (GPS) et les mouvements du corps sont quelques exemples de variables qui peuvent être mesurées avec les technologies actuelles de capteurs portables [Yang and Yang, 2006]. Elles peuvent être pratiques, par exemple, dans les applications où la surveillance continue des patients est nécessaire.

Contrairement aux capteurs ambiants, les capteurs portables présentent des avantages en termes d'intimité et de zone de fonctionnement. Dans le premier cas, les utilisateurs sont moins réticents à les utiliser en tout lieu s'il n'y a pas de capture d'images ou de vidéo. Deuxièmement, étant donné que ces capteurs sont toujours portés par l'utilisateur, ils sont omniprésents et leur couverture de localisation n'est pratiquement pas limitée. Ils ont également l'avantage d'être très portables et ne nécessitent pas d'équipement fixe. D'autre part, les capteurs portables ont également apporté de nouveaux défis : préserver la durée de vie des piles tout en étant capable de collecter des informations fiables à partir d'une détection limitée. Ces capteurs sont parfois inconfortables pour l'utilisateur courant (par exemple s'ils sont trop serrés ou câblés ou s'ils doivent être constamment repositionnés après l'habillage) et ne

peut fournir une solution à long terme pour la surveillance de l'activité sans rechargement régulier. En outre, les approches de détection hybrides, qui combinent des capteurs portables et ambiants provenant de différentes sources, offrent une option alternative robuste pour le HAR. Par exemple, dans [Tapia et al., 2006], un environnement riche en capteurs a été mis en place pour la collecte de signaux provenant de 72 capteurs environnementaux et corporels visant à évaluer des activités complexes dans un lieu intérieur.

Dans ce travail, nous utilisons des accéléromètres et des gyroscopes pour la classification des mouvements du corps humain. Nous décrivons leurs principales caractéristiques dans ce qui suit.

2.3.1 L'accéléromètre

L'accéléromètre est un instrument qui mesure l'accélération physique subie par un objet. Il a été utilisé pour plusieurs applications dans les domaines de la science, de la médecine, de l'ingénierie et de l'industrie, telles que la mesure des vibrations des machines, l'accélération des véhicules à grande vitesse et les charges mobiles sur les ponts.

Concernant le HAR, l'accéléromètre est l'un des capteurs les plus utilisés pour lire les signaux de mouvement du corps [Mannini and Sabatini, 2010]. Son principe de fonctionnement consiste généralement en une masse sismique qui se déplace en fonction de l'accélération à laquelle elle est soumise. Ce déplacement peut ensuite être traduit en un signal électrique mesurable. Ce phénomène a été appliqué pour le développement des capteurs des systèmes micro-électromécaniques (MEMS). Leur technologie permet de créer des dispositifs à l'échelle nanométrique fabriqués avec des semi-conducteurs. Elles sont avantageuses par rapport aux autres technologies de capteurs car il est possible de les produire à grande échelle et avec de faibles coûts de fabrication. Les accéléromètres MEMS les plus courants fonctionnent comme un capteur capacitif composé d'un faisceau en porte-à-faux avec une masse d'épreuve dont la déflexion est corrélée à l'accélération subie par le capteur ([Yang and Hsu, 2010]).

La magnitude et la direction de l'accélération peuvent être mesurées sous forme de quantité vectorielle en disposant orthogonalement les capteurs dans les trois dimensions spatiales. On peut également construire ces capteurs sur une seule puce et il est maintenant courant de trouver des accéléromètres triaxiaux dans plusieurs

appareils électroniques commerciaux.

L'un des problèmes que pose l'utilisation d'accéléromètres pour la détection des mouvements du corps est l'effet du champ gravitationnel qui est toujours présent dans les mesures et dont la magnitude ($g = 9,81m/s^2$) est relativement élevée. Cependant, il peut également être séparé du mouvement du corps par filtrage. La détection du vecteur de gravité peut également aider à déterminer l'orientation d'un objet par rapport à l'axe de la force gravitationnelle lorsque des accéléromètres triaxiaux sont utilisés 2.1.

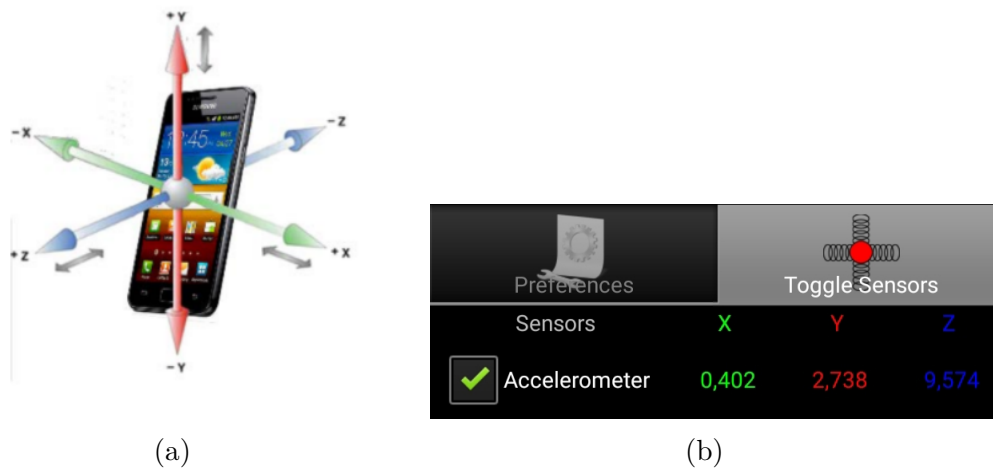


FIGURE 2.1 – (a) Accéléromètre (b) Accéléromètre dans smartphone

2.3.2 Le gyroscope

Le gyroscope est un dispositif de détection permettant de mesurer l'orientation [Woodman, 2007]. Il a été utilisé dans de nombreuses applications telles que les systèmes de navigation inertielle, les véhicules aériens pour augmenter la stabilité (par exemple dans les quadricoptères) et récemment, il a été introduit dans des dispositifs électroniques (par exemple les smartphones, les consoles de jeu) pour améliorer les interfaces utilisateur et expérience de jeu. Pour le HAR, ce capteur a été utilisé dans diverses applications telles que la classification activités (par exemple, la marche, la montée d'escaliers) et les transitions entre les postures (par exemple, de la position debout à la position assise) ([Coley et al., 2005, Altman, 1992]).

Des gyroscopes ont également été produits avec des technologies MEMS. Cependant, les capteurs de ce type ne peuvent mesurer l'orientation qu'indirectement. Ils

estiment plutôt la vitesse angulaire qui peut ensuite être intégrée dans le temps afin d'obtenir l'orientation. Cependant, il faut d'abord disposer d'une position angulaire initiale de référence pour y parvenir. Ces capteurs sont également très sensibles au bruit, ce qui peut entraîner une dérive de la mesure due à l'intégration.

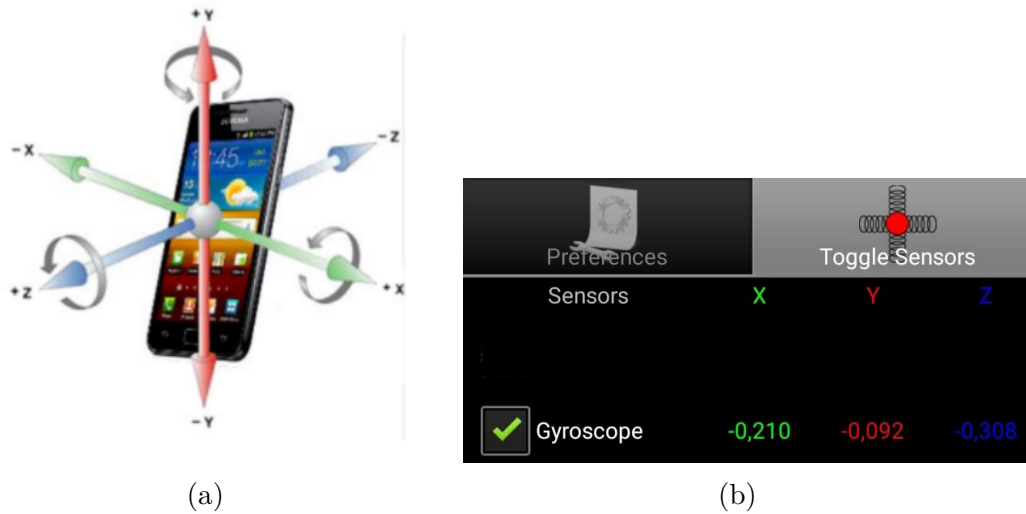


FIGURE 2.2 – (a) Gyroscope (b)Gyroscope dans smartphone

Les directions des axes sont indiquées sur la figure 2.2. Les données brutes d'un gyroscope sont la vitesse de rotation en (radian par seconde)rad/s.

2.4 L'apprentissage supervisé pour la HAR

2.4.1 Introduction

La reconnaissance de l'activité humaine est le problème de classification des séquences de données enregistrées par des smartphones en mouvements connus et bien définis. Les approches classiques impliquent l'élaboration manuelle de caractéristiques à partir des observations et l'application de modèles d'apprentissage automatique, ce qui est très différent car nous devons avoir une grande expertise pour acquérir des caractéristiques à partir de données. Plutôt que cela, nous pouvons opter pour des algorithmes d'apprentissage supervisé tels que le K-NN et le SVM.

2.4.2 K-Nearest Neighbors K-NN

L'algorithme du plus proche voisin est l'un des algorithmes d'apprentissage automatique les plus simples. Il s'agit d'une technique non paramétrique. Avant de détailler l'algorithme, quelques définitions sont nécessaires.

La **distance** entre deux objets est considérée comme la distance Euclidienne entre eux. Pour calculer cette distance, chaque objet doit être représenté par un vecteur dans un espace multidimensionnel.

Les vecteurs \vec{x} (x_1, x_2, \dots, x_p) et \vec{y} (y_1, y_2, \dots, y_p) sont deux échantillons d'entrée (objets) avec p caractéristique.

La distance Euclidienne entre l'échantillon \vec{x} et l'échantillon \vec{y} est défini comme dans l'équation 2.1.

$$d(\vec{x}, \vec{y}) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_p - y_p)^2} \quad (2.1)$$

Un objet est le **voisin** d'un autre objet si la distance qui les sépare est inférieure à un seuil prédéfini.

Le **plus proche voisin** d'un objet \vec{x} est l'objet échantillon dont la distance à \vec{x} est la plus faible parmi tous les échantillons d'entrée.

Le 2^{eme} **plus proche voisin** d'un objet \vec{x} est l'objet échantillon dont la distance à \vec{x} est la deuxième plus faible parmi tous les échantillons d'entrée.

Le n^{eme} **plus proche voisin** est défini de manière analogue.

les **K plus proches voisins** d'un objet \vec{x} sont la collection d'objets échantillons x_i où $i = 1, 2, \dots, K$.

Les étapes de l'algorithme du plus proche voisin peuvent être décrites comme suit :

Algorithm 1 Algorithme du plus proche voisins K-NN

- 1: **Étape 1** : sélectionner le nombre K des voisins.
 - 2: **Étape 2** : calculer la distance Euclidienne du nombre K de voisins.
 - 3: **Étape 3** : prendre les K voisins le plus proches selon la distance Euclidienne calculée.
 - 4: **Étape 4** : parmi ces les voisins, compter le nombre de points de données dans chaque catégorie.
 - 5: **Étape 5** : attribuer les nouveaux points de données à cette catégorie pour laquelle le nombre du voisin est maximum.
 - 6: **Étape 6** : notre modèle est prêt.
-

Dans l'exemple de la Figure 2.3, l'échantillon test (cercle vert) doit être classé soit dans la classe "carré bleu", soit dans la classe "triangle rouge". Si $K = 3$, il est classé dans la classe "triangle rouge" parce qu'il y a 2 triangles et seulement 1 carré à l'intérieur du cercle intérieur. Si $K = 5$ il est classé dans la classe "carré bleu" (3 carrés contre 2 triangles à l'intérieur du cercle extérieur).

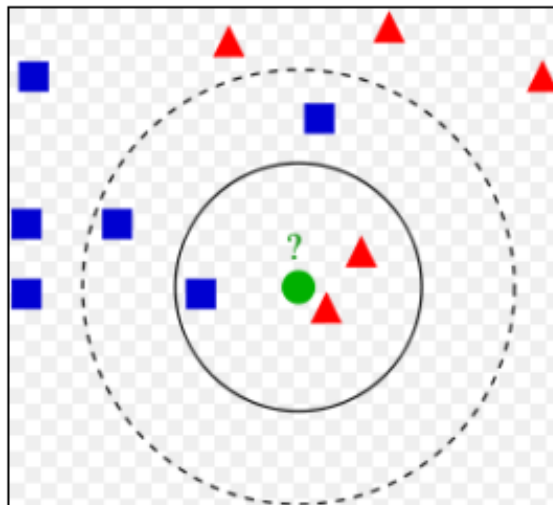


FIGURE 2.3 – Exemple de classification K-NN

2.4.3 L'algorithme Support Vector Machine SVM

Les machines à vecteurs de support (SVM) sont l'une des méthodes d'apprentissage supervisé les plus efficaces pour ce problème. Elles ont des fondements théoriques solides et ont montré d'excellents résultats dans divers domaines. Les machines à vecteurs de support sont entraînées de manière à ce que la fonction de décision

classe les données d'exemple non vues avec précision. Cette capacité à classer avec précision des données d'exemple non vues est appelée généralisation. Une capacité de généralisation élevée est l'une des principales raisons du succès des SVM. Le classifieur utilisé dans la classification par vecteurs de support est l'hyperplan de séparation. L'hyperplans qui séparent les deux classes peuvent différer par la marge qu'ils offrent entre eux et les points les plus proche comme dans la 2.4.

