



République Algérienne Démocratique et Populaire
Ministère de l'enseignement supérieur et de la recherche scientifique



Université Larbi Tébessi – Tébessa
Faculté des Sciences Exactes & des Sciences de la Nature et de la Vie
Département de Mathématiques et d'Informatique

MEMOIRE DE MASTER

Filière : Informatique
Spécialité : Réseaux & Sécurité Informatique

Thème

**Reconnaissance des activités humaines
à partir des senseurs de smartphone**

Présenté par :

Mr Achour Rabie
Mr Dris Farid

Encadré par :

Mr Gahmousse A.

Jury de soutenance :

Mr Matrouh A.	Président
Mr Djedai A.	Examineur

Promotion : Juin 2017

Remerciement

Nous tenons avant tout à remercier Dieu tout puissant de nous avoir donné la force et la volonté pour achever ce travail.

Nos vifs remerciements, accompagnés de toute notre gratitude, vont tout d'abord à notre encadreur Mr Abd'Elatif Gahmousse, Pour sa bonne volonté d'accepter de nous encadrer, pour nous avoir dirigé dans notre travail, pour tous les conseils qu'il nous a prodigué.

Nous adressons nos remerciements à messieurs les membres du jury.

Nous remercions également tous nos enseignants du département informatique de l'université de Tébessa.

Nous tenons à remercier aussi l'ensemble du centre d'accueil des personnes âgées d'Oum El Bouaghi pour leur aides et accueils.

Résumé

La reconnaissance des activités humaines (HAR) est un domaine de recherche, dont l'objectif principal est d'identifier les activités réalisées par un sujet en analysant les signaux obtenus à partir de capteurs. La croissance rapide de ce domaine de recherche est à l'origine d'une grande partie des travaux et applications liés au domaine de la santé.

Dans ce contexte, il se pose ce travail de fin d'étude, dont l'objectif principal est de maîtriser et évaluer les nouvelles techniques de collecte de données, d'extraction de caractéristiques, et faire des comparaisons et donc de trouver la méthode la plus adéquate pour la reconnaissance d'activité dans notre champ d'étude, grâce à l'intégration des signaux inertiels obtenue par un capteur généralisé dans la majeure partie des smartphones : l'accéléromètre.

Plus précisément, six types d'activités effectuées par vingt et un sujets sont évalués : marcher, monter les escaliers, descendre les escaliers, assis, debout et allongé. Nous avons réalisés plusieurs expériences impliquant des techniques d'apprentissage automatique supervisées. Puis, entraîner et tester plusieurs classifieurs (k-Nearest Neighbor, Support Vector Machines, Artificial Neural Networks, Decision Tree). Également, nous avons aboutés à la proposition d'un système comprenant cinquante-six fonctionnalités et un classifieur pour identifier et prédire de la manière la plus probable.

Abstract

The recognition of human activities (HAR) is a field of research, the main objective of which is to identify the activities carried out by a subject by analyzing the signals obtained from sensors. The rapid growth of this area of research is responsible for a large part of the work and applications related to the field of health.

In this context, this end-of-study work is aimed at mastering and evaluating new techniques for collecting data, extracting characteristics, making comparisons, and thus finding the method more suitable for the recognition of activity in our field of study, thanks to the integration of the inertial signals obtained by a generalized sensor in the majority of smartphones: the accelerometer.

More specifically, six types of activities carried out by twenty-one subjects are evaluated: walking, climbing stairs, stairs, sitting, standing and lying down. We performed several experiments involving supervised learning techniques. Then, train and test several classifiers (k-Nearest Neighbor, Support Vector Machines, Artificial Neural Networks, Decision Tree). Also, we have been interested in proposing a system comprising fifty-six features and a classifier to identify and predict in the most probable way.

SOMMAIRE

Sommaire

Abstract	
Résumé	
Liste des figures	i
Liste des tableaux	ii
Liste des abréviations	iii
Introduction Générale	1
Chapitre I – Concepts Fondamentaux	4
1. La Data science.....	4
2. Le Machine Learning.....	4
3. Catégorisation des algorithmes	5
Algorithmes supervisés et non supervisés	5
Algorithmes de régression et de classification.....	5
Algorithmes de classification.....	6
3.1.1.k-Nearest Neighbors (k-voisins les plus proches) (kNN)	6
3.3.1.Support Vector Machines (Machine à vecteur de support) (SVM)	6
3.3.2.Artificial Neural Networks (Réseaux de neurones artificiels) (ANN)	7
3.3.3.Decision Tree (Arbre de décision) (DT)	7
4. Les senseurs des smartphone	7
5. État de l'art	8
Conclusion.....	9
Chapitre II – Expérimentation	10
1. Processus d'identification d'activité	10
2. Collecte de données	11
3. Préparation et Traitement des données	13
3.1. Première Phase.....	13
3.2. Deuxième Phase.....	14
3.3. Troisième phase	14
3.4. Quatrième phase.....	14
3.5. Script de préparation et de traitement (Phase 2)	16
3.6. Script de préparation et de traitement (Phase 3)	18
3.7. Visualisation graphique des données (Plot)	20
3.7.1. Plot de x et y (Script)	20
3.7.2. Plot de l'accélération x par activités colorée	21
3.7.3. Plot de l'accélération x - amplitude (marche –allongé)	22
3.7.4. Plot de l'accélération x - amplitude (marcher –debout)	23
3.7.5. Plot de l'accélération x - amplitude (marcher –monter l'escalier)	24
3.7.6. Plot après filtrage numérique	25
3.7.7. Plot de Périodogramme	26

4.	Extraction des caractéristiques.....	27
4.1.	Caractéristiques du domaine temporel (Time-domain Features)	27
4.2.	Caractéristiques du domaine fréquentiel (Frequency-domain Features)	27
5.	Sélection des caractéristiques (Features Selection)	29
6.	Classification	31
6.1.	Résultat de classification.....	32
6.2.	Evaluation des performances	34
6.2.1.	Matrice de confusion.....	34
6.2.2.	Test avec un Dataset certifié	39
a)	Présentation du dataset	39
b)	Résultats obtenus	40
c)	Comparaison des résultats	45
	Conclusion.....	46
	Chapitre III – Implémentation	47
1.	Architecture détaillée de système	47
2.	Configuration utilisée dans l'implémentation	49
3.	Cycle de vie d'une activité Androïde	51
4.	Diagramme UML.....	52
4.1.	Cas d'utilisation.....	52
4.1.1.	Système : Application Androïde	52
4.1.2.	Système : Serveur Central	53
4.2.	Diagramme de classe	53
4.3.	Diagramme d'activité	54
5.	Modules et Fichiers.....	55
5.1.	Python	55
5.2.	PHP	55
5.3.	Interface application Androïde.....	56
5.4.	Interface Page Web	58
	Conclusion.....	59
	Conclusion Générale	60
	Bibliographie	
	Annexes	

Liste des tableaux

Tableau 1	Configuration Utilisée dans l'expérimentation	11
Tableau 2	Filtres de conditionnements et de traitements	15
Tableau 3	Fonctions d'extractions	28
Tableau 4	Caractéristiques obtenus	28
Tableau 5	Caractéristiques retenus après sélection	29
Tableau 6	Précision DECISION TREES	32
Tableau 7	Précision NEAREST NEIGHBOR Classifier	33
Tableau 8	Précision NEURAL NETWORK	33
Tableau 9	Précision SUPPORT VECTOR MACHINE	34
Tableau 10	Comparaison publié des Performances des classificateurs ML	39
Tableau 11	Précision DECISION TREES – 2eme Test	40
Tableau 12	Précision NEAREST NEIGHBOR Classifier – 2eme Test	40
Tableau 13	Précision NEURAL NETWORK – 2eme Test	41
Tableau 14	Précision SUPPORT VECTOR MACHINE – 2eme Test	41
Tableau 15	Comparaison N°1 des Performances des classificateurs ML – 2eme Test	45
Tableau 16	Comparaison N°2 des Performances des classificateurs ML – 2eme Test	45
Tableau 17	Configuration Utilisée dans l'implémentation	49
Tableau 18	Table user_t1	55
Tableau 19	Table user_t2	55

Liste des formules

Formule (1)	31
Formule (2)	31
Formule (3)	31

Liste des figures

Figure 1	Processus d'identification d'activité	10
Figure 2	Position du placement du Smartphone	12
Figure 3	Application de collecte de données	12
Figure 4	Système de coordonnées du Smartphone	13
Figure 5	Visualisation d'un fichier de données	16
Figure 6	Plot de x et y	20
Figure 7	Plot de l'accélération x par activités colorée	21
Figure 8	Plot de l'accélération x - amplitude (marche –allongé)	22
Figure 9	Plot de l'accélération x - amplitude (marcher –debout).....	23
Figure 10	Plot de l'accélération x - amplitude (marcher –monter l'escalier)	24
Figure 11	Plot après filtrage numérique	25
Figure 12	Plot de Périodogramme	26
Figure 13	Graphe d'importance des caractéristiques (Paired t test)	30
Figure 14	Matrice de Confusion TC/PC	35
Figure 15	Matrice de Confusion TPR/FNR	36
Figure 16	Matrice de Confusion PDV/FDR	37
Figure 17	Matrice de Confusion TC/PC– 2eme Test	42
Figure 18	Matrice de Confusion TPR/FNR– 2eme Test	43
Figure 19	Matrice de Confusion PPV/FDR– 2eme Test	44
Figure 20	Architecture générale	47
Figure 21	Architecture détaillée	48
Figure 22	Environnement virtuel.....	50
Figure 23	Eclipse EE Pydev	50
Figure 24	Cycle de vie d'une activité.....	51
Figure 25	Diagramme Cas d'utilisation : application	52
Figure 26	Diagramme Cas d'utilisation : Serveur central	53
Figure 27	Diagramme de classe.....	53
Figure 28	Diagramme d'activité.....	54
Figure 29	Interface application Android 1 – 2 - 3	56
Figure 30	Interface application Android 4 – 5 - 6	57
Figure 31	Interface Page Web Version Smartphone	58
Figure 32	Interface Page Web version PC.....	58