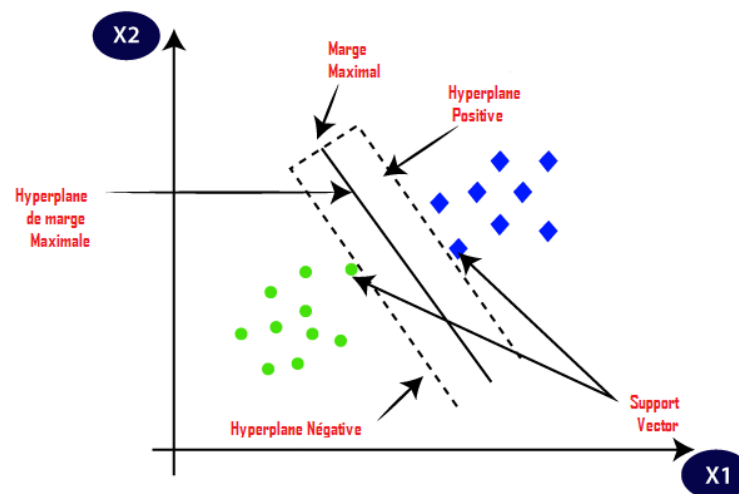


FIGURE 2.4 – Exemple de classification SVM

2.5 Résultats expérimentaux

Dans cette section, quelques notions de traitement de signal, la base de données, le protocole d'évaluation ainsi que les deux algorithmes utilisés et leurs résultats respectifs seront présentés.

2.5.1 Un peu de traitement de signal

2.5.1.1 Le filtre médian

Le filtre médian est une technique de filtrage numérique non linéaire, souvent utilisée pour éliminer le bruit d'une image ou d'un signal. Une telle réduction du bruit est une étape typique de prétraitement pour améliorer les résultats d'un traitement ultérieur.

Le filtre médian est un filtre statistique qui utilise une fenêtre glissante comportant un nombre impair de points sur le signal . Le point central de la fenêtre est remplacé par la médiane des valeurs du signal sur toute la fenêtre.

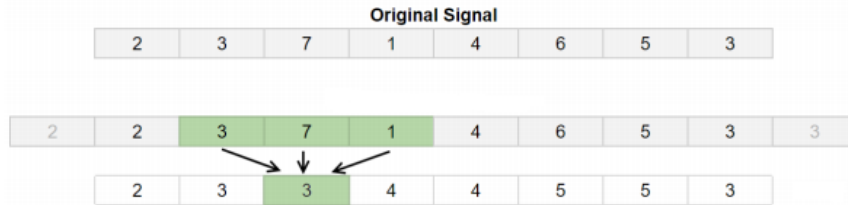


FIGURE 2.5 – Exemple du filte median

Dans la Figure 2.5 , nous donne une démonstration du processus de calcul de la médiane d'une séquence, à l'aide d'un signal échantillon généré de façon aléatoire comme entrée. La fenêtre traitée de taille 3 par le filtre médian est surlignée en vert, et la ligne du bas représente la sortie du filtre, qui est la médiane des valeurs de la fenêtre lorsqu'elle glisse sur le signal original de gauche à droite. Les valeurs ajoutées au début et à la fin du signal d'entrée sont grisées.

2.5.1.2 Le filtre de Butterworth

Le filtre de Butterworth est un type de filtre de traitement du signal conçu pour avoir une réponse en fréquence aussi plate que possible dans la bande passante. Il est également appelé filtre à amplitude maximale plate.

La réponse en fréquence du filtre de Butterworth est extrêmement plate (c'est-à-dire qu'elle n'a pas d'ondulation) dans la bande passante et diminue vers zéro dans la bande d'arrêt. Sur un diagramme de Bode logarithmique, la réponse diminue linéairement vers l'infini négatif. Les filtres de Butterworth ont une fonction d'amplitude variant de façon monotone avec ω , contrairement aux autres types de filtres qui présentent une ondulation non monotone dans la bande passante et/ou la bande d'arrêt.

Comme tous les filtres, le prototype typique est le filtre passe-bas. Le gain $G(\omega)$ d'un filtre passe-bas de Butterworth d'ordre n est donné en termes de fonction de

transfert H comme suit :

$$G^2(\omega) = |H(j\omega)|^2 = \frac{G_0^2}{1 + \left(\frac{j\omega}{j\omega_c}\right)^{2n}} \quad (2.2)$$

Où :

n = ordre du filtre

ω_c = fréquence de coupure (approximativement la fréquence de -3dB)

G_0 : est le gain en courant continu (gain à la fréquence zéro)

on peut voir que lorsque n s'approche de l'infini, le gain devient une fonction rectangle et les fréquences inférieures à ω_c seront passées avec un gain G_0 , tandis que les fréquences supérieures à c seront supprimées. Pour des valeurs plus petites de n , la coupure sera moins nette.

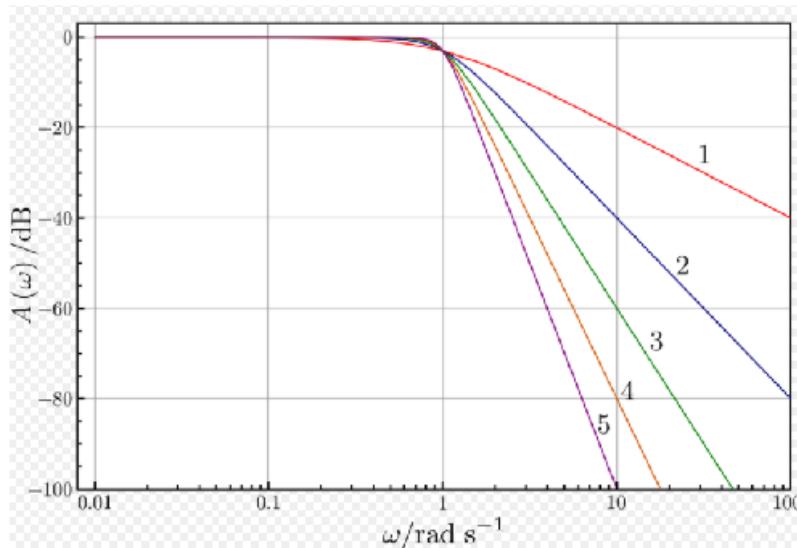


FIGURE 2.6 – Graphe du gain des filtres passe-bas de Butterworth d'ordres 1 à 5, avec fréquence de coupure.

La Figure 2.6 représente gain des filtres passe-bas de Butterworth d'ordres 1 à 5.

2.5.2 Base de données

Pour l'évaluation des algorithmes K-NN et SVM, nous avons utilisé la base de données UCI-HAR générée lors International Workshop of Ambient Assisted Living (IWAAL) qui s'est tenu en Espagne en 2012 [Anguita et al., 2012]. Formée avec un groupe de 30 volontaires dans une tranche d'âge de 19 à 48 ans, chacun a réalisé six

activités (marcher, marcher en haut, marcher en bas, s’asseoir, se lever, s’allonger) en portant un smartphone (Samsung Galaxy S II) à la taille comme illustré dans la Figure 2.7. Avec l’accéléromètre et le gyroscope intégrés, l’accélération linéaire à 3 axes et la vitesse angulaire à 3 axes à une fréquence constante de 50 Hz. Les expériences ont été enregistrées sur vidéo pour étiqueter les données manuellement.



FIGURE 2.7 – Exemple des activités réalisées lors de la collecte des données expérimentales. De gauche à droite et de haut en bas, debout, assis, couché, marchant, descendant les escaliers et remontant les escaliers [Ortiz, 2015]

Les signaux du capteur (accéléromètre et gyroscope) ont été prétraités à l’aide d’un filtre médian et d’un filtre Butterworth passe-bas de 3eme ordre avec une fréquence de coupure de 20 Hz. Les signaux traités ont été échantillonnés dans une fenêtre fixe d’une longueur de 2,56 secondes avec un chevauchement de 50 %. Chaque fenêtre comportait 128 points de données pour chacune des caractéristiques originales enregistrées, à savoir l’accélération du corps, le gyroscope du corps et l’accélération gravitationnelle sur les axes X, Y et Z. Les signaux inertiels fenêtrés ont été analysés et plusieurs caractéristiques temporelles et fréquentielles ont été

extraites de chaque fenêtre.

La Figure 2.8 montre quelques signaux provenant de l'accéléromètre pendant la réalisation des activités marcher et monter les escaliers.

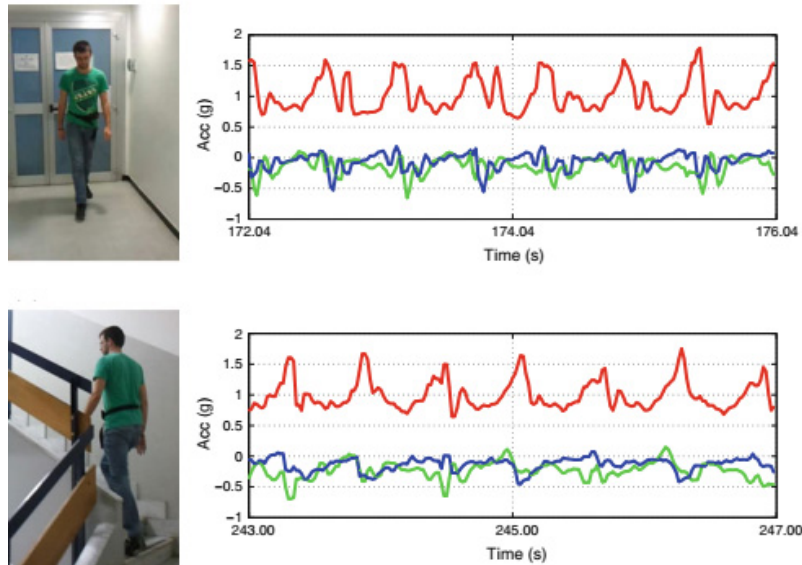


FIGURE 2.8 – Exemple de signaux provenant de l'accéléromètre pendant la réalisation des activités marcher et monter les escaliers [Ortiz, 2015]

Le signal d'accélération du capteur, qui a des composants gravitationnels et de mouvement corporel, a été séparé à l'aide d'un filtre passe-bas Butterworth en accélération corporelle et gravité. On suppose que la force gravitationnelle n'a que des composantes à basse fréquence, c'est pourquoi un filtre avec une fréquence de coupure de 0.3 Hz a été utilisé. Des vecteurs de caractéristique dans les domaines temporel et fréquentiel ont été obtenus comme le montre le tableau 2.2

TABLE 2.2 – Description des signaux bruts de l'expérience HAR

Le signal brute	Définition
tBodyAcc-XYZ	accélération du corps dans le temps
tGravityAcc-XYZ	accélération de la gravité dans le temps
tBodyAccJerk-XYZ	jerk ¹ dans l'accélération du corps dans le temps
tBodyGyro-XYZ	mesure du gyroscope du corps dans le temps
tBodyGyroJerk-XYZ	jerk dans le corps gyroscope mesurer dans le temps.
tBodyAccMag	magnitude de l'accélération du corps dans le temps
tGravityAccMag	magnitude de l'accélération de la gravité dans le temps
tBodyAccJerkMag	amplitude de jerk dans l'accélération du corps dans le temps
tBodyGyroMag	magnitude de la mesure du gyroscope du corps dans le temps
tBodyGyroJerkMag	mesure de l'amplitude du gyroscope corporel dans le temps
fBodyAcc-XYZ	accélération du corps en fréquence
fBodyAccJerk-XYZ	jerk dans l'accélération du corps en fréquence
fBodyGyro-XYZ	mesure du gyroscope du corps en fréquence
fBodyAccMag	magnitude de l'accélération du corps en fréquence
fBodyAccJerkMag	magnitude de jerk in body acceleration in fre-quency
fBodyGyroMag	magnitude de la mesure du gyroscope du corps en fréquence
fBodyGyroJerkMag	magnitude du jerk dans le gyroscope corporel mesure la fréquence

Les caractéristiques extraites de l'expérience ont été dérivées selon un processus élaboré et sont présentées dans le tableau 2.2 où "XYZ" désigne les directions des trois axes X, Y, Z pour chacun des signaux triaxiaux; "t" indique les variables du domaine temporel et "f" les variables du domaine fréquentiel. Les signaux bruts initialement obtenus, tAcc-XYZ et tGyro-XYZ, étaient les signaux triaxiaux obtenus à partir de l'accéléromètre et du gyroscope. L'accélération totale a ensuite été divisée en tBodyAcc-XYZ et tGravityAcc-XYZ. Les signaux Jerk, qui représentent le taux de variation de l'accélération dans le temps, sont dérivés des signaux bruts et sont désignés par tBodyAccJerk-XYZ et tBodyGyroJerk-XYZ. La norme Euclidienne est utilisée pour calculer l'amplitude de chacun des signaux, ce qui donne les composantes suivantes : tBodyAccMag, tGravityAccMag, tBodyAccJerkMag, tBodyGyroMag, tBodyGyroJerkMag.

En outre, une transformé de Fourier rapide a été appliquée pour produire les caractéristiques suivantes : fBodyAcc-XYZ, fBodyAccJerk-XYZ, fBodyGyro-XYZ, fBodyAccJerkMag, fBodyGyroMag, fBodyGyroJerkMag. Le tableau 2.3 énumère

l'ensemble des variables descriptives calculées pour chacune des caractéristiques brutes ci-dessus.

TABLE 2.3 – Description des variables dérivées des signaux bruts

Descriptive	Définition
mean()	valeur moyenne
std()	ecart-type
mad()	ecart absolu médian
max()	valeur maximale
min()	valeur minimale
sma()	zone d'amplitude du signal
energy()	valeur énergétique
iqr()	intervalle interquartile
entropy()	valeur de l'entropie du signal
arcoeff()	coefficient d'autorégression
correlation()	coefficient de corrélation
maxinds()	indice de la composante de fréquence de plus grande magnitude
meanfreq()	moyenne pondérée de la composante fréquence
skewness()	skewness du signal dans le domaine de la fréquence
kurtosis()	kurtosis du signal dans le domaine de la fréquence
bandsenergy()	energie de la fréquence dans la FFT de chaque fenêtre
angle()	angle entre les vecteurs

Remarque : L'angle est calculé uniquement pour les vecteurs gravityMean, tBodyAccMean, tBodyAccJerkMean, tBodyGyroMean, tBodyGyroJerkMean.

Le Tableau 2.4 donne une description détaillée du domaine des différentes caractéristiques extraites des différents types de signaux conduisant à un vecteur caractéristique de 561 éléments.

TABLE 2.4 – Description des 561 caractéristiques

Domaine	Types de signaux	Nombre de cara
Temporel (265 caractéristiques)	tBodyAcc-XYZ	40
	tGravityAcc-XYZ	40
	tBodyAccJerk-XYZ	40
	tBodyGyro-XYZ	40
	tBodyGyroJerk-XYZ	40
	tBodyAccMag	13
	tGravityAccMag	13
	tBodyAccJerkMag	13
	tBodyGyroMag	13
	tBodyGyroJerkMag	13
Temporel (7 caractéristiques)	angle	7
Fréquentiel(289 caractéristiques)	fBodyAcc-XYZ	79
	fBodyAccJerk-XYZ	79
	fBodyGyro-XYZ	79
	fBodyAccMag	13
	fBodyAccJerkMag	13
	fBodyGyroMag	13
	fBodyGyroJerkMag	13

2.5.3 Protocole d'évaluation

2.5.3.1 Split Dataset

La première étape du protocole d'évaluation consiste à créer des ensembles de données d'apprentissage et de test. Chaque classifieur sera entraîné sur l'ensemble de données d'apprentissage puis évalué par rapport à l'ensemble de données de test. La répartition des données utilisée dans ce travail est de 70% (7352) et 30% (2947) pour les ensembles d'apprentissage et de test respectivement.

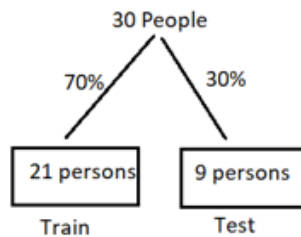


FIGURE 2.9 – Répartition de la base de données UCI-HAR

2.5.3.2 Tableau de confusion

Dans un problème à deux classes, le modèle de classification prédit l'une des deux classes qui sont généralement désignées sous le nom de classes positive et négative. Étant donné un classificateur et un exemple à classifier, quatre situations sont envisageables. Si l'exemple est positif et qu'il est classé comme positif, il compte comme un vrai positif TP ; s'il est classé négatif, il compte comme un faux négatif FN . Si l'exemple est négatif et qu'il est classé comme négatif, il compte comme un vrai négatif TN ; s'il est classé positif, il compte comme un faux positif FP . Les sorties produites par le classificateur peuvent être représentées par une matrice de confusion de 2×2 . La Table 2.5 montre la matrice de confusion pour un problème à deux classes où les lignes indiquent la classe réelle et les colonnes indiquent la classe prédite.

TABLE 2.5 – Tableau de confusion d'un problème à deux classes.

		Classe prédite	
		Positive	Negative
Classe réelle	Positive	TP	FN
	Negative	FP	TN

2.5.3.3 Taux de classification

Le taux de bonne classification est la mesure de performance la plus simple et qui est à la base de tout autre critère [Baldi et al., 2000]. Le taux de classification calcule le pourcentage d'échantillons correctement classés par rapport au nombre total d'échantillons. À partir des valeurs de le tableau de confusion (voir le Tableau 2.5), le taux de classification est donné par l'équation suivante :

$$TC = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.3)$$

2.5.4 Environnement de travail

Il est nécessaire de mettre en place l'environnement de travail du projet. Pour ce projet, nous proposons Google Colab.

Google Colab est un service en cloud gratuit qui prend désormais en charge le GPU gratuit, qui permet d'évaluer les compétences en langage de programmation

Python. Il s'agit d'un environnement de carnet de notes Jupyter composé de cellules pouvant contenir du code, du texte, des figures et ne nécessitant aucune installation ou configuration.[2.10](#)

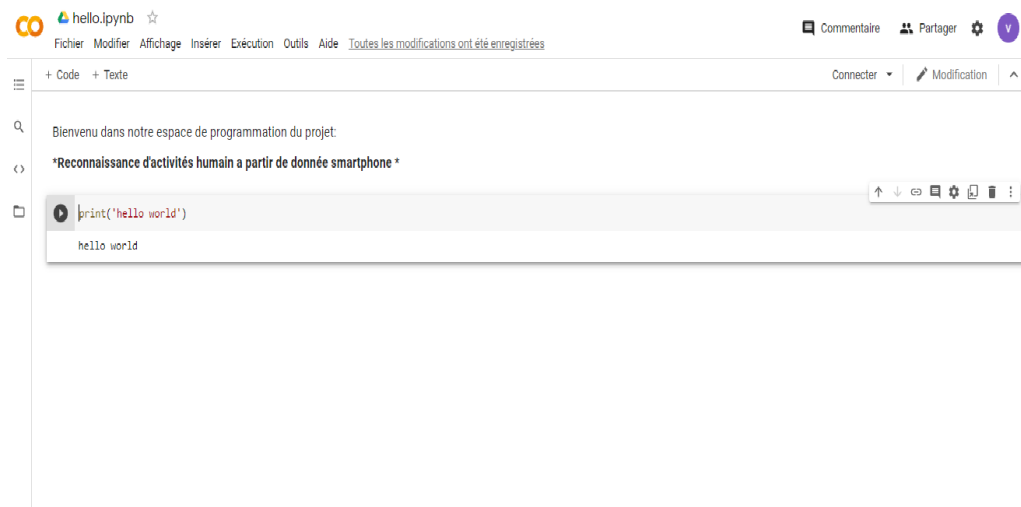


FIGURE 2.10 – Colaboratory (Exemple)

Python est un langage de programmation interprété, orienté objet, de haut niveau et à sémantique dynamique. Ses structures de données intégrées de haut niveau, combinées au typage dynamique et à la liaison dynamique, le rendent très attrayant pour le développement rapide d'applications, ainsi que pour une utilisation en tant que langage de script ou langage de liaison pour connecter des composants existants. Les bibliothèques Python populaires que nous avons utilisées sont :

- **Numpy** : est une bibliothèque permettant d'effectuer des calculs numériques avec Python. Elle introduit une gestion plus facile des tableaux de nombres, des fonctions sophistiquées (distribution).
- **Matplotlib** : est une bibliothèque de traçage pour le langage de programmation Python. Elle fournit une API orientée objet pour intégrer des graphiques dans des applications utilisant des kits d'outils d'interface utilisateur graphique.
- **Scikit-learn** : est une bibliothèque libre Python destinée à l'apprentissage automatique.

2.5.5 Évaluation et discussions

L'évaluation est effectuée en prévoyant les classes des ensembles de données de test en utilisant les modèles générés de l'apprentissage des algorithmes K-NN et SVM.

La Table 2.6 indique les taux de reconnaissance en utilisant les classifieurs K-NN et SVM. Le SVM a produit le meilleur résultat, soit un taux de classification de 96.4%. Le K-NN a produit des résultats avec des précisions inférieures à ceux

TABLE 2.6 – Taux de classification des algorithmes KNN et SVM

Algorithme ML	Taux de classification (%)
SVM	96.4
K-NN(K=5)	90.2
K-NN(K=3)	89.07

obtenus avec le SVM. Le taux de classification est de 89.07 % lorsque K est fixé à 3 et atteint 90.2 % lorsque K est fixé à 5.

Une fois que les résultats des modèles soient générés, les matrices de confusion sont tracées avec les données d'entrée comme valeurs prédites et la référence comme étiquettes de caractéristiques cibles originales.

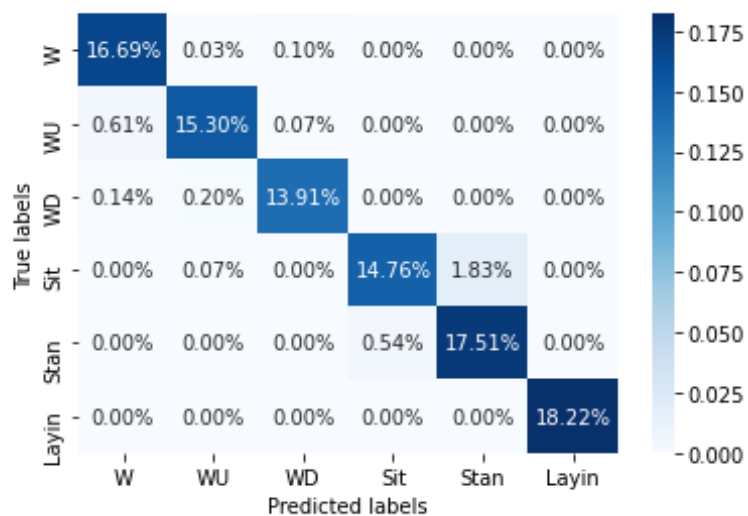


FIGURE 2.11 – Tableau de confusion du SVM

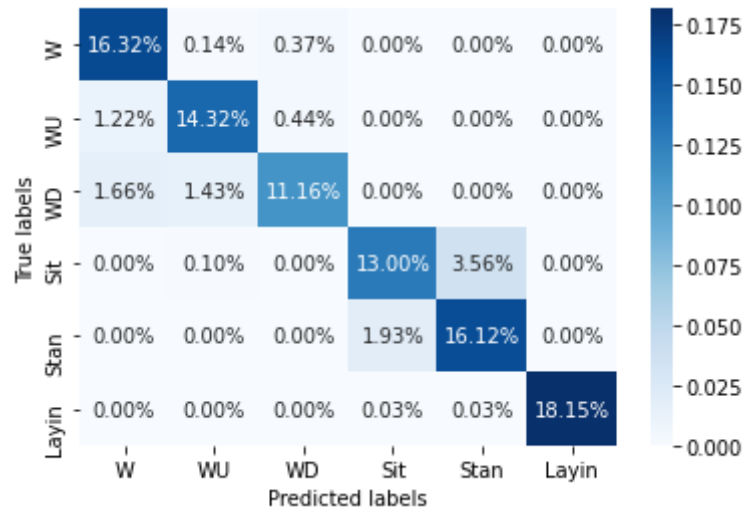


FIGURE 2.12 – Tableau de confusion du K-NN lorsque $K=5$

Les Figures 2.11 et 2.12 présentent les tableaux de confusion pour le SVM et le K-NN lorsque de $K = 5$ respectivement. On remarque que l'algorithme SVM a très peu d'erreurs de classification comparé avec le K-NN. A titre d'exemple, Le SVM classe l'activité monter comme descendre avec un taux de 0.07%, alors que le K-NN enregistre un taux de 0.44% sous les mêmes hypothèses.

2.6 Conclusion

Dans ce chapitre, nous avons décrit et évalué l'approche utilisant les algorithmes KNN et SVM pour la classification de l'activité humaine. Le meilleur taux de classification dans notre expérience était atteint par le SVM en utilisant un vecteur caractéristique contenant 561 éléments.

Chapitre 3

Classification de l'Activité par le Deep Learning

Sommaire

3.1	Introduction	42
3.2	Deep Learning (DL)	42
3.2.1	Présentation de quelques types de DL	43
3.3	Les réseaux de neurones artificiels (ANN)	43
3.3.1	Neurone artificiel	44
3.3.2	Fonction d'activation	45
3.4	Convolutional Neural Network	46
3.4.1	Couche de convolution	46
3.4.2	Couche de Pooling	47
3.4.3	Dropout	48
3.4.4	Flatten	48
3.4.5	Couche entièrement connectée	48
3.5	La classification de l'activité humaine par le CNN sur Python	48
3.5.1	Apprentissage d'un réseau CNN	49
3.5.2	Évaluation d'un réseau CNN	51
3.6	Conclusion	51

3.1 Introduction

L'intelligence artificielle est une discipline scientifique recherchant des méthodes de solution de problèmes à forte complexité logique ou algorithmique. L'apprentissage profond (en anglais deep learning, deep structured learning) est un ensemble de méthodes d'apprentissage automatique tentant de modéliser avec un haut niveau d'abstraction des données grâce à des architectures articulées de différentes transformations non linéaires. Dans ce chapitre nous allons présenter tout d'abord les notions en relation avec l'apprentissage profond.

3.2 Deep Learning (DL)

L'apprentissage profond(DL) est l'une des principales technologies d'apprentissage automatique et d'intelligence artificielle. Il s'est imposé et se développe rapidement comme l'un des domaines scientifiques les plus importants. Il se caractérise par un réseau de neurones, qui contient de nombreuses couches cachées composées de milliers de neurones qui effectuent chacun de petites opérations simples. Les résultats d'une première couche de neurones sont utilisés comme entrée pour les calculs d'une deuxième couche et ainsi de suite. Une illustration d'un réseau profond est donnée dans la Figure 3.1.