Liste des abréviations

kNN	k-Nearest Neighbor
SVM	Support Vector Machines
NB	Naïve Bayes
RF	Random forest
ANN	Artificial Neural Networks
DT	Decision Tree
RF	Random forest
LR	Logistic regression
MLP	Multilayer perceptron
HAR	Human Activity Recognition
TXT	Text
Hz	Hertz
fs	Frequency Sample
RMS	Root Mean Square
Std	Standard Deviation
PSD	Power Spectral Density
acc	Acceleration
ML	Machine Learning
BDD	Base de Données
PHP	Hypertext Preprocessor
HTML	Hypertext Markup Language
CSS	Cascading Style Sheets
JS	Java Script
UML	Unified Modeling Language
EE	Entreprise Edition
GPU	Graphical Processor Unit

INTRODUCTION

GENERALE

Un système d'identification de l'activité humaine (HAR) peut reconnaître automatiquement les activités physiques, qui est un enjeu de recherche clé dans l'informatique mobile et omniprésente. Un système HAR effectue des tâches de reconnaissance des différentes activités quotidiennes humaines, du simple au complexe. Les capteurs impliqués dans un système HAR peuvent être des capteurs vidéo, des capteurs d'inertie et des capteurs d'environnement.

Le HAR et les Smartphones :

Les applications basées sur les Smartphones sont une solution de plus en plus importante car les Smartphones sont devenus un élément indispensable de notre vie quotidienne. Particulièrement avec l'évolution rapide du matériel, la capacité croissante de calcul et de réseau, et les capteurs intégrés riches, les systèmes HAR basés sur Smartphone peuvent nous dire différents types d'activités humaines en temps réel en utilisant des techniques d'apprentissage de machine. [1]

Tout cela bénéficie du développement rapide des logiciels et du matériel de téléphonie mobile. Les téléphones intelligents sur le marché ont intégré des capteurs, et le MEMS avancé (micro-Électromécaniques) a permis des capteurs de faible puissance et de haute qualité pour la détection mobile. Les capteurs MEMS les plus connus des Smartphones sont l'accéléromètre et le gyroscope, mais il y a beaucoup plus de capteurs MEMS dans les appareils mobiles d'aujourd'hui comme la boussole électronique, le capteur de pression, le capteur de lumière et le microphone.[2]

Les accéléromètres sont une solution pratique et peu coûteuse pour surveiller l'activité physique. Des accéléromètres triaxiaux sont attachés à l'orthèse et/ou à la hanche du sujet pour acquérir des données d'accélération. Ces ensembles de données sont utilisés pour évaluer les différentes activités physiques quotidiennes, car «l'accélérométrie est préférable parce que l'accélération est proportionnelle à la force externe et peut donc refléter l'intensité et la fréquence du mouvement humain» [3].

L'accéléromètre est un instrument qui mesure l'accélération physique expérimentée d'un objet. Il a été utilisé pour plusieurs applications en science, médecine, technologie et industrie comme pour des vibrations de mesure dans les machines, accélération dans des véhicules à grande vitesse et charges mobiles sur des ponts. Pour quels soucis à HAR, l'accéléromètre est l'une des sondes les plus employées couramment pour des signaux de mouvement de corps de lecture [4].

Le HAR et la santé :

Le mode de vie sédentaire devient dominant particulièrement pour les activités intellectuelles. Sachant que ce mode peut être une cause majeure de beaucoup de maladies d'ordre physique et psychique ; surtout qu'il n'est pas évident d'adopter un style de vie plus sain vu le manque de

procédés et systèmes appropriés destinés au rappel, contrôle et surveillance des activités physiques des individus.

Reconnaissance de l'activité humaine (HAR) est un domaine de recherche émergent dans le but d'identifier les actions menées par une personne donnée. La forte croissance dans ce domaine de recherche au sein de la communauté scientifique s'explique principalement par le nombre élevé d'applications qui se sont développées ces dernières années. Une grande partie des applications les plus prometteuses sont liées au domaine de la santé, où il est possible de suivre la mobilité des patients souffrant de dysfonctionnement moteur ainsi que l'activité physique chez les patients présentant un risque cardiovasculaire.[5]

Comme rapporté par l'Organisation mondiale de la Santé, 3,2 millions de décès annuels étaient liés au style de vie physique d'inactivité. En 2004, c'est le quatrième facteur de risque qui a causé des décès dans le monde. Plus les personnes sont inactives, plus elles risquent beaucoup de maladies chroniques. [6]

Afin de faire empêcher les maladies cardiaques, les gens à tous les âges devraient pratiquer des activités physiques au moins trente minutes par jour selon le Département des États-Unis des services de la santé et des affaires sociales. [7]

Des patients présentant le diabète, l'obésité, ou la maladie cardiaque sont souvent requis de suivre une routine bien définie d'exercice en tant qu'élément de leur traitement. Par conséquent, l'identification des activités telles que la marche, le fonctionnement, ou le repos devient tout à fait utile pour fournir la rétroaction au travailleur social au sujet du comportement du patient. De même, des patients présentant la démence et d'autres pathologies mentales pourraient être surveillés pour détecter des activités anormales et pour empêcher de ce fait les conséquences indésirables.

Objectifs de notre mémoire :

Depuis quelques années, en utilisant différents types de capteurs, un suivi du patient était possible, tel que l'enregistrement quotidien de la condition physique, les alertes anormales de situation, l'identification de la signature biométrique personnelle, la localisation et la navigation intérieure.

Toutefois, loin d'être une solution à long terme et avec l'irruption du smartphone, cette surveillance peut être réalisée de manière *non invasive* en utilisant les capteurs du smartphone intégré.

Dans notre étude, " *le centre d'accueil des personnes âgées d'Oum El Bouaghi* " a été notre cible. Il comporte dans sa structure une administration, un centre de soins médicaux, un restaurant,

une cafeteria, une bibliothèque, des bungalows ainsi que des espaces verts et des aires de repos et d'activités.

Ce centre a atteint une capacité d'accueil de 80 personnes ; néanmoins on se décline sur un problème majeur qui est le manque d'assistance proche de personnel qualifié pour assurer la sécurité et l'assurance d'une activité physique quotidienne de minimum 20 minutes, d'où vient l'importance d'assurer une solution à distance utilisant un système de reconnaissance d'activité humaine (HAR) par le biais d'un smartphone.

Le premier objectif de notre travail est de maîtriser et évaluer les nouvelles techniques de collecte de données, d'extraction de caractéristiques, et faire des comparaisons et donc de trouver la méthode la plus adéquate pour la reconnaissance d'activité dans notre champ d'étude "Le centre d'accueil" grâce à l'intégration des signaux inertiels obtenue par un capteur généralisé dans la majeure partie des smartphones : l'accéléromètre. En particulier, six activités différentes sont évaluées : marche, monter l'escalier, descendre en bas, assis, debout et allongé.

Le deuxième objectif est de développer une application de reconnaissance d'activité humaine qui assure l'acheminement des données à partir du smartphone vers un serveur et la reconnaissance online de l'activité en cours, d'où l'avantage de confirmer la pratique de l'exercice quotidien formellement indique.

Pour l'organisation de notre mémoire, nous l'avons débuté par cette introduction générale, il comporte aussi trois chapitres : le Premier chapitre contient les concepts Fondamentaux, le second consiste à la phase d'expérimentation et le troisième est consacré à l'implémentation. Et nous terminons notre travail par une conclusion générale.

CHAPITRE I

CONCEPTS

FONDAMENTAUX

1. La Data science :

La *data science* est une démarche empirique qui se base sur des données pour apporter une réponse à des problèmes et n distingue généralement les données quantitatives des données qualitatives.

Les données quantitatives sont des valeurs qui décrivent une quantité mesurable, sous la forme de nombres sur lesquels on peut faire des calculs (moyenne, etc.) et des comparaisons (égalité/différence, infériorité/supériorité, etc.). Elles répondent typiquement à des questions du type « combien ». On fait parfois la différence entre :

- les données quantitatives continues, qui peuvent prendre n'importe quelle valeur dans un ensemble de valeurs : la température, le taux de chômage ;
- les données quantitatives discrètes, qui ne peuvent prendre qu'un nombre limité de valeurs dans un ensemble de valeurs : le nombre d'enfants par famille, le nombre de pièces d'un logement, etc.

Les données qualitatives décrivent quant à elles des qualités ou des caractéristiques. Elles répondent à des questions de la forme « quel type » ou « quelle catégorie ». Ces valeurs ne sont plus des nombres, mais un ensemble de modalités. On ne peut pas faire de calcul sur ces valeurs, même dans l'éventualité où elles prendraient l'apparence d'une série numérique. Elles peuvent toutefois être comparées entre elles et éventuellement triées. On distingue :

- les données qualitatives nominales (ou catégorielles), dont les modalités ne peuvent être ordonnées. Par exemple : la couleur des yeux (bleu, vert, marron, etc.), le sexe (homme, femme) ;
- les données qualitatives ordinales, dont les modalités sont ordonnées selon un ordre « logique ». Par exemple : les tailles de vêtements (S, M, L, XL), le degré d'accord à un test d'opinion (fortement d'accord, d'accord, pas d'accord, fortement pas d'accord).[8]

2. Le Machine Learning :

Le *machine Learning* n'est pas un domaine d'étude récent. On en trouve une première définition dès 1959, due à Arthur Samuel, l'un des pionniers de l'intelligence artificielle, qui définit le *machine Learning* comme le champ d'étude visant à donner la capacité à une machine d'apprendre sans être explicitement programmée. En 1997, Tom Mitchell, de l'université de Carnegie Mellon, propose une définition plus précise :

« On dit qu'un programme d'ordinateur apprend de l'expérience E par rapport à une classe de tâches T et une mesure de performance P , si sa performance aux tâches en T , mesurée par P , s'améliore avec l'expérience E ».

Avec le *machine learning*, on passe d'une informatique impérative basée sur des hypothèses à une informatique probabiliste basée sur des informations réelles.

L'objectif du machine learning est d'apprendre à partir de données issues d'observations réelles.[9]

3. Catégorisation des algorithmes

Les algorithmes visent à en extraire des régularités qui permettront l'apprentissage, ils ne sont pas tous destinés aux mêmes usages. On les classe usuellement selon deux composantes :

- le mode d'apprentissage : on distingue les algorithmes supervisés des algorithmes non supervisés ;
- le type de problème à traiter : on distingue les algorithmes de régression de ceux de classification.[10]

3.1. Algorithmes supervisés et non supervisés :

La différence entre algorithmes supervisés et non supervisés est fondamentale. Les algorithmes supervisés extraient de la connaissance à partir d'un ensemble de données contenant des couples entrée-sortie. Ces couples sont déjà « connus », dans le sens où les sorties sont définies a priori. La valeur de sortie peut être une indication fournie par un expert : par exemple, des valeurs de vérité de type OUI/NON ou MALADE/SAIN. Ces algorithmes cherchent à définir une représentation compacte des associations entrée-sortie, par l'intermédiaire d'une fonction de prédiction. Les algorithmes non supervisés n'intègrent pas la notion d'entrée-sortie. Toutes les données sont équivalentes (on pourrait dire qu'il n'y a que des entrées). Dans ce cas, les algorithmes cherchent à organiser les données en groupes. Chaque groupe doit comprendre des données similaires et les données différentes doivent se retrouver dans des groupes distincts.

3.2. Algorithmes de régression et de classification :

La distinction régression/classification se fait au sujet des algorithmes supervisés. Elle distingue deux types de valeurs de sorties qu'on peut chercher à traiter. Dans le cadre d'un problème de régression, Y peut prendre une infinité de valeurs dans l'ensemble continu des réels (noté $Y \in \mathbb{R}$). Ce peut être des températures, des tailles, des taux de chômage, ou tout autre type de mesure n'ayant pas de valeurs finies a priori.

Dans le cadre d'un problème de classification, Y prend un nombre fini k de valeurs ($Y = \{1, \dots, k\}$). On parle alors d'étiquettes attribuées aux valeurs d'entrée. C'est le cas des valeurs de vérité de type OUI/NON ou MALADE/SAIN.

Pour utiliser ces algorithmes, les données doivent être mises en forme sous une représentation matricielle.

L'objectif du *machine learning* est d'apprendre à partir de données issues d'observations réelles.

Les données peuvent être de diverses natures et provenir de différentes sources. Selon le cas, elles sont plus ou moins complexes à manipuler.

Les algorithmes visent à en extraire des régularités qui permettront l'apprentissage. On distingue :

- les algorithmes supervisés (quand il existe une valeur à prédire) des algorithmes non supervisés (quand il n'y a pas de valeur à prédire) ;
- les algorithmes de régression (quand la valeur à prédire est continue) des algorithmes de classification (quand la valeur à prédire est catégorielle).

Pour utiliser ces algorithmes, les données doivent être mises en forme sous une représentation matricielle.[11]

3.3. Algorithmes de classification :

Les méthodes de classification sont appliquées pour reconnaître les modèles d'activité humaine dans les données d'accélération. Plusieurs solutions avec différentes méthodes pour ce processus existent comme : k-Nearest Neighbor(kNN), Support Vector Machines(SVM), Naives Bayes(NB), Random forest(RF), Artificial Neural Networks(ANN), Decision Tree (DT)...

Pour notre expérimentation nous avons choisies quatre algorithmes que nous présentons brièvement :

3.3.1. k-Nearest Neighbors (k-voisins les plus proches) (kNN)

L'algorithme k-voisins les plus proches est une technique non paramétrique de classification et de régression. Il est aussi l'un des algorithmes d'apprentissage machine les plus simples. Le kNNs est un type de classificateur basé sur l'instance, ce qui signifie qu'il ne résume aucune information des données d'entraînement pendant la phase de formation. Pour un exemple inconnu, il est étiqueté comme la même classe que celle de la majorité de ses k voisins les plus proches. Les voisins les plus proches sont mesurés par une fonction de distance. [12]

3.3.2. Support Vector Machines (Machine à vecteur de support) (SVM)

Le procédé de la machine à vecteur de support est considéré comme un algorithme d'optimisation. Les idées de base des SVMs sont qu'elles créent un hyper plan optimal pour des motifs linéairement séparables, et pour les motifs qui ne sont pas séparables linéairement, les fonctions du noyau peuvent être utilisées pour transformer les données originales en un nouvel espace. Les SVM et les méthodes du noyau ont un modèle théorique qui garantit la performance.[13]

3.3.3. Artificial Neural Networks (Réseaux de neurones artificiels) (ANN)

Les RNA sont des modèles composés de plusieurs nœuds de calculs non linéaires en parallèle et fonctionnent comme des réseaux de neurones biologiques [14]. Le modèle le plus connu des RNA est le réseau neuronal d'alimentation, qui est également connu sous le nom de perceptron multicouche. Les nœuds ou neurones du réseau neuronal sont reliés par des poids et mis à jour pendant le processus d'apprentissage.[15]

3.3.4. Decision Tree (Arbre de décision) (DT)

Arbre de Décision (DT) est un non-paramétrique méthode d'apprentissage supervisé pour la classification. Il existe plusieurs implémentations d'arbre de décision. Parmi eux, ID3 (Iterative Dichotomiser 3) a été développé par Ross Quinlan en 1986 [16]. Il calcule d'abord l'entropie (ou le gain d'information) pour chaque entité dans l'ensemble de données S. Ensuite, il divise l'ensemble de données S en sous-ensembles selon la caractéristique qui a l'entropie minimale (ou le gain d'information). Ensuite, un nœud d'arbre de décision est généré contenant cette fonctionnalité. Cette procédure se répète jusqu'à ce qu'il n'y ait pas de fonctionnalités restantes.[17]

4. Les senseurs des smartphones :

Dans les environnements Android, les capteurs les plus utilisés et les plus installés peuvent être classés comme suit [18] :

- Capteurs de mouvement : les capteurs de mouvement sont basés sur la force d'inertie.
- Capteurs environnementaux : ces capteurs mesurent des paramètres environnementaux, comme la température et la pression, à l'aide de baromètres ou de thermomètres.
- Capteurs de position : ces capteurs comprennent des capteurs d'orientation et des magnétomètres, mesurant la position physique de l'appareil.

5. État de l'art

La reconnaissance de l'activité humaine (HAR) peut être considérée comme un problème de reconnaissance de formes, où un système extrait les caractéristiques des signaux de capteurs, génère un modèle pour chaque activité et classe les activités suivantes en fonction de ces modèles. Habituellement, les activités à reconnaître sont des activités quotidiennes courantes telles que la marche, le jogging, la marche à l'étage, la marche en bas, assis, debout, la conduite, la bicyclette, la course et ainsi de suite.

De nos jours, les différentes solutions peuvent être classées selon plusieurs critères. Le premier est basé sur le type de capteur utilisé, leur emplacement (sur le corps, les capteurs placés ou ambiants) et le type de signaux mesurés (par exemple inertiel ou vision). Le second est lié au

principe de modélisation, qui peut être basé sur les données ou sur la connaissance, selon que les modèles HAR sont construits à partir de jeux de données préexistants ou de l'exploitation des connaissances antérieures concernant un domaine particulier. La troisième est l'approche d'apprentissage, qui peut être soit supervisée, semi-supervisée ou non supervisée, et la dernière, selon les algorithmes de classification utilisés.

Plusieurs approches d'apprentissage automatique ont déjà été appliquées à la reconnaissance d'activité. Certains auteurs ont utilisé des classificateurs heuristiques simples [19], tandis que d'autres ont employé des méthodes plus automatiques, dont Naïve Bayes [20], Decision Tree [21], Support Vector Machines (SVM), Les réseaux bayésiens et les réseaux neuronaux [22]. D'autres chercheurs ont également utilisé l'outil d'apprentissage WEKA pour réduire le bruit présent dans les données d'accéléromètre collectées par téléphone et pour comparer les taux d'exactitude de plusieurs approches d'apprentissage automatique:

Une autre approche récente est la combinaison de différents types de capteurs. Dans [23], les auteurs combinent l'accéléromètre de smartphone apparié avec un capteur de poitrine. Les chercheurs ont utilisé des algorithmes de classification C4.5, SVM, Multi-Layer Perceptron et Naïve Bayes.

Alors que certaines études révèlent que les activités de petite durée sont mieux modélisées avec les réseaux neuronaux, les systèmes Neuro-flous et C4.5 certains autres obtenir de meilleurs résultats avec un arbre de décision bien taillée. [24]

Des activités simples peuvent être modélisées avec précision. Cependant, les activités complexes ou peu familières sont souvent difficiles à comprendre et le modèle dû à ces chaînes requiert une connaissance spécifique de l'application.

Conclusion :

La conclusion que nous pouvons attirer de ce que nous avons vus dans ce chapitre est que le domaine de recherche de la reconnaissance de l'activité humaine est en pleine explantions, et beaucoup de futur travaux peuvent être entamé pour découvrir, améliorer des fonctionnalités, des méthodes ou même de nouvelle applications pour la vie quotidienne.

CHAPITRE II

EXPÉRIMENTATION

Dans ce chapitre, nous procédons à la conception d'un système d'identification d'activités humaines basé sur l'accéléromètre d'un Smartphone au sein du **Centre d'accueil des personnes âgées**, qui peut être une aide de valeur à la surveillance et à l'évaluation de leurs différentes activités physiques quotidiennes. Ces activités comprennent la marche, la station debout, la station assis, s'allonger, monter et descendre l'escalier.