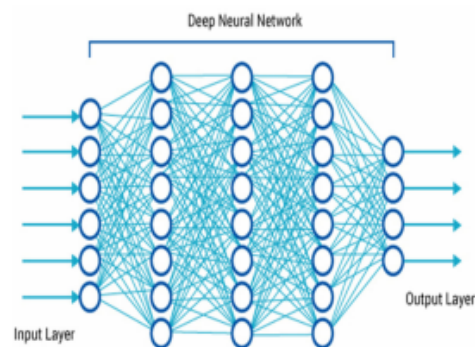


FIGURE 3.1 – Deep neural network

3.2.1 Présentation de quelques types de DL

Il existe beaucoup de types de DL, chaque type étant développé pour un objectif particulier

3.2.1.1 Fully Connected Neural Network (FCN)

Est une structure composée de plusieurs couches cachées de neurones, où tous les nœuds contenus sont connectés à tous les nœuds de la couche suivante. Les FCN sont couramment utilisés dans les réseaux neuronaux convolutifs et les réseaux neuronaux récurrents.

3.2.1.2 Recurrent Neural Network (RNN)

Un réseau neuronal récurrent est un type de réseau neuronal qui contient des boucles, permettant de stocker des informations au sein du réseau. En bref, les réseaux neuronaux récurrents utilisent le raisonnement des expériences précédentes pour informer les événements à venir.

3.2.1.3 Convolutional neural network (CNN)

Un CNN a plusieurs couches à travers lesquelles les données sont filtrées en catégories. Les CNN se sont avérés très efficaces dans des domaines tels que la reconnaissance d'images, le traitement du langage textuel et la classification.

3.2.1.4 Long Short-Term Memory (LSTM)

Les réseaux LSTM sont un type de réseau neuronal récurrent capable d'apprendre la dépendance de l'ordre dans les problèmes de prédiction de séquence. Ce comportement est nécessaire dans des domaines complexes comme la traduction automatique, la reconnaissance vocale, etc. Les réseaux LSTM constituent un domaine complexe de l'apprentissage profond.

3.3 Les réseaux de neurones artificiels (ANN)

Les réseaux de neurones artificiels (ANN) sont généralement inspirés des réseaux de neurones biologiques qui imitent la fonctionnalité du cerveau. Ces systèmes apprennent généralement en considérant des exemples au lieu de définir des règles

spécifiques pour certaines situations ou certains cas. Un ANN est un réseau de nœuds, appelés neurones artificiels, qui sont connectés à d'autres neurones par un lien appelé synapse. Chaque neurone reçoit l'entrée, la traite et transmet la sortie au neurone suivant. Dans son état le plus élémentaire, le réseau ANN se compose de trois couches : la couche d'entrée, la couche cachée et la couche de sortie.

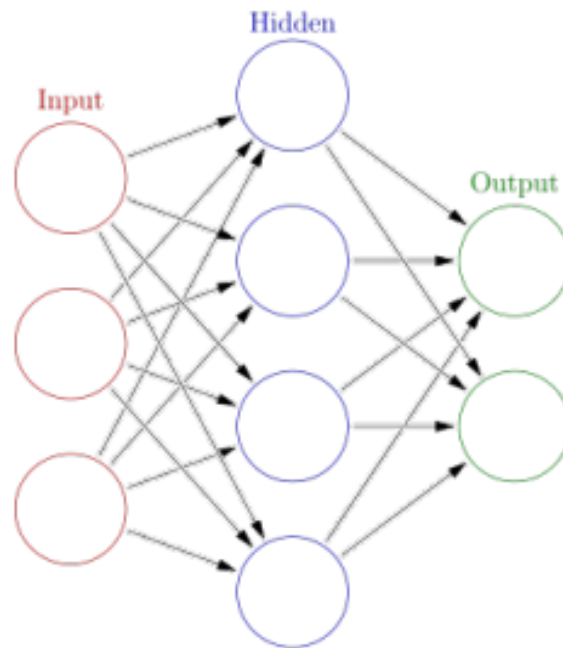


FIGURE 3.2 – Un réseau neuronal artificiel avec trois couches de base

3.3.1 Neurone artificiel

Un neurone artificiel est l'unité la plus basique du ANN. Il prend des entrées et produit une sortie. En général, les entrées sont multipliées par certains poids afin de spécifier quelles entrées sont les plus importantes. Plus la valeur du poids est élevée, plus l'entrée est importante. Dans la Figure 3.3, les entrées sont représentées par a , b et c , et les poids par w_1 , w_2 et w_3 . Les produits sont additionnés et transmis à la fonction d'activation. Ainsi, si la valeur additionnée est supérieure à la valeur seuil de la fonction d'activation, la sortie est produite ou, en d'autres termes, le neurone a été activé. Dans l'autre cas, aucune sortie n'est produite et le neurone n'est pas activé.

Les neurones artificiels ajustent les poids au fur et à mesure de l'apprentissage. Pon-

dérations est appelé apprentissage. ANN considère de nombreux exemples différents et trouve la meilleure combinaison possible de poids pour obtenir les résultats les plus précis.

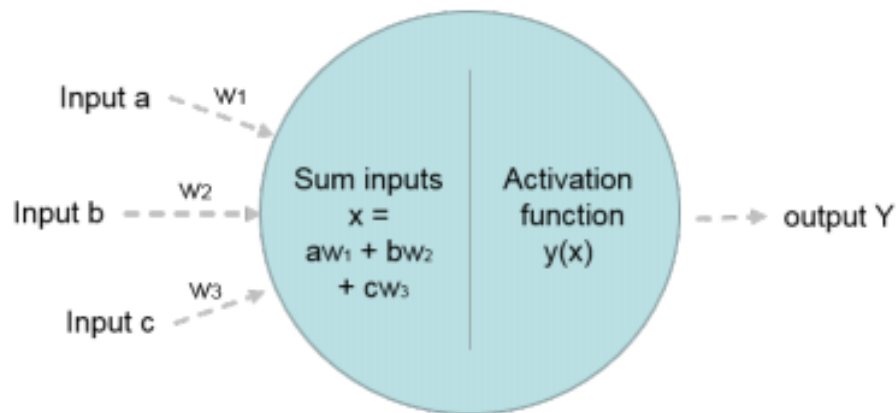


FIGURE 3.3 – Un seul neurone artificiel

3.3.2 Fonction d'activation

La fonction d'activation est un élément crucial de l'apprentissage profond. Elle détermine si un neurone se déclenche ou non. Dans ce qui suit, nous allons décrire les différentes fonctions d'activation.

Fonction sigmoïde (Sigmoid function) : Le but de cette fonction est de réduire la valeur d'entrée entre 0 et 1. Elle est définie comme suit :

$$F(x) = \frac{1}{1 + e^{-x}} \quad (3.1)$$

Fonction Tanh : Également appelée "tangente hyperbolique", elle produit un résultat compris entre -1 et 1. Elle est définie comme suit :

$$F(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (3.2)$$

Unité linéaire rectifiée (ReLU) : ReLU est la fonction la plus couramment utilisée en apprentissage profond. Elle est définie comme suit :

$$F(x) = \max(0, x) \quad (3.3)$$

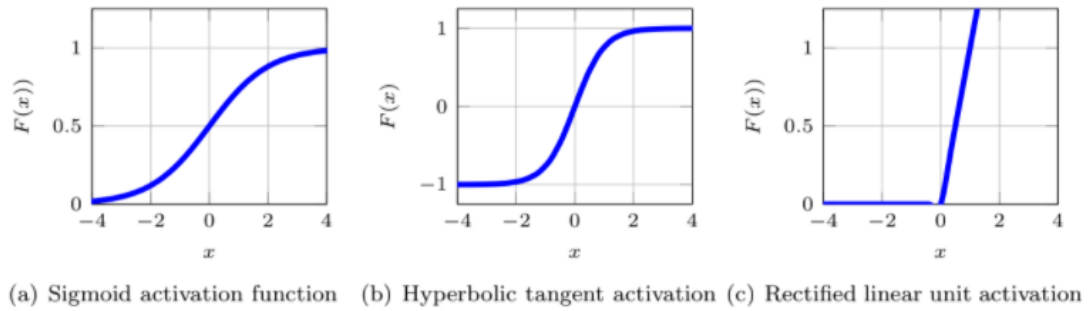


FIGURE 3.4 – Fonctions d’activation

Une représentation graphique des fonctions d’activation su-cités est donnée dans la Figure 3.4.

3.4 Convolutional Neural Network

Le réseau neuronal convolutif (CNN) est une classe de réseaux neuronaux profonds, qui utilise des perceptrons multicouches. Il se compose d’une couche d’entrée, d’une couche de sortie et de plusieurs couches cachées. Les couches cachées peuvent être des couches convolutionnelles, des couches de pooling, des couches flatenn ou des couches entièrement connectées. La Figure 3.5 présente un exemple de CNN.

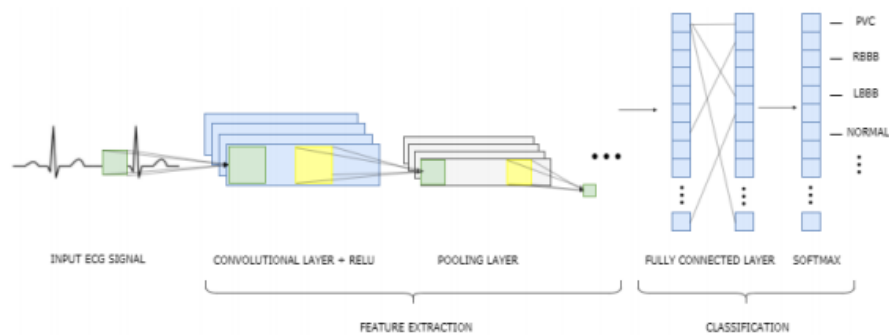


FIGURE 3.5 – Exemple du réseau de neurones convolutifs [Ahmed, 2018]

Dans ce qui suit, les différentes couches cachées d’un réseau CNN sont présentées.

3.4.1 Couche de convolution

Dans un CNN, la première couche est toujours la couche convolutive. Cette couche applique une opération convolutive à l’entrée et transmet la sortie à la couche

suivante. On utilise un filtre (ou parfois appelé noyau) qui parcourt toutes les zones de l'entrée et en extrait les caractéristiques. Une couche convolutionnelle unidimensionnelle utilise une opération de corrélation croisée unidimensionnelle dans laquelle la fenêtre de convolution part du côté le plus à gauche du signal d'entrée et parcourt le signal d'entrée de gauche à droite successivement. Lorsque la fenêtre de convolution glisse jusqu'à une certaine position, les sous-réseaux d'entrée dans la fenêtre et le réseau de noyau sont multipliés et additionnés élément par élément pour obtenir le résultat à l'emplacement correspondant dans le réseau de sortie comme le montre la Figure 3.6

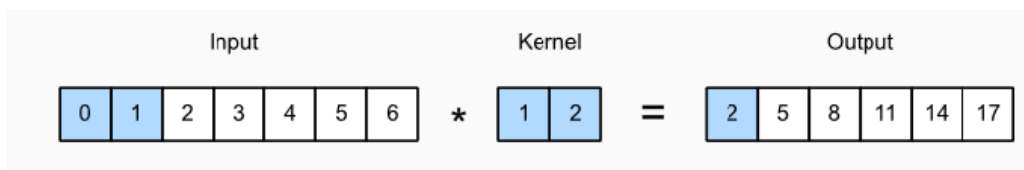


FIGURE 3.6 – Opération de corrélation croisée unidimensionnelle. Les parties en bleu sont le premier élément d'entrée ainsi que l'élément de sortie du noyau utilisé dans lequel : $0 \times 1 + 1 \times 2 = 2$

3.4.2 Couche de Pooling

La couche de pooling est utilisée pour sous-échantillonner la sortie de la couche convolutive. Il existe plusieurs types de pooling tels que, le mean pooling, le max pooling. La couche max pooling prend essentiellement une fenêtre de taille $l \times 1$ et sélectionne ensuite le plus grand des l nombres dans cette fenêtre. Le même processus est appliqué aux différentes sous-régions. Un exemple de fonctionnement de la couche max pooling est illustré à la Figure 3.7

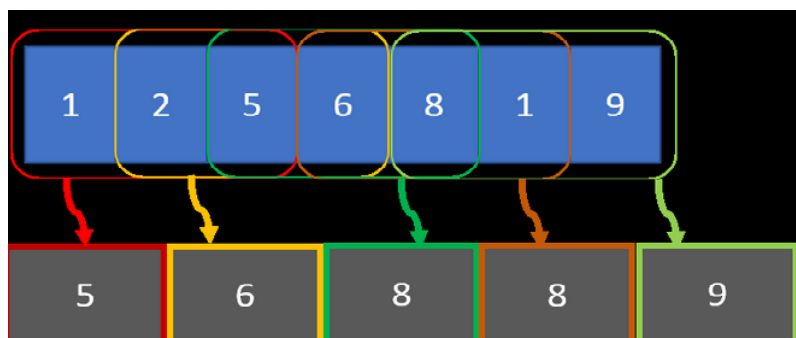


FIGURE 3.7 – 1D MaxPooling avec une taille de fenêtre $l = 3$

3.4.3 Dropout

Le Dropout est une technique utilisée pour éviter qu'un modèle ne soit surajusté. Le Dropout fonctionne en fixant de manière aléatoire les bords sortants des unités cachées (neurones qui composent les couches cachées) à 0 à chaque mise à jour de la phase de l'information.

3.4.4 Flatten

Le rôle de la couche Flatten consiste à convertir les données en un tableau unidimensionnel pour les introduire dans la couche suivante. La sortie des couches convolutionnelles est mise à plat pour créer un seul long vecteur de caractéristiques, qui sera connecté à la couche appelée couche entièrement connectée.

3.4.5 Couche entièrement connectée

Dans une couche entièrement connectée, les neurones d'une couche sont connectés à tous les neurones de la deuxième couche. Cette couche est également présente dans le réseau neuronal classique. La fonction softmax est appliquée à la sortie de la deuxième couche. La sortie de la fonction softmax permet de calculer les probabilités de chaque étiquette.