Nous allons utiliser plusieurs techniques : d'acquisition et de traitement de données, d'extraction et de sélection de caractéristiques, de classification, tout en analysant les résultats issus de notre expérimentation.

La collecte de données et l'extraction de caractéristiques sont effectuées à l'aide d'une application basée sur le Smartphone tandis que la génération du modèle est effectuée dans un ordinateur en utilisant un processus d'apprentissage supervisé, ensuite le modèle est implémenté dans une autre application pour la prédiction d'activité.

1. Processus d'identification d'activité :

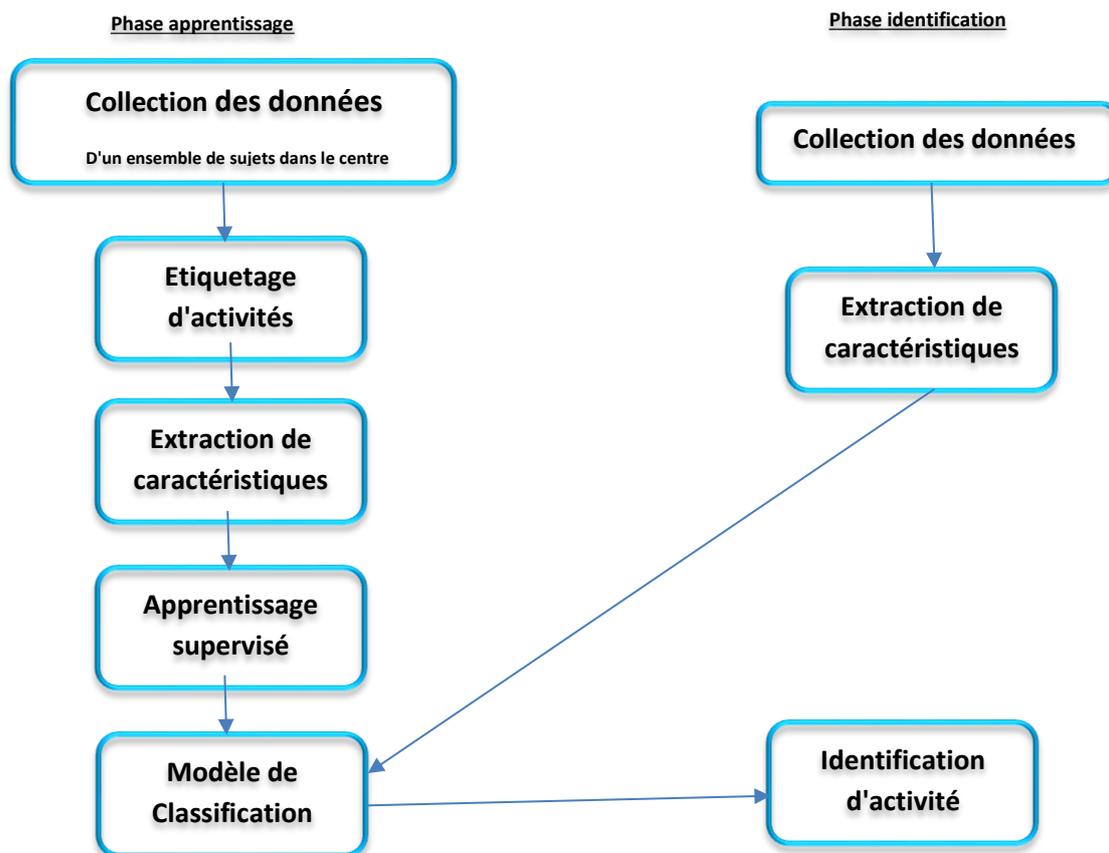


Figure 1 : Processus d'identification d'activité

2. Collecte de données :

Dans la phase d'apprentissage nous avons collecté les données à partir de 21 sujets (9 femmes et 12 hommes) âgées entre 42 et 73 ans de tailles différentes. Pour collecter les données du capteur, un sujet place un Smartphone dans une pochette (**figure 2**) avec une position fixe et effectue certaines activités.

Pour rendre les données faciles à analyser, la position du smartphone est fixée lors de l'exécution des tâches de collecte de données. Une application Android est en cours d'exécution au cours des expériences. Les données des capteurs sont sauvegardées sous forme de fichiers TXT.

Afin d'augmenter l'exactitude de la classification, les données enregistrées au début et à la fin de chaque activité sont coupées des fichiers de données.

La **figure 3** illustre l'application de collecte de données qui sont les lectures brutes de l'accéléromètre sur trois axes et la **figure 4** illustre le système de coordonnées utilisé.

Sujet :	21 sujets
Smartphone :	Oppo A37f / QualcommMSM8916 Quad Cœur / Ram 2Go
Capteur :	Accéléromètre : lis3dh-accel de STMicroelectronics 0.01m/sec ² , max range 156.8, min delay : 5000 microseconds
Position :	Horizontale, Ventrale, dans une pochette
Application :	Accelerometer Analyzer 16.11.27 (Proposé par Mobile Tools)
Plateforme :	Android 4.1, Windows 10
Logiciel et Environnement :	Matlab R 2016 a (64bit)
Activités :	Marche, Assis, Debout, Allongé, Monte l'escalier, Descend l'escalier
La fréquence d'échantillonnage :	50Hz / Sensor speed : Game / Sensor unit : meters

Tableau 1 : Configuration Utilisée dans l'expérimentation

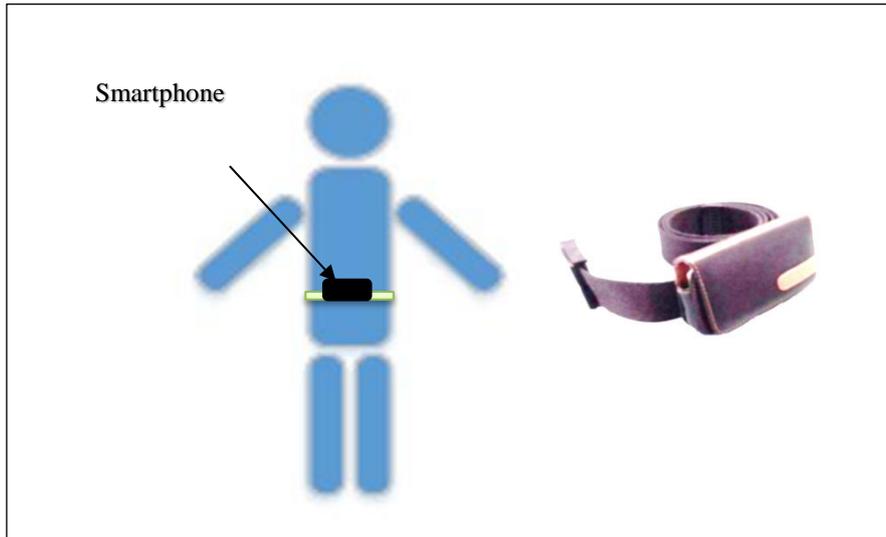


Figure 2 : Position du placement du Smartphone

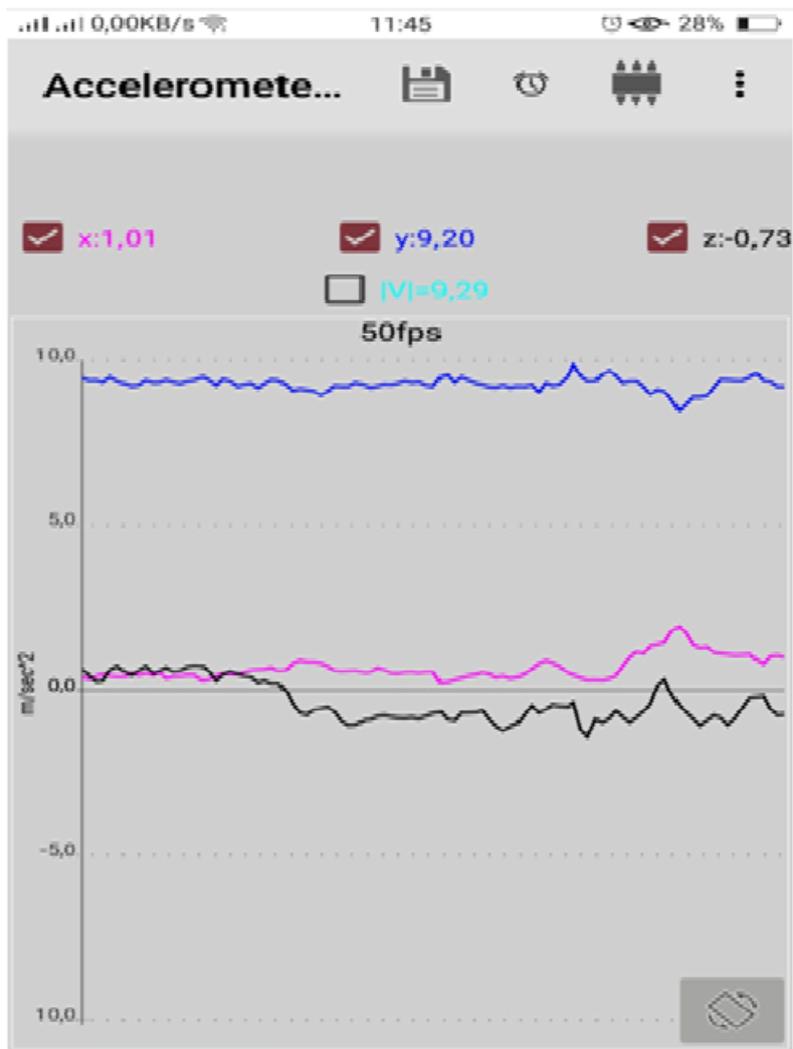


Figure 3 : Application de collecte de données

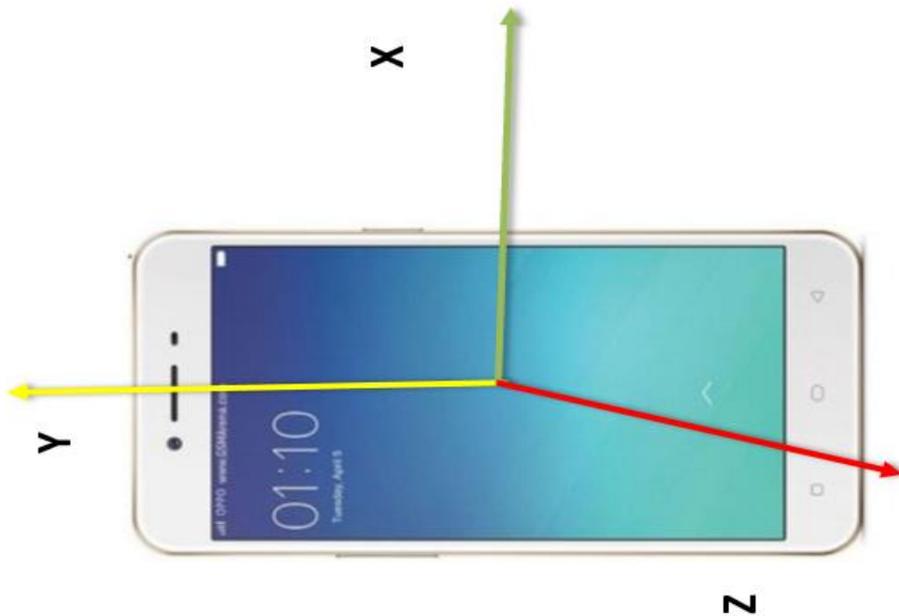


Figure 4 : Système de coordonnées du Smartphone

3. Préparation et Traitement des données :

Dans notre expérimentation nous avons procédé aux étapes suivantes :

3.1. Première Phase : Les étapes sous citées de 1 à 6 représentent la première phase de préparation, qui est réalisée manuellement :

1. Collecte de données à partir de 21 sujets, selon les conditions et les protocoles mentionnés dans : **Tableau 1, figure 2, figure 3** et expliqués dans (**3. Collecte de données**)
2. Importation des enregistrements (format .TXT) dans **Matlab** : les données enregistrées comme l'illustre la **figure 5** comprennent quatre colonnes séparées par un espace (x,y, z, t)
3. Suppression de la quatrième colonne (t).
4. Plot, observation et indication des points de début et fin de chaque activité (en se référant à l'ordre et au timing de réalisation d'activités de chaque sujet).
5. Ajout de l'étiquette id correspondante à chaque activité comme quatrième colonne (x, y, z, a) comme suit :
(1 → Marcher, 2 → Monter escalier, 3 → Descendre escalier, 4 → Assis,
5 → Debout, 6 → Allongé, 0 → Transition ou inconnu)
6. Suppression des activités de Transition ou inconnus (id=0)

3.2. Deuxième Phase : Les étapes sous citées de 7 à 15 représentent la deuxième phase de préparation, ou nous effectuons un échantillonnage des fenêtres dans une forme de fenêtre

coulissante à largeur fixe. Nous utilisons une fenêtre de l'activité d'une seconde ($1 \text{ s} \times 50 \text{ Hz} = 50$ samples) guidée par la raison suivante "Les tailles de fenêtres plus longues n'ont pas été préférées car elles augmenteraient les temps de latence dans la prédiction des activités lorsqu'elles seraient utilisées en ligne" [25]. (Etapes réalisées par script 1) :

7. Chargement des données d'accélération (total) : 4 colonnes (x, y, z, a)
8. Transformation des lignes en une matrice de 50 lignes (window =50)
9. Création d'une colonne pour chaque sujet selon la longueur de ses colonnes
10. Création de la colonne activité correspondante
11. Concaténation de tous les x des sujets
12. Concaténation de tous les y des sujets
13. Concaténation de tous les z des sujets
14. Concaténation de toutes les activités des sujets
15. Concaténation de tous les sujets

3.3. Troisième phase : Les étapes sous citées de 16 à 19 représentent la troisième phase de préparation afin de prétraiter les signaux bruts de l'accéléromètre par les filtres de conditionnement. Tout d'abord, la réduction du bruit avec un filtre **median** de 3ième ordre, puis suivie d'un filtre de **Butterworth low-pass** de 3ième ordre, l'application de ces filtres est représentée par la fonction **matlab Lowp ()**. (Etapes réalisées par script 2) :

16. Chargement sujets - activités
17. Réduction du bruit par filtre médian et Butterworth low-passe
18. Enregistrement de Fréquence d'échantillonnage (Hz) et vecteur de temps
19. Création des fichiers Accelerations.mat, AccBySubject.mat

3.4. Quatrième phase : Les étapes sous citées de 20 à 26 représentent la phase Quatrième de préparation et de traitement, où les données seront plotées et visualisées afin de les analyser et de les caractériser, Le signal d'accélération est encore traité car il est interprété comme l'effet combiné de la force gravitationnelle et l'accélération due au mouvement du corps. Par conséquent, en supposant que la composante gravitationnelle affecte uniquement les fréquences les plus basses, il est possible de séparer le signal d'accélération du mouvement du corps ($a(t)$) par le filtrage **high-pass (a(t))**. La fonction **matlab fdafilter** a été générée automatiquement à partir de l'outil de conception et d'analyse de filtre.

- 20. Plot de x et y
- 21. Plot de l'accélération x par activités colorée
- 22. Plot de l'accélération x - amplitude (marche –allongé)
- 23. Plot de l'accélération x - amplitude (marcher –debout
- 24. Plot de l'accélération x - amplitude (marcher –monter l'escalier)
- 25. Plot après filtrage numérique
- 26. Plot de Périodogramme

Le tableau ci-après récapitule quelque opérations de filtrages que nous utilisons dans la préparation et traitement des données :

Name	Symbol	Formulation
Signaux brut (Raw sensor signals)	$(a_r(t))$	
Accélération Totale (Total acceleration)	$a_t(t)$	filter (Lowp, $a_r(t)$)
Accélération du corps (Body acceleration)	$a_b(t)$	filter (fdafiltar, $a_t(t)$)
A-coup du corps (Body jerk)	$a_j(t)$	$d(a_b(t))/dt \quad dt=1/fs$
Body acc magnitude Total acc magnitude jerk acc magnitude	$amag_b(t)$ $amag_t(t)$ $amag_j(t)$	$\ a_b(t)\ = \sqrt{x^2 + y^2 + z^2}$ $\ a_t(t)\ $ $\ a_j(t)\ $

Tableau 2 : Filtres de conditionnements et de traitements



```

1 10.211 -2.6458 0.2332 5
2 10.217 -2.7461 0.74818 5
3 10.19 -2.87 1.4462 5
4 10.344 -2.8673 1.372 5
5 10.085 -2.803 1.1762 5
6 9.8676 -2.7287 1.316 5
7 9.4378 -2.5372 1.3079 5
8 9.1123 -2.446 1.3973 5
9 9.2481 -2.579 1.5782 5
10 9.3323 -2.5781 1.3894 5
11 9.8234 -2.5247 1.2018 5
12 9.9805 -2.6363 1.237 5
13 9.699 -2.7536 1.3659 5
14 9.869 -2.8674 1.5615 5
15 9.825 -2.7946 1.5716 5
16 9.8057 -2.53 1.4942 5
17 9.8958 -2.4949 1.4862 5
18 9.7224 -2.5677 1.4023 5
19 9.7716 -2.5382 1.3782 5
20 9.7291 -2.5025 1.4432 5
21 9.6083 -2.4828 1.456 5
22 9.7003 -2.5825 1.447 5
23 9.813 -2.6528 1.3719 5
24 9.8747 -2.4763 1.3289 5
25 9.736 -2.3472 1.4598 5
26 9.6899 -2.5322 1.5369 5
27 9.6127 -2.7026 1.4852 5

```

Figure 5 : Visualisation d'un fichier de données

3.5. Script de préparation et de traitement (Phase 2):

❖ Script de Chargement des données d'accélération (total) :

```

[x1,y1,z1,a1] = importfile ('subject1.txt');
[x2,y2,z2,a2] = importfile ('subject2.txt');
[x3,y3,z3,a3] = importfile ('subject3.txt');
[x4,y4,z4,a4] = importfile ('subject4.txt');
[x5,y5,z5,a5] = importfile ('subject5.txt');
[x6,y6,z6,a6] = importfile ('subject6.txt');
[x7,y7,z7,a7] = importfile ('subject7.txt');
[x8,y8,z8,a8] = importfile ('subject8.txt');
[x9,y9,z9,a9] = importfile ('subject9.txt');
[x10,y10,z10,a10] = importfile ('subject10.txt');
[x11,y11,z11,a11] = importfile ('subject11.txt');
[x12,y12,z12,a12] = importfile ('subject12.txt');
[x13,y13,z13,a13] = importfile ('subject13.txt');
[x14,y14,z14,a14] = importfile ('subject14.txt');
[x15,y15,z15,a15] = importfile ('subject15.txt');
[x16,y16,z16,a16] = importfile ('subject16.txt');
[x17,y17,z17,a17] = importfile ('subject17.txt');
[x18,y18,z18,a18] = importfile ('subject18.txt');
[x19,y19,z19,a19] = importfile ('subject19.txt');
[x20,y20,z20,a20] = importfile ('subject20.txt');
[x21,y21,z21,a21] = importfile ('subject21.txt');