3.5 La classification de l'activité humaine par le CNN sur Python

Keras est une bibliothèque open source de réseaux de neurones artificiels écrite en Python. Il s'agit d'une bibliothèque très puissante et facile à utiliser pour développer des réseaux de neurones. Elle a la capacité de fonctionner au-dessus de TensorFlow, Microsoft Cognitive Toolkit (CNTK) ou Theano, et peut fonctionner à la fois sur le CPU et le GPU. Avant l'introduction de Keras, le développement d'un réseau sur TensorFlow ou Theano prenait beaucoup de temps. Keras prend en charge les réseaux neuronaux convolutifs et les réseaux neuronaux récurrents, ainsi qu'une combinaison des deux.

3.5.1 Apprentissage d'un réseau CNN

En pratique, le modèle est d'abord défini comme séquentiel à l'aide de la fonction `Sequential()` de Keras, puis les différentes couches sont ajoutées au modèle à l'aide de la fonction `add()`. Enfin, le modèle est compilé en appelant la fonction `compile()`, qui a comme entrées :

- **optimiser** utilisé pour optimiser le réseau neuronal, tel que Adagrad et Adam.
- **loss** qui fait référence à la valeur que le modèle doit minimiser, comme l'entropie croisée et l'erreur quadratique moyenne.
- **Metrics** qui évaluent la précision des problèmes de classification.

Le modèle est entraîné à l'aide de la méthode `fit()`. Cette méthode permet au modèle d'itérer sur les données et de trouver le réseau neuronal le plus optimal pour ces données.

Le modèle proposé dans cette étude est présenté dans la Figure 3.8. Il se compose de 3 couches alternant le convolution, le max pooling et le dropout. Une couche flatten est ajoutée pour aplatir les sorties de la dernière couche. Une couche entièrement contenant 256 neurones connectée reliée à une autre couche de même type contenant 128 neurones. La couche de sortie contient 6 neurones et la fonction d'activation softmax, définie dans l'équation 3.4, est utilisée pour différencier les différentes activités humaines.

$$\delta(z)_j = \frac{e^{z_j}}{\sum_{i=1}^N e^{z_i}} \quad (3.4)$$

Dans ce modèle, la forme de l'entrée est donnée par $(n_timesteps, n_features)$ où $n_timesteps = 128$ et $n_features = 9$. Après avoir défini l'architecture du modèle, l'étape d'apprentissage est effectuée avec des hyper paramètres définis comme `Batch_size = 64` et `Epochs = 100`.

L'utilisation de d'Adam donne de bonnes performances au début de la phase d'apprentissage. Les métriques semblent converger plus tôt, se stabilisant à 98% de précision et à moins de 0.05 de perte. Par la suite, le modèle est resté stable comme le montre l'historique Colab illustrées Dans la Figure 3.9.

Les graphiques de la précision et de la perte en fonction de l'epoch sont illustrés dans les figures Figure 3.10.

```

model = Sequential()
model.add(Conv1D(filters=128, kernel_size=5, activation='relu', input_shape=(n_timesteps,n_features)))
model.add(MaxPooling1D(pool_size=2))
model.add(Dropout(0.3))
model.add(Conv1D(filters=64, kernel_size=3, activation='relu'))
model.add(MaxPooling1D(pool_size=2))
model.add(Dropout(0.3))
model.add(Conv1D(filters=128, kernel_size=3, activation='relu'))
model.add(MaxPooling1D(pool_size=2))
model.add(Dropout(0.2))
model.add(MaxPooling1D(pool_size=2))
model.add(Flatten())
model.add(Dense(256, activation='relu'))
model.add(Dense(128, activation='relu'))
model.add(Dense(n_outputs, activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

```

FIGURE 3.8 – Exemple des couches utilisé

```

Epoch 11/100
145/145 [=====] - 9s 64ms/step - loss: 0.0784 - accuracy: 0.9671
Epoch 12/100
145/145 [=====] - 9s 65ms/step - loss: 0.0793 - accuracy: 0.9670
Epoch 13/100
145/145 [=====] - 9s 64ms/step - loss: 0.0797 - accuracy: 0.9661
Epoch 14/100
145/145 [=====] - 9s 64ms/step - loss: 0.0782 - accuracy: 0.9708
Epoch 15/100
145/145 [=====] - 9s 65ms/step - loss: 0.0748 - accuracy: 0.9687
Epoch 16/100
145/145 [=====] - 9s 63ms/step - loss: 0.0641 - accuracy: 0.9764
Epoch 17/100
145/145 [=====] - 9s 62ms/step - loss: 0.0616 - accuracy: 0.9737
Epoch 18/100
145/145 [=====] - 9s 62ms/step - loss: 0.0561 - accuracy: 0.9761
Epoch 19/100
145/145 [=====] - 9s 61ms/step - loss: 0.0547 - accuracy: 0.9785
Epoch 20/100
145/145 [=====] - 9s 62ms/step - loss: 0.0551 - accuracy: 0.9765
Epoch 21/100
145/145 [=====] - 9s 62ms/step - loss: 0.0530 - accuracy: 0.9793
Epoch 22/100
145/145 [=====] - 9s 62ms/step - loss: 0.0484 - accuracy: 0.9804
Epoch 23/100
145/145 [=====] - 9s 62ms/step - loss: 0.0500 - accuracy: 0.9809
Epoch 24/100
145/145 [=====] - 9s 61ms/step - loss: 0.0465 - accuracy: 0.9803

```

FIGURE 3.9 – L'historique de Colab durant la phase d'apprentissage

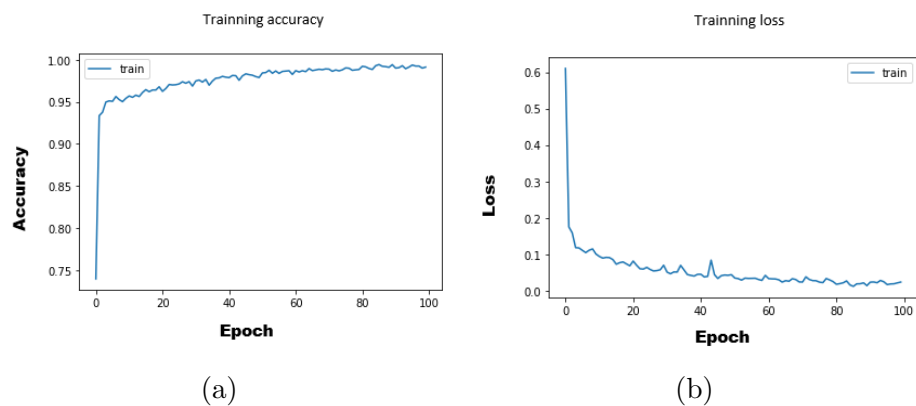


FIGURE 3.10 – (a) Résultats de précision de l'apprentissage (b) Fonction de perte de l'apprentissage du modèle CNN pour 100 itérations

3.5.2 Évaluation d'un réseau CNN

La phase d'évaluation est effectuée à travers le modèle appris précédemment. L'architecture du réseau CNN qui utilise les combinaisons des différents types de sources de signaux au niveau de la première couche de convolution donne un taux de classification atteignant 94.26% sur des activités non-apprises.

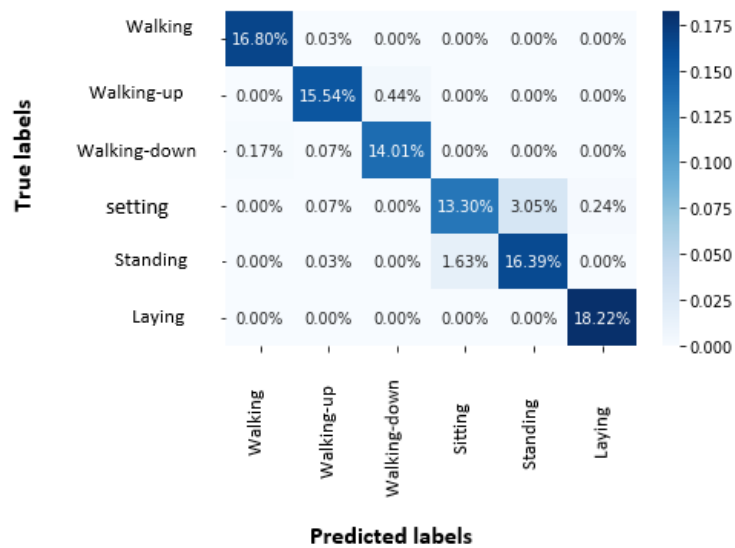


FIGURE 3.11 – Tableau de confusion de l'architecture CNN

La Figure 3.11 présente le tableau de confusion pour chaque classe (activité). Par exemple, le modèle CNN proposé a classé les 6 activités de test comme suit. Il classe l'activité se mettre debout comme s'asseoir avec un taux de 1.63% et inversement, il classe l'activité s'asseoir comme se mettre debout avec un taux de 3.05%. Le modèle proposé enregistre le meilleur taux de classification pour l'activité s'allonger.

3.6 Conclusion

Dans ce chapitre, on a présenté les notions importantes qui sont en relation avec l'apprentissage profond (définition, types...etc). Ainsi qu'une vision générale sur l'apprentissage profond, tout en donnant en détail la méthode choisie dans ce travail. L'architecture CNN proposée a donné des résultats satisfaisants

Chapitre 4

Classification de l'Activité sur La Raspberry Pi4

Sommaire

4.1	Introduction	53
4.2	Réorganisation des données	53
4.3	Normalisation des données	53
4.4	Évaluation	54
4.5	Implémentation sur Raspberry Pi4	55
4.5.1	Présentation du Raspberry Pi4	55
4.5.2	Architecture de la Raspberry Pi4	56
4.5.3	Configuration de la Raspberry Pi4	57
4.6	L'application Sensorstream IMU+GPS	59
4.7	La classification de l'activité humaine dans la Raspberry	62
4.8	Conclusion	64

4.1 Introduction

Le but de notre travail est la reconnaissance d'activité humaine à partir de données smartphone. Dans le chapitre précédent, on a présenté le DL et ses principales architectures. Le modèle proposé précédemment, entraîné sur 70% de la base de données UCI-HAR a donné un taux de classification 94.26%. Une augmentation de la base d'apprentissage entraînerait sûrement une augmentation du taux de classification.

4.2 Réorganisation des données

L'ensemble précédent contenant 10299 données a été divisé en deux sous ensembles, ou 7352 données ont été sélectionnées pour générer le sous ensemble d'apprentissage et 2947 données ont été sélectionnées pour générer le sous ensemble de test. Dans cette partie, nous allons :

- tout d'abord, fusionner les sous ensembles sus-cités en un unique ensemble
- partitionner aléatoirement à nouveau les données en deux sous ensembles. Lors de cette partition 90% soit 9269 données sont utilisées pour l'apprentissage et 10% soit 1030 données pour le test. L'ajout d'un plus grand nombre d'exemples à l'ensemble d'apprentissage permet de garantir la précision des résultats est par conséquent le succès du modèle sur des données non apprises .

4.3 Normalisation des données

La normalisation (standardisation) est le concept et l'étape qui consiste à placer différentes variables sur la même échelle. Ce concept permet de comparer les scores entre différents types de variables. La standardisation peut également être appliquée quand les descripteurs ont des unités différentes ou ont été pris par des capteurs différents.

La Normalisation est le processus de transformer un descripteur en une autre qui répondra à la loi normale (Gaussian Distribution) $X \sim \mathcal{N}(\mu, \sigma)$ avec :

- $\mu = 0$ La moyenne de la loi de distribution.
- $\sigma = 1$ est l'Écart-type (Standard Deviation).

La formule de standardisation d'un descripteur x est la suivante :

$$x^* = \frac{x - \mu}{\sigma} \quad (4.1)$$

avec :

- x la valeur qu'on veut standardiser (input variable).
- μ la moyenne (mean) des observations pour cette feature.
- σ est l'écart-type (Standard Deviation) des observations pour ce descripteur.

4.4 Évaluation

Dans la section 3.5 du chapitre 3, nous avons proposé et évalué une architecture de deep learning, dont les données d'entrées représentant 70% de la base de données UCI-HAR, atteignant plus de 94% de bonne classification de l'activité humaine.

Il est bien établi que les performances des algorithmes de deep learning augmentent en accroissant la taille des ensembles d'apprentissage.

Le modèle CNN proposé dans la section 3.5.1 est entraîné par un ensemble contenant 9269 données normalisées suivant l'équation 4.1 et testé sur un ensemble incluant 1030 données normalisées. Un taux de classification de 99.22% est atteint. La nouvelle phase d'évaluation est effectuée à travers le modèle appris précédemment.