```

❖ **Script de Transformation des lignes en une matrice de 50 lignes (window =50):**

```

x_1=vec2mat(x1,50);y_1=vec2mat(y1,50);z_1=vec2mat(z1,50);a_1=vec2mat(a1,50);
x_2=vec2mat(x2,50);y_2=vec2mat(y2,50);z_2=vec2mat(z2,50);a_2=vec2mat(a2,50);
x_3=vec2mat(x3,50);y_3=vec2mat(y3,50);z_3=vec2mat(z3,50);a_3=vec2mat(a3,50);
x_4=vec2mat(x4,50);y_4=vec2mat(y4,50);z_4=vec2mat(z4,50);a_4=vec2mat(a4,50);
x_5=vec2mat(x5,50);y_5=vec2mat(y5,50);z_5=vec2mat(z5,50);a_5=vec2mat(a5,50);
x_6=vec2mat(x6,50);y_6=vec2mat(y6,50);z_6=vec2mat(z6,50);a_6=vec2mat(a6,50);
x_7=vec2mat(x7,50);y_7=vec2mat(y7,50);z_7=vec2mat(z7,50);a_7=vec2mat(a7,50);
x_8=vec2mat(x8,50);y_8=vec2mat(y8,50);z_8=vec2mat(z8,50);a_8=vec2mat(a8,50);
x_9=vec2mat(x9,50);y_9=vec2mat(y9,50);z_9=vec2mat(z9,50);a_9=vec2mat(a9,50);
x_10=vec2mat(x10,50);y_10=vec2mat(y10,50);z_10=vec2mat(z10,50);a_10=vec2mat(a10,50);
x_11=vec2mat(x11,50);y_11=vec2mat(y11,50);z_11=vec2mat(z11,50);a_11=vec2mat(a11,50);
x_12=vec2mat(x12,50);y_12=vec2mat(y12,50);z_12=vec2mat(z12,50);a_12=vec2mat(a12,50);
x_13=vec2mat(x13,50);y_13=vec2mat(y13,50);z_13=vec2mat(z13,50);a_13=vec2mat(a13,50);
x_14=vec2mat(x14,50);y_14=vec2mat(y14,50);z_14=vec2mat(z14,50);a_14=vec2mat(a14,50);
x_15=vec2mat(x15,50);y_15=vec2mat(y15,50);z_15=vec2mat(z15,50);a_15=vec2mat(a15,50);
x_16=vec2mat(x16,50);y_16=vec2mat(y16,50);z_16=vec2mat(z16,50);a_16=vec2mat(a16,50);
x_17=vec2mat(x17,50);y_17=vec2mat(y17,50);z_17=vec2mat(z17,50);a_17=vec2mat(a17,50);
x_18=vec2mat(x18,50);y_18=vec2mat(y18,50);z_18=vec2mat(z18,50);a_18=vec2mat(a18,50);
x_19=vec2mat(x19,50);y_19=vec2mat(y19,50);z_19=vec2mat(z19,50);a_19=vec2mat(a19,50);
x_20=vec2mat(x20,50);y_20=vec2mat(y20,50);z_20=vec2mat(z20,50);a_20=vec2mat(a20,50);
x_21=vec2mat(x21,50);y_21=vec2mat(y21,50);z_21=vec2mat(z21,50);a_21=vec2mat(a21,50);

L1 = size(x_1,1);      L2 = size(x_2,1);      L3 = size(x_3,1);
L4 = size(x_4,1);      L5 = size(x_5,1);      L6 = size(x_6,1);
L7 = size(x_7,1);      L8 = size(x_8,1);      L9 = size(x_9,1);
L10 = size(x_10,1);    L11 = size(x_11,1);    L12 = size(x_12,1);
L13 = size(x_13,1);    L14 = size(x_14,1);    L15 = size(x_15,1);
L16 = size(x_16,1);    L17 = size(x_17,1);    L18 = size(x_18,1);
L19 = size(x_19,1);    L20 = size(x_20,1);    L21 = size(x_21,1);

```

❖ **Création d' une colonne pour chaque sujet selon la longueur de ses colonnes :**

```

s1(1:L1,1)=1;          s2(1:L2,1)=2;          s3(1:L3,1)=3;
s4(1:L4,1)=4;          s5(1:L5,1)=5;          s6(1:L6,1)=6;
s7(1:L7,1)=7;          s8(1:L8,1)=8;          s9(1:L9,1)=9;
s10(1:L10,1)=10;       s11(1:L11,1)=11;       s12(1:L12,1)=12;
s13(1:L13,1)=13;       s14(1:L14,1)=14;       s15(1:L15,1)=15;
s16(1:L16,1)=16;       s17(1:L17,1)=17;       s18(1:L18,1)=18;
s19(1:L19,1)=19;       s20(1:L20,1)=20;       s21(1:L21,1)=21;

```

❖ **Création de la colonne activité correspondante**

```

t1=a_1(:,1);           t2=a_2(:,1);           t3=a_3(:,1);
t4=a_4(:,1);           t5=a_5(:,1);           t6=a_6(:,1);
t7=a_7(:,1);           t8=a_8(:,1);           t9=a_9(:,1);
t10=a_10(:,1);         t11=a_11(:,1);         t12=a_12(:,1);
t13=a_13(:,1);         t14=a_14(:,1);         t15=a_15(:,1);
t16=a_16(:,1);         t17=a_17(:,1);         t18=a_18(:,1);
t19=a_19(:,1);         t20=a_20(:,1);         t21=a_21(:,1);

```

❖ Concaténation de tous les x des sujets

```
x=[];
y=[];
z=[];
x=[x_1;x_2;x_3;x_4;x_5;x_6;x_7;x_8;x_9;x_10;x_11;x_12;x_13;x_14;x_15;x_16;x_17;x_18;x_19;x_20;x_21];
```

❖ Concaténation de tous les y des sujets

```
y=[y_1;y_2;y_3;y_4;y_5;y_6;y_7;y_8;y_9;y_10;y_11;y_12;y_13;y_14;y_15;y_16;y_17;y_18;y_19;y_20;y_21];
```

❖ Concaténation de tous les z des sujets

```
z=[z_1;z_2;z_3;z_4;z_5;z_6;z_7;z_8;z_9;z_10;z_11;z_12;z_13;z_14;z_15;z_16;z_17;z_18;z_19;z_20;z_21];
```

❖ Concaténation de toutes les activités des sujets

```
t=[t1;t2;t3;t4;t5;t6;t7;t8;t9;t10;t11;t12;t13;t14;t15;t16;t17;t18;t19;t20;t21];
```

❖ Concaténation de tous les sujets

```
s=[s1;s2;s3;s4;s5;s6;s7;s8;s9;s10;s11;s12;s13;s14;s15;s16;s17;s18;s19;s20;s21];
dlmwrite(strcat('total_acc_x.txt'),x,'delimiter',' ');
dlmwrite(strcat('total_acc_y.txt'),y,'delimiter',' ');
dlmwrite(strcat('total_acc_z.txt'),z,'delimiter',' ');
dlmwrite(strcat('labels.txt'),t);
dlmwrite(strcat('subject.txt'),s);
```

3.6. Script de préparation et de traitement (Phase 3) :

❖ Chargement accélération -sujets – activités

```
function Data_prep()
% acceleration data
ax1 = load('data/total_acc_x.txt');
ay1 = load('data/total_acc_y.txt');
az1 = load('data/total_acc_z.txt');
actnames = {'Marcher','monter les escaliers', 'Descendre les
escaliers','Assis','Debout','Allongé'}; %#ok<NASGU>
% Étiquettes - sujet et activité id -
subid = load('data/subject.txt');
actid = load('data/labels.txt');
```

❖ Réduction du bruit par filtre médian et Butterworth low-passe

```

ax = [];
ay = [];
az = [];
ax2 = [];
ay2 = [];
az2 = [];
lp = Lowp;
for n = 1:size(ax1,1)
    ax2(n,:) = medfilt1(ax1(n,:),3);
    ay2(n,:) = medfilt1(ay1(n,:),3);
    az2(n,:) = medfilt1(az1(n,:),3);
    ax(n,:) = filter(lp,ax2(n,:));
    ay(n,:) = filter(lp,ay2(n,:));
    az(n,:) = filter(lp,az2(n,:));
end

```

❖ Enregistrement de Fréquence d'échantillonnage (Hz) et vecteur de temps

```

fs = 50;
T = 1/fs;
t = T * (0:size(ax,2)-1); %#ok<NASGU>

```

❖ Création des fichiers Accelerations.mat, AccBySubject.mat

```

save('prep/Accelerations.mat','ax','ay','az','subid','actid','actnames',
'fs','t')
fprintf('Done.\n')

% choisir un sujet -- exemple sujet
for ks = 1:length(subid)

    subject = subid(ks);
    if subject == 9        % données du sujet

        subidx = subid==subject;
        tx1 = ax(subidx,:);
        ty1 = ay(subidx,:);
        tz1 = az(subidx,:);
        tx = reshape(tx1(:, :)', [], 1); %#ok<NASGU>
        ty = reshape(ty1(:, :)', [], 1); %#ok<NASGU>
        tz = reshape(tz1(:, :)', [], 1); %#ok<NASGU>
        aid =
repelem(actid(subidx,:), 50*ones(numel(actid(subidx)), 1)); %#ok<NASGU>
        fprintf('Done.\n')
    end
end
% Enregistrer dans fichier MAT -- AccBySubject.mat
save('prep/AccBySubject.mat','tx','ty','tz','aid','fs','actnames');
fprintf('Done.\n')

```

3.7. Visualisation graphique des données (Plot) :

3.7.1. Plot de x et y (Script)

```
%% Ouvrir un seul sujet 1
load('prep\AccBySubject.mat')
accx = tx;
accy = ty;
accz = tz;
t1 = (1/fs) * (0:size(accx)-1);
t = vec2mat(t1,1);
clear t1
% plot x et y
figure
subplot(2,1,1);
plot(t,accx)
subplot(2,1,2);
plot(t,accy)
```