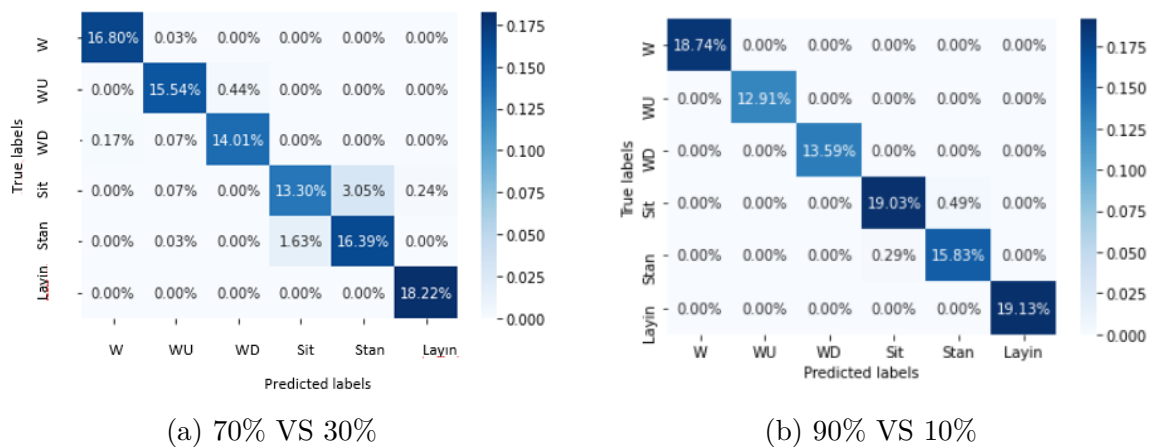


FIGURE 4.1 – Les matrices de confusion de l'architecture CNN des deux modèles entraînés

La Figure 4.1 représente les matrices de confusion pour chaque classe (activité) en utilisant le réseau CNN proposé. Elle donne une comparaison entre les deux modèles entraînés. A titre d'exemple, le modèle CNN précédent a classé l'activité s'asseoir comme se tenir debout avec une erreur de classification de 3.05% (voir la Figure 4.1a), tandis le nouveau modèle effectue la même tâche avec une erreur de classification de 0.49% (voir la Figure 4.1b).

4.5 Implémentation sur Raspberry Pi4

Depuis l'invention du transistor, les ordinateurs sont devenus de plus en plus petits après chaque génération, nous avons aujourd'hui ce que l'on appelle les nano-ordinateurs. Avec de hautes performances et une petite taille, ils peuvent être facilement mis en œuvre dans des dispositifs plus petits pour exécuter des tâches de calcul complexes. Dans ce projet, le nano-ordinateur Raspberry Pi4 est utilisé pour mettre en œuvre le système de classification de la reconnaissance d'activité humaine. Son architecture et sa configuration seront abordées plus en détail dans cette section.

4.5.1 Présentation du Raspberry Pi4

La Raspberry Pi est plus qu'une carte, elle est considérée comme un mini-ordinateur avec un système d'exploitation. Elle dispose de tous les ports de connexion nécessaires où l'utilisateur peut brancher des périphériques tels qu'une souris ou un clavier. La Raspberry Pi est capable d'effectuer les mêmes tâches qu'un ordinateur, comme surfer sur Internet, éditer des textes, regarder des vidéos haute définition ou programmer.

La Raspberry Pi a la capacité d'interagir avec le monde extérieur et d'exécuter plusieurs variantes du système d'exploitation libre (GNU/Linux, Raspbian Debian ...). Différents modèles Raspberry Pi existent. Dans ce qui suit, nous allons présenter une comparaison entre les cartes les plus utilisées dans le Tableau 4.1.

TABLE 4.1 – Comparaison entre différents modèles de Raspberry

Type de modele	Pi 3 Model B	Pi 3 Model B+	Pi 3 Model B
CPU type	ARM Cortex-A53	ARM Cortex-A53	ARM Cortex-A72
RAM Size	1GB LPDDR2	1GB LPDDR2	4GB LPDDR4
Integrated WI-FI	2.4GHz	5GHz	5GHz
Gigabit Ethernet	No	Over usb2	Yes
Bluetooth	4.1 BLE	4.2BLU	5BLE
USB	4*USB2	4*USB2	2*USB3,USB2
Video Decode	H.264	H.264	H.265
Video Encode	H.264	H.264	H.264

4.5.2 Architecture de la Raspberry Pi4

Dans ce projet, la Raspberry Pi4 est utilisée pour la mise en œuvre d'un système de reconnaissance de l'activité. Elle a l'architecture représentée dans la Figure 4.2 Elle est composée de :

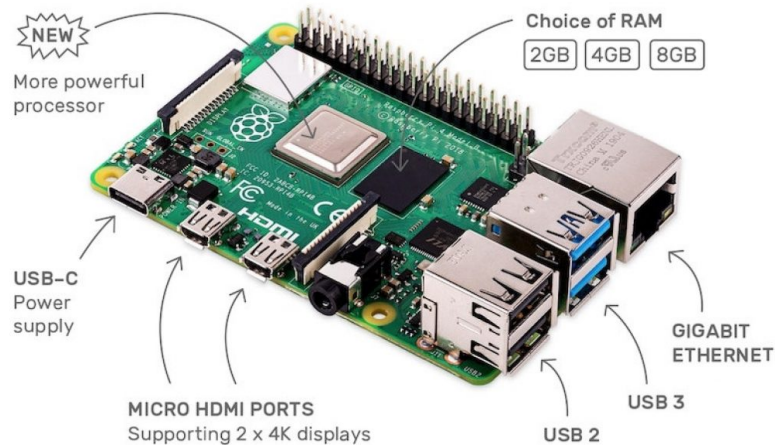


FIGURE 4.2 – Architecture du Raspberry Pi 4

- A Broadcom BCM2711 processor (4 Cortex-A72 cores clocked at 1.4 GHz)
- A VideoCore VI graphics processor
- 1, 2 or 4 GB of LPDDR4 RAM

- A slot for a microSD card
- Two USB 3.0 ports and two USB 2.0 ports
- One USB-C port
- One Gigabit Ethernet port
- Two microHDMI ports
- WiFi 5
- Bluetooth 5.0

Toutes les Raspberry Pi sont livrées par défaut sans système d'exploitation ni périphérique de stockage.

4.5.3 Configuration de la Raspberry Pi4

Pour démarrer une carte Raspberry Pi4, il est nécessaire d'avoir :

Une carte mémoire microSD : utilisée pour stocker le système d'exploitation et les programmes qui seront créés. Il est préférable qu'elle soit de haute capacité (au moins 16 GByte).

Un chargeur : alimentation 5V / 2A, avec un adaptateur micro USB.

Un moniteur et câble HDMI : pour afficher la sortie vidéo du Raspberry Pi4 en utilisant le port HDMI pour la connexion.

Un clavier et une souris : pour interagir avec le système d'exploitation de la carte Raspberry Pi, connectée par le port USB.

Un système d'exploitation : pour faire fonctionner le Raspberry.

Dans ce travail, la distribution recommandée "Raspbian" est utilisée. Il s'agit d'un système d'exploitation libre basé sur Debian et optimisé pour le matériel Raspberry Pi. Il est livré avec plus de 35 000 paquets, des logiciels précompilés qui facilitent le développement de logiciels

Pour configurer la Raspberry Pi, nous devons :

Télécharger l'image du système d'exploitation : La dernière version de l'image Raspbian peut être téléchargée par le lien suivant [¹].

Gravez l'image du système d'exploitation sur la carte SD :

La carte SD est insérée dans le lecteur de cartes du PC, puis l'image du système d'exploitation est gravée sur la carte SD à l'aide d'un des outils appropriés comme Win32Image ou Etcher. Lorsque cette opération est terminée, la carte SD est insérée dans la Raspberry Pi. Le clavier et la souris sont connectés aux ports USB du Raspberry Pi. Le câble HDMI est également connecté à la prise correspondante sur le Raspberry Pi et l'écran.

Configurer le système : Raspbian peut être configuré à l'aide de l'outil "Raspberry Pi software configuration" (voir la Figure 4.5) auquel on peut accéder à partir du terminal en tapant : `sudo raspi-config`.

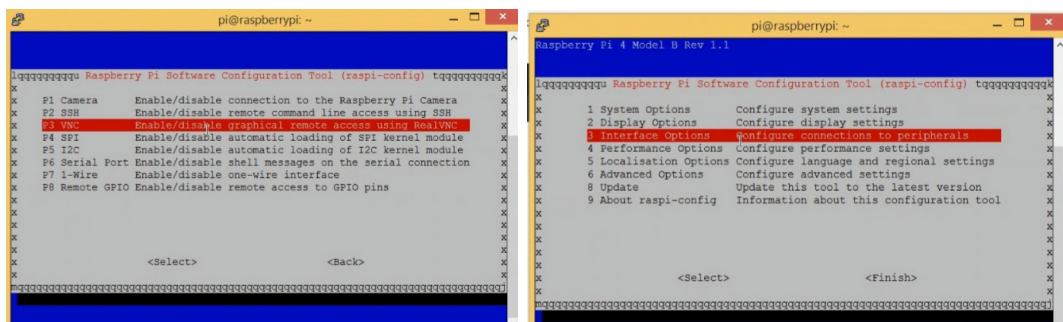


FIGURE 4.3 – Menu de configuration de Raspian

Il permet de configurer les paramètres suivants :

1. Changer le mot de passe de l'utilisateur
2. Modifier les paramètres du réseau.
3. Modifier les options d'emplacement.
4. Configurer les connexions des périphériques.
5. Configurer l'overclocking du Raspberry Pi Configurations avancées.
6. Mettre à jour Raspbian ou certains paquets.

Il est essentiel et nécessaire de maintenir le système à jour pour s'assurer que tous les pilotes fonctionnent dans de bonnes conditions. Cette opération peut être effectuée

1. <https://www.raspberrypi.org/software/>

très simplement en ouvrant le terminal et en tapant les commande sudo apt get update. La Raspberry Pi se connectera au référentiel hébergeant les paquets (fichiers, logiciels) de la distribution Raspbian installée, afin de récupérer les dernières versions comme le montre la Figure 4.4.

Une illustration du Desktop de la Raspberry Pi4 est montrée dans la Figure 4.5

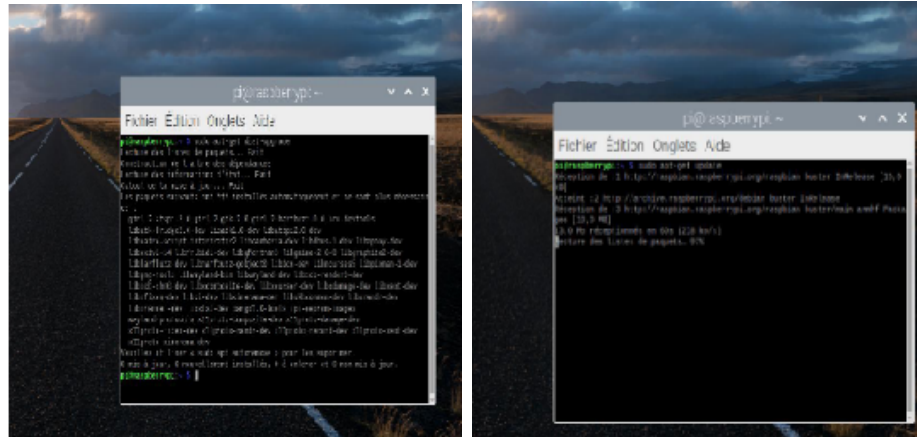


FIGURE 4.4 – Illustration de la mise à jour du système

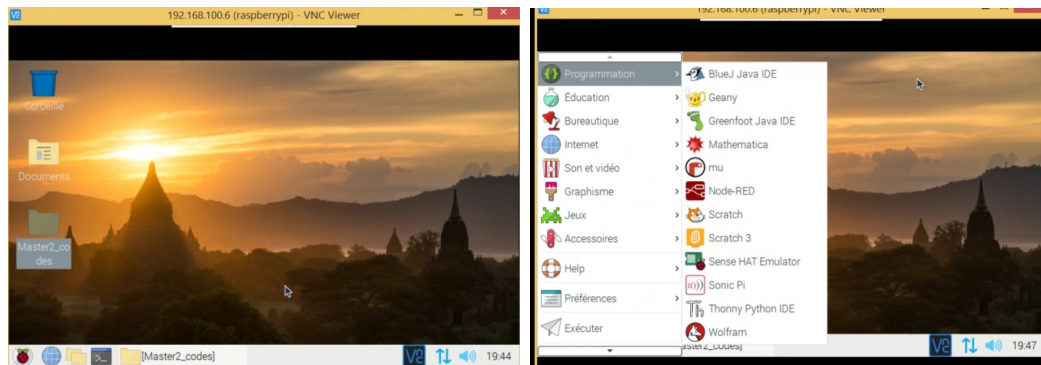


FIGURE 4.5 – Le bureau du Raspberry Pi4

4.6 L'application Sensorstream IMU+GPS

L'application Sensorstream IMU+GPS peut être utilisée pour collecter des données telles que les coordonnées GPS, les valeurs de accéléromètre, les valeurs du gyroscope etc. Cette application a été utilisée dans notre projet pour collecter des données du gyroscope et de l'accéléromètre de l'activité humaine comme suit :

Étape 1 : Télécharger et ouvrez l'application(disponible sur Google Play) (voir la Figure 4.6).

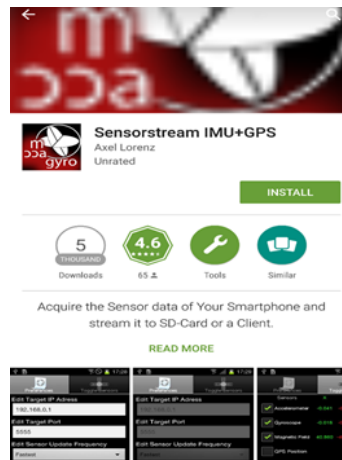


FIGURE 4.6 – SensorstreamIMU+GPS

Étape 2 : Choisir les données du capteur à enregistrer (voir la Figure 4.7).

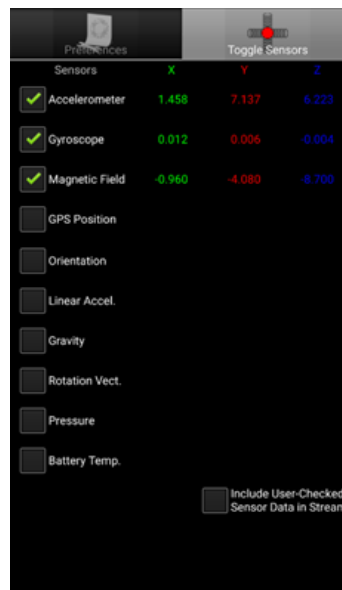


FIGURE 4.7 – Choix des capteurs sur SensorstreamIMU+GPS

Étape 3 : Configurer le flux de la carte SD (sélectionnez 'SD-Card Stream') (voir la Figure 4.8).