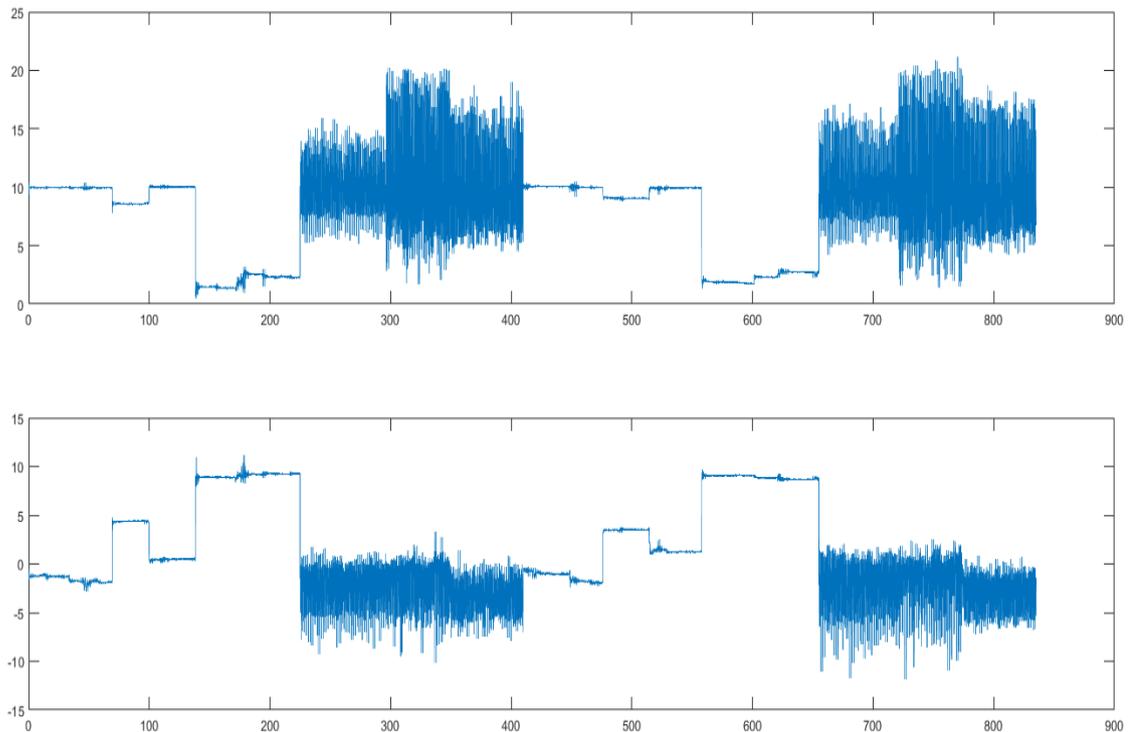


Figure 6 : Plot de x et y

3.7.2. Plot de l'accélération x par activités colorée

```
%%  
% plot x acceleration par activités colorée  
plotAccelerationColor(t, accx, aid, { 'acceleration X' })
```

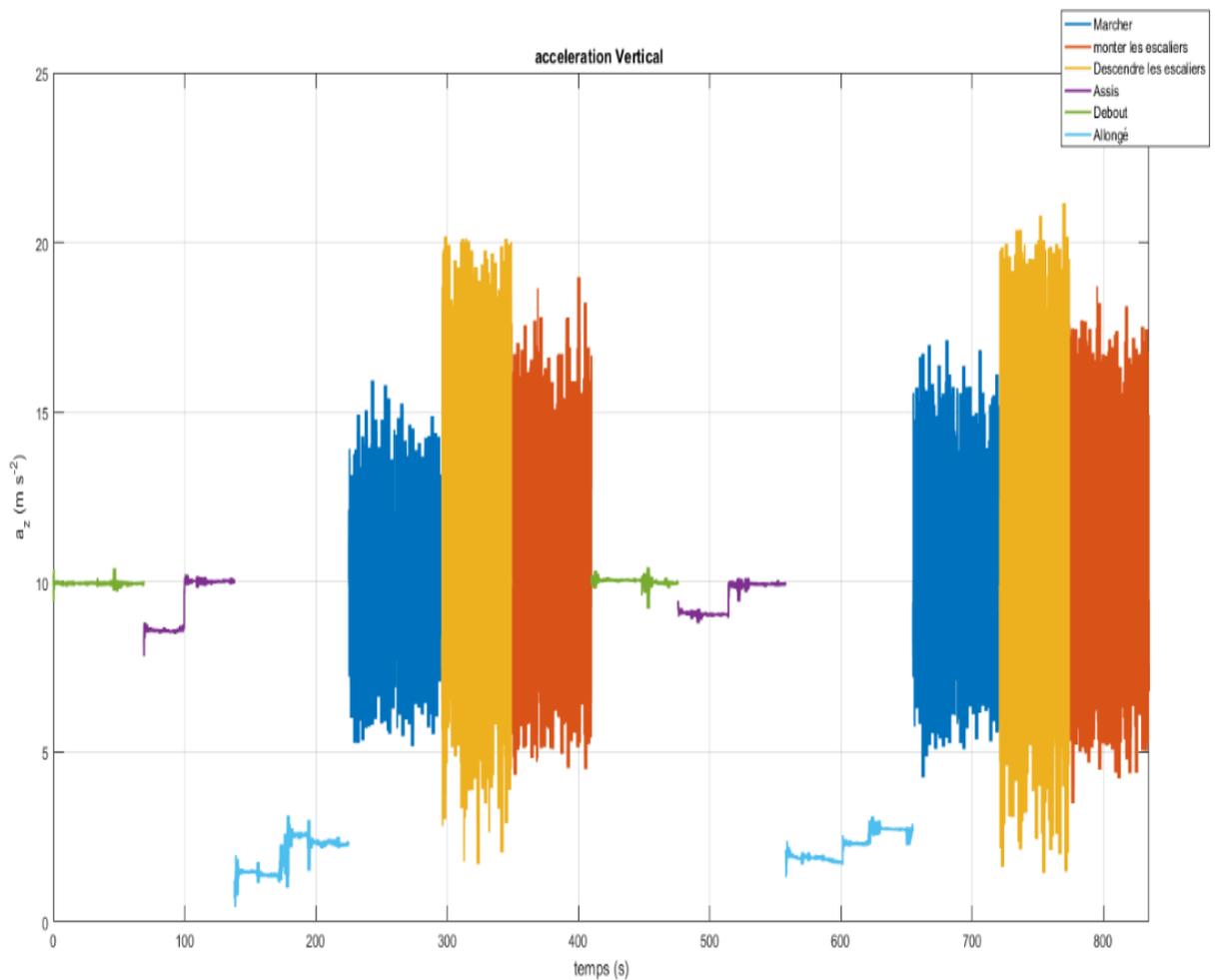


Figure 7 : Plot de l'accélération x par activités colorée

3.7.3. Plot de l'accélération x - amplitude (marche –allongé)

```

%% Utilisation du mean mesure
% Cela peut facilement aider à séparer par exemple. Marcher et
% Allongé
figure
plotHistogramComp(acy, aid, 'Marcher', 'Allongé')

```

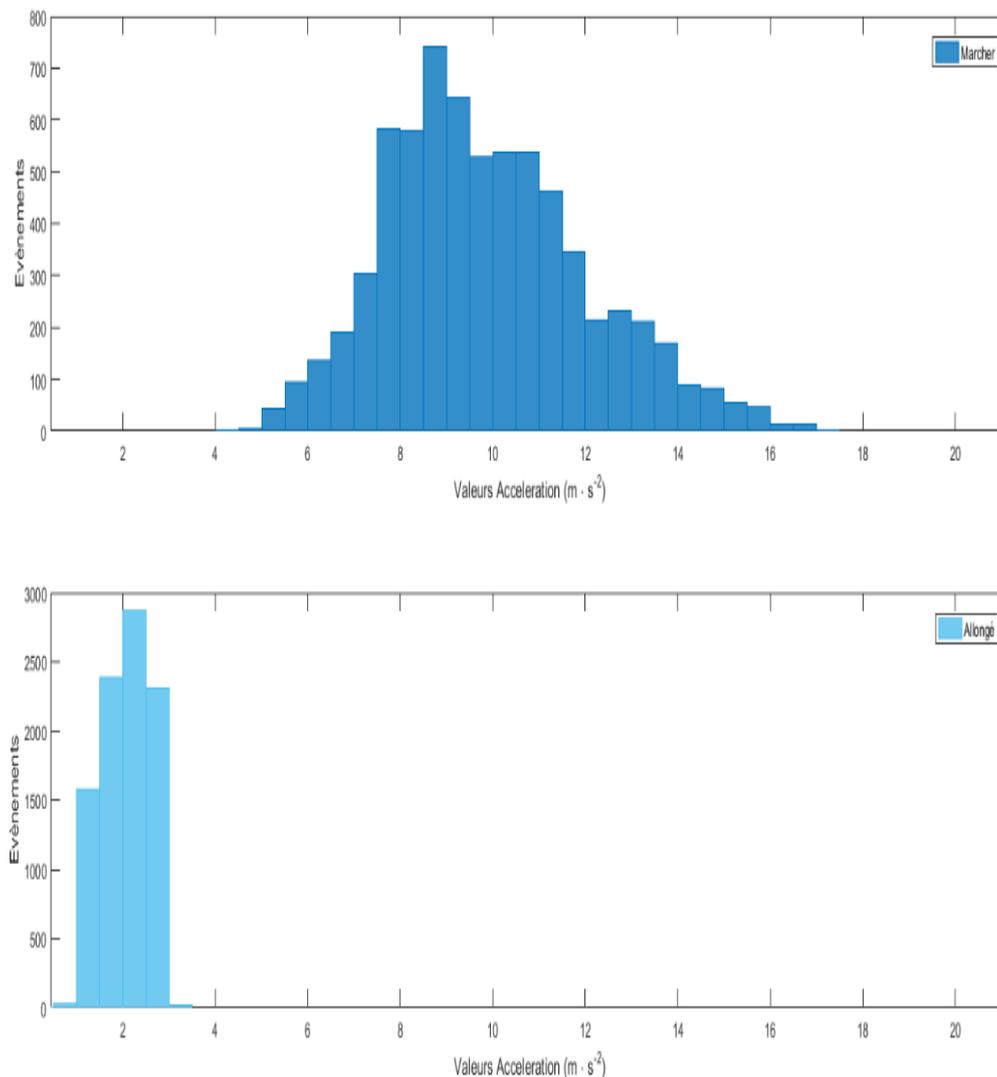


Figure 8 : Plot de l'accélération x - amplitude (marche –allongé)

Remarque : Nous pouvons constater que les méthodes d'amplitude dans le cas (marcher et Allongé) sont suffisantes, puisque nous pouvons facilement distinguer entre les deux activités, comme le montre le plot précédent.