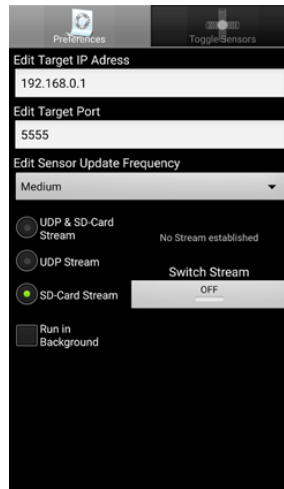


FIGURE 4.8 – Configuration de flux

Étape 4 : Commencer à enregistrer les données (pour activer l'enregistrement, appuyez sur le bouton intitulé "OFF" sous "Switch Stream) (voir la Figure 4.9).

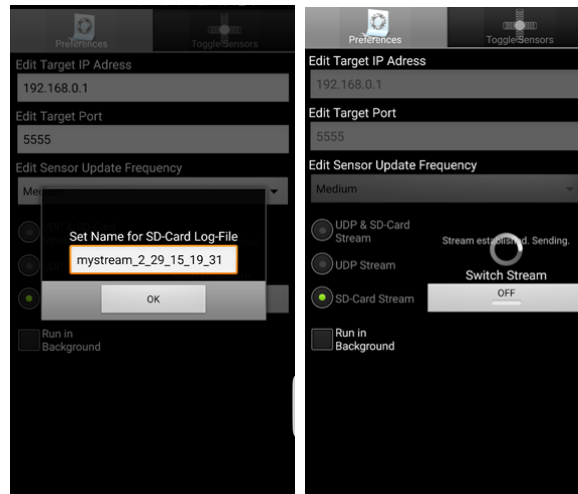


FIGURE 4.9 – Gestion du flux

Étape 5 : Désactiver le flux (pour arrêter l'enregistrement des données, cliquez sur "OFF" de sorte que le cercle au-dessus de "Switch Stream" disparaisse).

Avec l'application Sensorstream IMU+GPS, on peut envoyer les données en ligne avec un même réseaux (application, Raspberry) comme le montre la Figure 4.10.

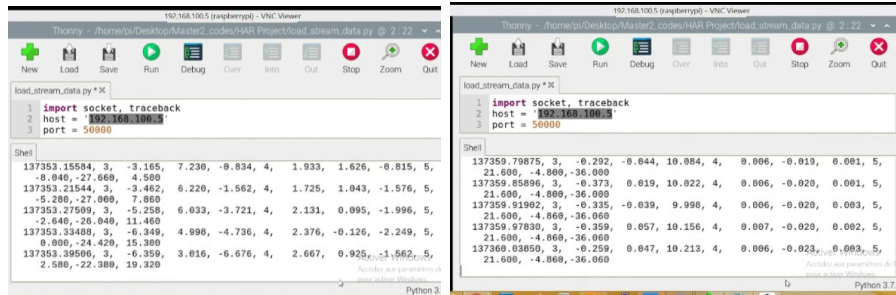


FIGURE 4.10 – Envoi des données en ligne dans la Raspberry Pi4

4.7 La classification de l'activité humaine dans la Raspberry

Le modèle appris et évalué et dont le taux de classification a atteint plus de 99% a été transféré du cloud Colab vers la Raspberry. De plus, le code Python effectuant la classification de l'activité humaine a été également transféré et enregistré dans un répertoire nommé HAR Project comme est montré dans la Figure 4.11.

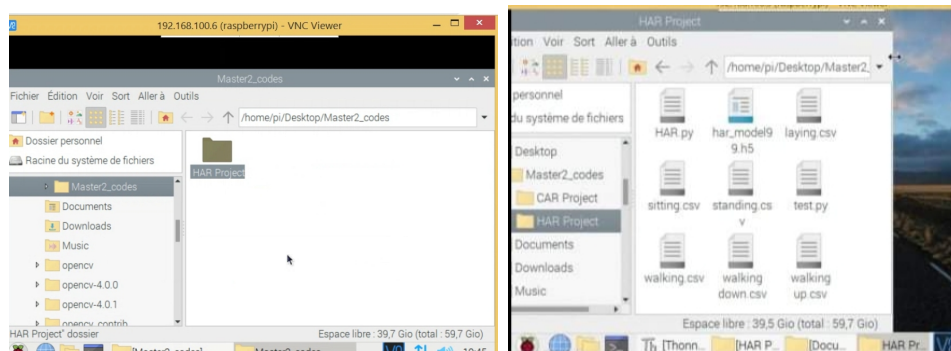


FIGURE 4.11 – Projet HAR dans la Raspberry Pi4

La Figure 4.12 représente quelques illustrations du code Python faisant appel au chargement des bibliothèques, lecture des données relatives à différentes activités humaines et leurs représentations graphiques dans la Raspberry Pi4.

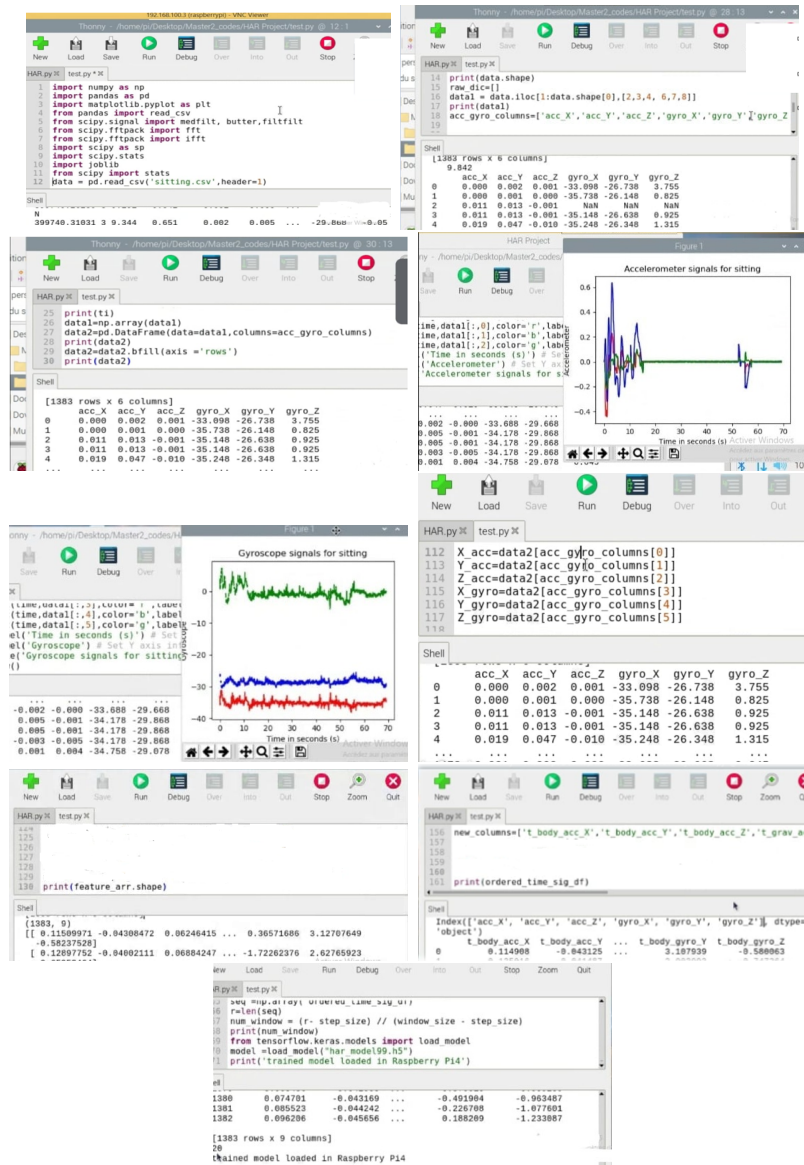


FIGURE 4.12 – Quelques résultats du programmation Python avec la Raspberry Pi4

La Figure 4.13 montre la classification de quelques activités telles que Walking, Standing et Sitting dans la Raspberry Pi4. Ces résultats montrent une correcte classification de l'activité humaine, prouvant la robustesse des approches basées sur le deep learning.

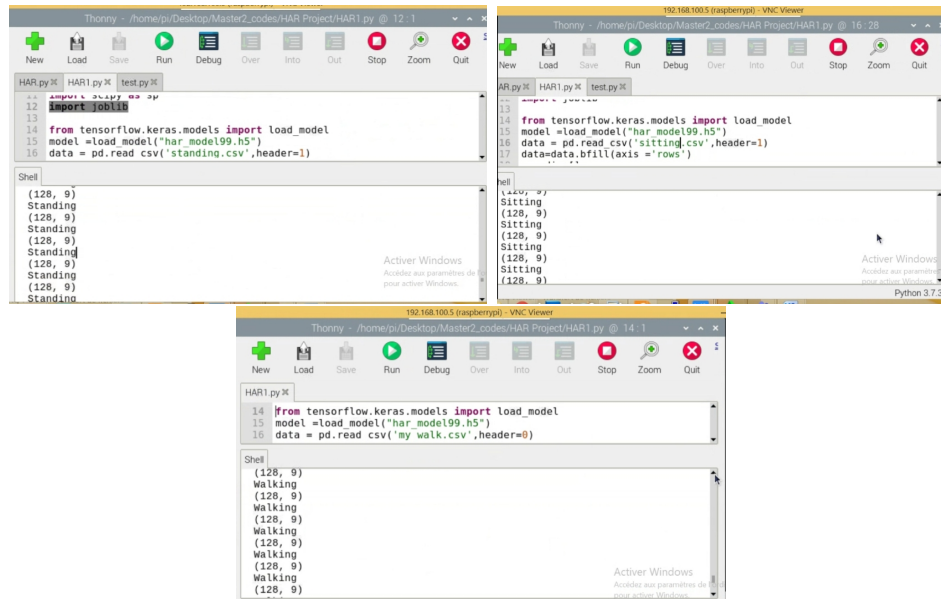


FIGURE 4.13 – La classification par la Raspberry de deux activité (Walking, Standing, Sitting)

4.8 Conclusion

Dans ce chapitre, nous avons évalué le modèle proposé dans le précédent chapitre sur la base UCI-HAR réorganisée dans le but d'atteindre de meilleur taux de classification. Par la suite, nous avons présenté la Raspberry Pi4 et ses avantages par rapport aux autres modèles de Raspberry. Nous avons ensuite défini l'architecture et l'installation de ce mini-ordinateur ainsi que la configuration du système d'installation Raspbian. Enfin, une implémentation du modèle sur la Raspberry Pi 4 et une évaluation de la classification des données provenant de smartphone est donnée.

Conclusion Générale

Le développement récent de la technologie consiste à essayer de rendre chaque dispositif de notre environnement intelligent dans la mesure du possible. L'un des objectifs des systèmes intelligents est de reconnaître les activités humaines quotidiennes à l'aide d'un téléphone intelligent afin d'encourager une vie saine et sûre pour les personnes préoccupées par leur santé. Dans notre travail, nous avons proposé une méthode pour automatiser les caractéristiques d'un ensemble de données de téléphones intelligents pour la tâche HAR en utilisant des algorithmes d'apprentissage automatique tels que le K-NN, le SVM et l'apprentissage profond qui est le réseau neuronal à convolution (CNN).

Cette méthode proposée construit une nouvelle architecture d'apprentissage profond pour les CNN pour étudier les données de séries temporelles. Elle utilise des opérations de convolution pour extraire des modèles à partir signaux provenant de capteurs. Les données que nous avons utilisées dans le système proposé ont été collectées sur la base du système HAR en utilisant deux capteurs trouvés dans un smartphone Samsung Galaxy S2 à partir du site Web UCI Machine Learning Repository. Le nano-ordinateur Raspberry Pi4 est utilisé pour mettre en œuvre le système de la reconnaissance de l'activité humaine.

Bibliographie

- [Ahmed, 2018] Ahmed, W. (2018). *Deep learning of cardiac-related conditions using a non-contact multisensor system*. PhD thesis, Master thesis, RWTH Aachen University.
- [Altman, 1992] Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3) :175–185.
- [Anguita et al., 2012] Anguita, D., Ghio, A., Oneto, L., Parra, X., and Reyes-Ortiz, J. L. (2012). Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine. In *International workshop on ambient assisted living*, pages 216–223. Springer.
- [Anguita et al., 2013] Anguita, D., Ghio, A., Oneto, L., Parra, X., and Reyes-Ortiz, J. L. (2013). Training computationally efficient smartphone-based human activity recognition models. In *International Conference on Artificial Neural Networks*, pages 426–433. Springer.
- [Anjum and Ilyas, 2013] Anjum, A. and Ilyas, M. U. (2013). Activity recognition using smartphone sensors. In *2013 IEEE 10th Consumer Communications and Networking Conference (CCNC)*, pages 914–919. IEEE.
- [Antos et al., 2014] Antos, S. A., Albert, M. V., and Kording, K. P. (2014). Hand, belt, pocket or bag : Practical activity tracking with mobile phones. *Journal of neuroscience methods*, 231 :22–30.
- [Atallah et al., 2010] Atallah, L., Lo, B., King, R., and Yang, G.-Z. (2010). Sensor placement for activity detection using wearable accelerometers. In *2010 International conference on body sensor networks*, pages 24–29. IEEE.
- [Avci et al., 2010] Avci, A., Bosch, S., Marin-Perianu, M., Marin-Perianu, R., and Havinga, P. (2010). Activity recognition using inertial sensing for healthcare,

- wellbeing and sports applications : A survey. In *23th International conference on architecture of computing systems 2010*, pages 1–10. VDE.
- [Baldi et al., 2000] Baldi, P., Brunak, S., Chauvin, Y., Andersen, C. A., and Nielsen, H. (2000). Assessing the accuracy of prediction algorithms for classification : an overview. *Bioinformatics*, 16(5) :412–424.
- [Barna et al., 2019] Barna, A., Masum, A. K. M., Hossain, M. E., Bahadur, E. H., and Alam, M. S. (2019). A study on human activity recognition using gyroscope, accelerometer, temperature and humidity data. In *2019 international conference on electrical, computer and communication engineering (ecce)*, pages 1–6. IEEE.
- [Bayat et al., 2014] Bayat, A., Pomplun, M., and Tran, D. A. (2014). A study on human activity recognition using accelerometer data from smartphones. *Procedia Computer Science*, 34 :450–457.
- [Bishop, 2006] Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.
- [Boser et al., 1992] Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152.
- [Breiman et al., 2001] Breiman, L. et al. (2001). Statistical modeling : The two cultures (with comments and a rejoinder by the author). *Statistical science*, 16(3) :199–231.
- [Brezmes et al., 2009] Brezmes, T., Gorricho, J.-L., and Cotrina, J. (2009). Activity recognition from accelerometer data on a mobile phone. In *International work-conference on artificial neural networks*, pages 796–799. Springer.
- [Bulbul et al., 2018] Bulbul, E., Cetin, A., and Dogru, I. A. (2018). Human activity recognition using smartphones. In *2018 2nd international symposium on multidisciplinary studies and innovative technologies (ismsit)*, pages 1–6. IEEE.
- [Catal et al., 2015] Catal, C., Tufekci, S., Pirmitt, E., and Kocabag, G. (2015). On the use of ensemble of classifiers for accelerometer-based activity recognition. *Applied Soft Computing*, 37 :1018–1022.
- [Chen et al., 2019] Chen, Z., Jiang, C., Xiang, S., Ding, J., Wu, M., and Li, X. (2019). Smartphone sensor-based human activity recognition using feature fusion and maximum full a posteriori. *IEEE Transactions on Instrumentation and Measurement*, 69(7) :3992–4001.

- [Coley et al., 2005] Coley, B., Najafi, B., Paraschiv-Ionescu, A., and Aminian, K. (2005). Stair climbing detection during daily physical activity using a miniature gyroscope. *Gait & posture*, 22(4) :287–294.
- [Cook and Das, 2012] Cook, D. J. and Das, S. K. (2012). Pervasive computing at scale : Transforming the state of the art. *Pervasive and Mobile Computing*, 8(1) :22–35.
- [Cvetković et al., 2018] Cvetković, B., Szeklicki, R., Janko, V., Lutomski, P., and Luštrek, M. (2018). Real-time activity monitoring with a wristband and a smart-phone. *Information Fusion*, 43 :77–93.
- [Ermes et al., 2008] Ermes, M., Parkka, J., and Cluitmans, L. (2008). Advancing from offline to online activity recognition with wearable sensors. In *2008 30th annual international conference of the ieee engineering in medicine and biology society*, pages 4451–4454. IEEE.
- [Fergus et al., 2009] Fergus, R., Weiss, Y., and Torralba, A. (2009). Semi-supervised learning in gigantic image collections. In *NIPS*, volume 1, page 2. Citeseer.
- [Franc et al., 2011] Franc, V., Zien, A., and Schölkopf, B. (2011). Support vector machines as probabilistic models. In *ICML*.
- [Gandetto et al., 2003] Gandetto, M., Marchesotti, L., Sciutto, S., Negroni, D., and Regazzoni, C. S. (2003). From multi-sensor surveillance towards smart interactive spaces. In *2003 International Conference on Multimedia and Expo. ICME'03. Proceedings (Cat. No. 03TH8698)*, volume 1, pages I–641. IEEE.
- [Garcia-Ceja et al., 2018] Garcia-Ceja, E., Riegler, M., Nordgreen, T., Jakobsen, P., Oedegaard, K. J., and Tørresen, J. (2018). Mental health monitoring with multi-modal sensing and machine learning : A survey. *Pervasive and Mobile Computing*, 51 :1–26.
- [George et al., 2011] George, J., Vunckx, K., Tejpar, S., Deroose, C. M., Nuyts, J., Loeckx, D., and Suetens, P. (2011). Fuzzy statistical unsupervised learning based total lesion metabolic activity estimation in positron emission tomography images. In *International Workshop on Machine Learning in Medical Imaging*, pages 233–240. Springer.
- [He and Jin, 2009] He, Z. and Jin, L. (2009). Activity recognition from acceleration data based on discrete cosine transform and svm. In *2009 IEEE International Conference on Systems, Man and Cybernetics*, pages 5041–5044. IEEE.

- [Karantonis et al., 2006] Karantonis, D. M., Narayanan, M. R., Mathie, M., Lovell, N. H., and Celler, B. G. (2006). Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring. *IEEE transactions on information technology in biomedicine*, 10(1) :156–167.
- [Khan et al., 2010] Khan, A. M., Lee, Y.-K., Lee, S.-Y., and Kim, T.-S. (2010). Human activity recognition via an accelerometer-enabled-smartphone using kernel discriminant analysis. In *2010 5th international conference on future information technology*, pages 1–6. IEEE.
- [Kwapisz et al., 2011] Kwapisz, J. R., Weiss, G. M., and Moore, S. A. (2011). Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter*, 12(2) :74–82.
- [Lane et al., 2010] Lane, N. D., Miluzzo, E., Lu, H., Peebles, D., Choudhury, T., and Campbell, A. T. (2010). A survey of mobile phone sensing. *IEEE Communications magazine*, 48(9) :140–150.
- [LeCun et al., 1989] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4) :541–551.
- [Lee et al., 2017] Lee, S.-M., Yoon, S. M., and Cho, H. (2017). Human activity recognition from accelerometer data using convolutional neural network. In *2017 IEEE international conference on big data and smart computing (bigcomp)*, pages 131–134. IEEE.
- [Lin et al., 2008] Lin, W., Sun, M.-T., Poovandran, R., and Zhang, Z. (2008). Human activity recognition for video surveillance. In *2008 IEEE international symposium on circuits and systems*, pages 2737–2740. IEEE.
- [Mannini and Sabatini, 2010] Mannini, A. and Sabatini, A. M. (2010). Machine learning methods for classifying human physical activity from on-body accelerometers. *Sensors*, 10(2) :1154–1175.
- [Maurer et al., 2006] Maurer, U., Smailagic, A., Siewiorek, D. P., and Deisher, M. (2006). Activity recognition and monitoring using multiple sensors on different body positions. In *International Workshop on Wearable and Implantable Body Sensor Networks (BSN’06)*, pages 4–pp. IEEE.
- [Mejia-Ricart et al., 2017] Mejia-Ricart, L. F., Helling, P., and Olmsted, A. (2017). Evaluate action primitives for human activity recognition using unsupervised lear-

- ning approach. In *2017 12th international conference for internet technology and secured transactions (icitst)*, pages 186–188. IEEE.
- [Miao et al., 2015] Miao, F., He, Y., Liu, J., Li, Y., and Ayoola, I. (2015). Identifying typical physical activity on smartphone with varying positions and orientations. *Biomedical engineering online*, 14(1) :1–15.
- [Murphy, 2012] Murphy, K. P. (2012). *Machine learning : a probabilistic perspective*. MIT press.
- [Nurhanim et al., 2017] Nurhanim, K., Elamvazuthi, I., Izhar, L., and Ganesan, T. (2017). Classification of human activity based on smartphone inertial sensor using support vector machine. In *2017 IEEE 3rd international symposium in robotics and manufacturing automation (roma)*, pages 1–5. IEEE.
- [Ortiz, 2015] Ortiz, J. L. R. (2015). Smartphone-based human activity recognition.
- [Parasuraman et al., 2000] Parasuraman, R., Sheridan, T. B., and Wickens, C. D. (2000). A model for types and levels of human interaction with automation. *IEEE Transactions on systems, man, and cybernetics-Part A : Systems and Humans*, 30(3) :286–297.
- [Quinlan, 1986] Quinlan, J. (1986). Induction of decision trees. *mach. learn.*
- [Quinlan, 1993] Quinlan, J. (1993). C4. 5 : Programs for machine learning morgan kaufmann publishers san francisco. *CA Google Sch*, pages 155–164.
- [Ravi et al., 2016] Ravi, D., Wong, C., Lo, B., and Yang, G.-Z. (2016). A deep learning approach to on-node sensor data analytics for mobile or wearable devices. *IEEE journal of biomedical and health informatics*, 21(1) :56–64.
- [Ravi et al., 2005] Ravi, N., Dandekar, N., Mysore, P., and Littman, M. L. (2005). Activity recognition from accelerometer data. In *Aaai*, volume 5, pages 1541–1546. Pittsburgh, PA.
- [Saha et al., 2017] Saha, J., Chakraborty, S., Chowdhury, C., Biswas, S., and Aslam, N. (2017). Designing device independent two-phase activity recognition framework for smartphones. In *2017 IEEE 13th international conference on wireless and mobile computing, networking and communications (wimob)*, pages 257–264. IEEE.
- [Schmidt et al., 1999] Schmidt, A., Aidoo, K. A., Takaluoma, A., Tuomela, U., Van Laerhoven, K., and Van de Velde, W. (1999). Advanced interaction in context. In *International Symposium on Handheld and Ubiquitous Computing*, pages 89–101. Springer.

- [Shalev-Shwartz et al., 2011] Shalev-Shwartz, S. et al. (2011). Online learning and online convex optimization. *Foundations and trends in Machine Learning*, 4(2) :107–194.
- [Stikic et al., 2011] Stikic, M., Larlus, D., Ebert, S., and Schiele, B. (2011). Weakly supervised recognition of daily life activities with wearable sensors. *IEEE transactions on pattern analysis and machine intelligence*, 33(12) :2521–2537.
- [Stikic et al., 2008] Stikic, M., Van Laerhoven, K., and Schiele, B. (2008). Exploring semi-supervised and active learning for activity recognition. In *2008 12th IEEE International Symposium on Wearable Computers*, pages 81–88. IEEE.
- [Tapia et al., 2007] Tapia, E. M., Intille, S. S., Haskell, W., Larson, K., Wright, J., King, A., and Friedman, R. (2007). Real-time recognition of physical activities and their intensities using wireless accelerometers and a heart rate monitor. In *2007 11th IEEE international symposium on wearable computers*, pages 37–40. IEEE.
- [Tapia et al., 2006] Tapia, E. M., Intille, S. S., Lopez, L., and Larson, K. (2006). The design of a portable kit of wireless sensors for naturalistic data collection. In *International Conference on Pervasive Computing*, pages 117–134. Springer.
- [Tran and Phan, 2016] Tran, D. N. and Phan, D. D. (2016). Human activities recognition in android smartphone using support vector machine. In *2016 7th international conference on intelligent systems, modelling and simulation (isms)*, pages 64–68. IEEE.
- [Voicu et al., 2019] Voicu, R.-A., Dobre, C., Bajenaru, L., and Ciobanu, R.-I. (2019). Human physical activity recognition using smartphone sensors. *Sensors*, 19(3) :458.
- [Wang et al., 2016] Wang, A., Chen, G., Yang, J., Zhao, S., and Chang, C.-Y. (2016). A comparative study on human activity recognition using inertial sensors in a smartphone. *IEEE Sensors Journal*, 16(11) :4566–4578.
- [Wang et al., 2012] Wang, Z., Jiang, M., Hu, Y., and Li, H. (2012). An incremental learning method based on probabilistic neural networks and adjustable fuzzy clustering for human activity recognition by using wearable sensors. *IEEE Transactions on Information Technology in Biomedicine*, 16(4) :691–699.
- [Weiser, 1993] Weiser, M. (1993). Some computer science issues in ubiquitous computing. *Communications of the ACM*, 36(7) :75–84.

- [Wolpert and Macready, 1997] Wolpert, D. H. and Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, 1(1) :67–82.
- [Woodman, 2007] Woodman, O. J. (2007). An introduction to inertial navigation. Technical report, University of Cambridge, Computer Laboratory.
- [Wu et al., 2012] Wu, W., Dasgupta, S., Ramirez, E. E., Peterson, C., and Norman, G. J. (2012). Classification accuracies of physical activities using smartphone motion sensors. *Journal of medical Internet research*, 14(5) :e130.
- [Yang and Hsu, 2010] Yang, C.-C. and Hsu, Y.-L. (2010). A review of accelerometry-based wearable motion detectors for physical activity monitoring. *Sensors*, 10(8) :7772–7788.
- [Yang and Yang, 2006] Yang, G.-Z. and Yang, G. (2006). *Body sensor networks*, volume 1. Springer.
- [Yu and Qin, 2018] Yu, S. and Qin, L. (2018). Human activity recognition with smartphone inertial sensors using bidir-lstm networks. In *2018 3rd international conference on mechanical, control and computer engineering (icmce)*, pages 219–224. IEEE.
- [Zheng et al., 2009] Zheng, V. W., Hu, D. H., and Yang, Q. (2009). Cross-domain activity recognition. In *Proceedings of the 11th international conference on Ubiquitous computing*, pages 61–70.
- [Zhu and Sheng, 2009] Zhu, C. and Sheng, W. (2009). Human daily activity recognition in robot-assisted living using multi-sensor fusion. In *2009 IEEE international conference on robotics and automation*, pages 2154–2159. IEEE.