3.7.4. Plot de l'accélération x - amplitude (marcher –debout)

```
%% Utilisation d'une mesure RMS
% Cela peut aider à séparer des activités comme, par xemple, Marche
% et debout
plotHistogramComp(accy, aid, 'Marcher', 'Debout')
```

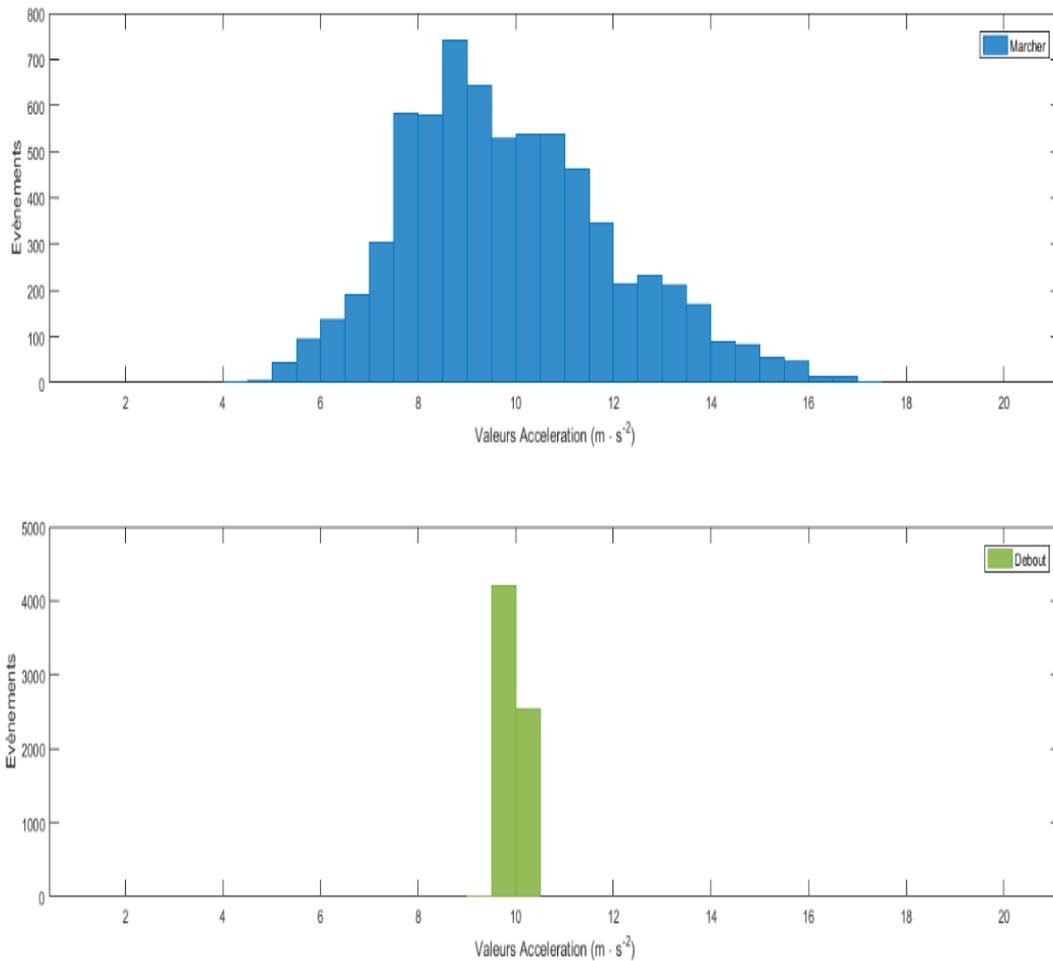


Figure 9 : Plot de l'accélération x - amplitude (marcher –debout)

Remarque : Comme le montre le plot précédent, les méthodes d'amplitude dans le cas (marcher et debout) sont suffisantes, puisque nous pouvons distinguer entre les deux activités.

3.7.5. Plot de l'accélération x - amplitude (marcher –monter l'escalier)

```
% Marcher et Monter les escaliers
figure
plotHistogramComp(accy, aid, 'Marcher', 'Monter les escaliers')
```

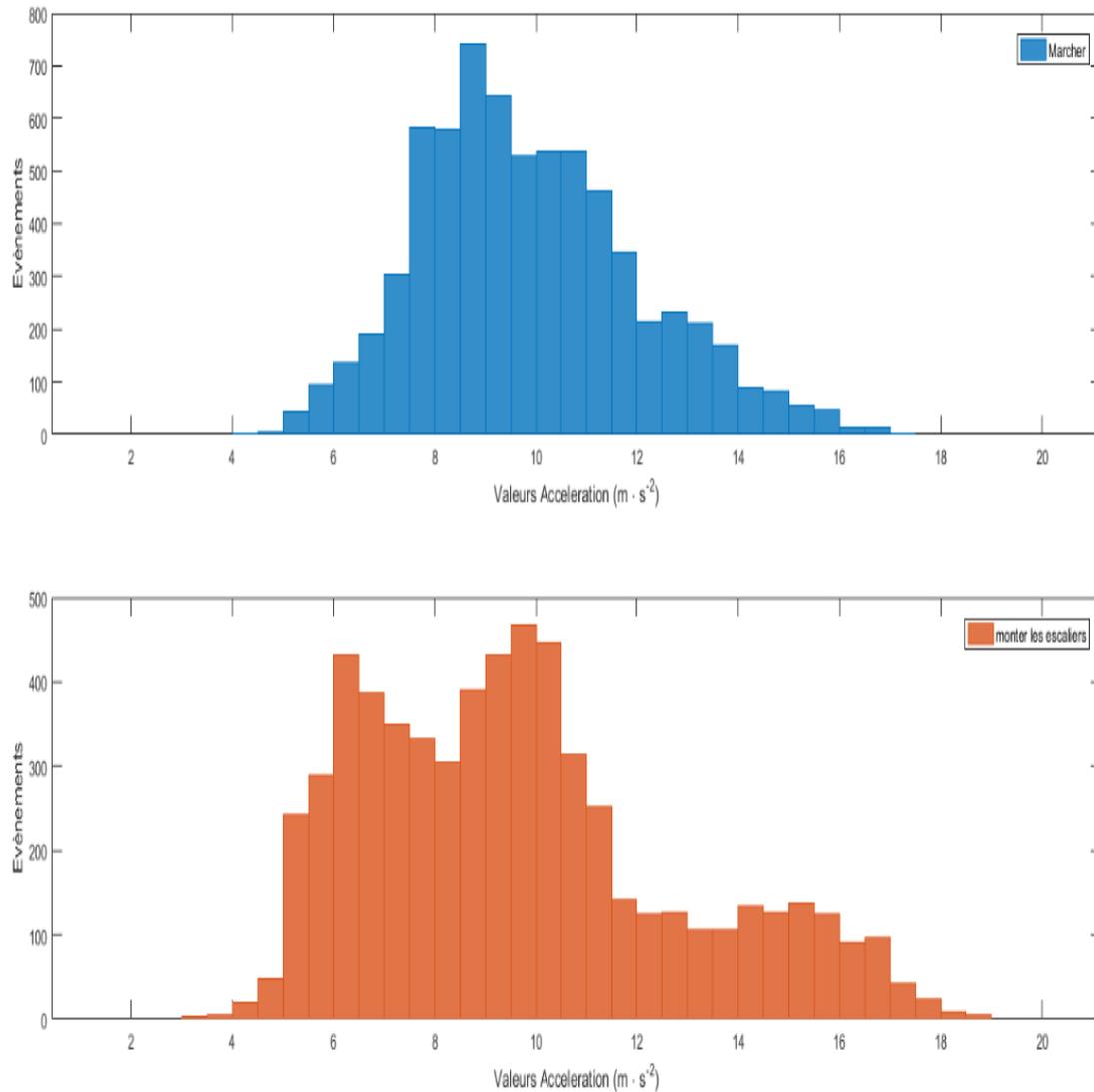


Figure 10 : Plot de l'accélération x - amplitude (marcher –monter l'escalier)

Remarque : Les méthodes d'amplitude uniquement ne suffisent pas, comme nous pouvons le constater dans le plot précédant (marche et monte-escalier) puisque les deux moments statistiques sont presque similaires.

Une première conclusion est que l'analyse statistique simple n'est pas suffisante, alors nous devons également envisager des méthodes qui mesurent les variations de signal au fil du temps

3.7.6. Plot après filtrage numérique

```

%% filtrage numérique
% Filtrer l'accélération gravitationnelle

% filter high pass
fda = fdafilter;

% L'accélération «découplée» due à la dynamique du corps par gravité
accf = filter(fda,accx);

% Comparez le signal filtré avec l'originale
plotAccelerationColor(t, [accx, accf], aid,{'Originale','High-pass
filtre'})

```

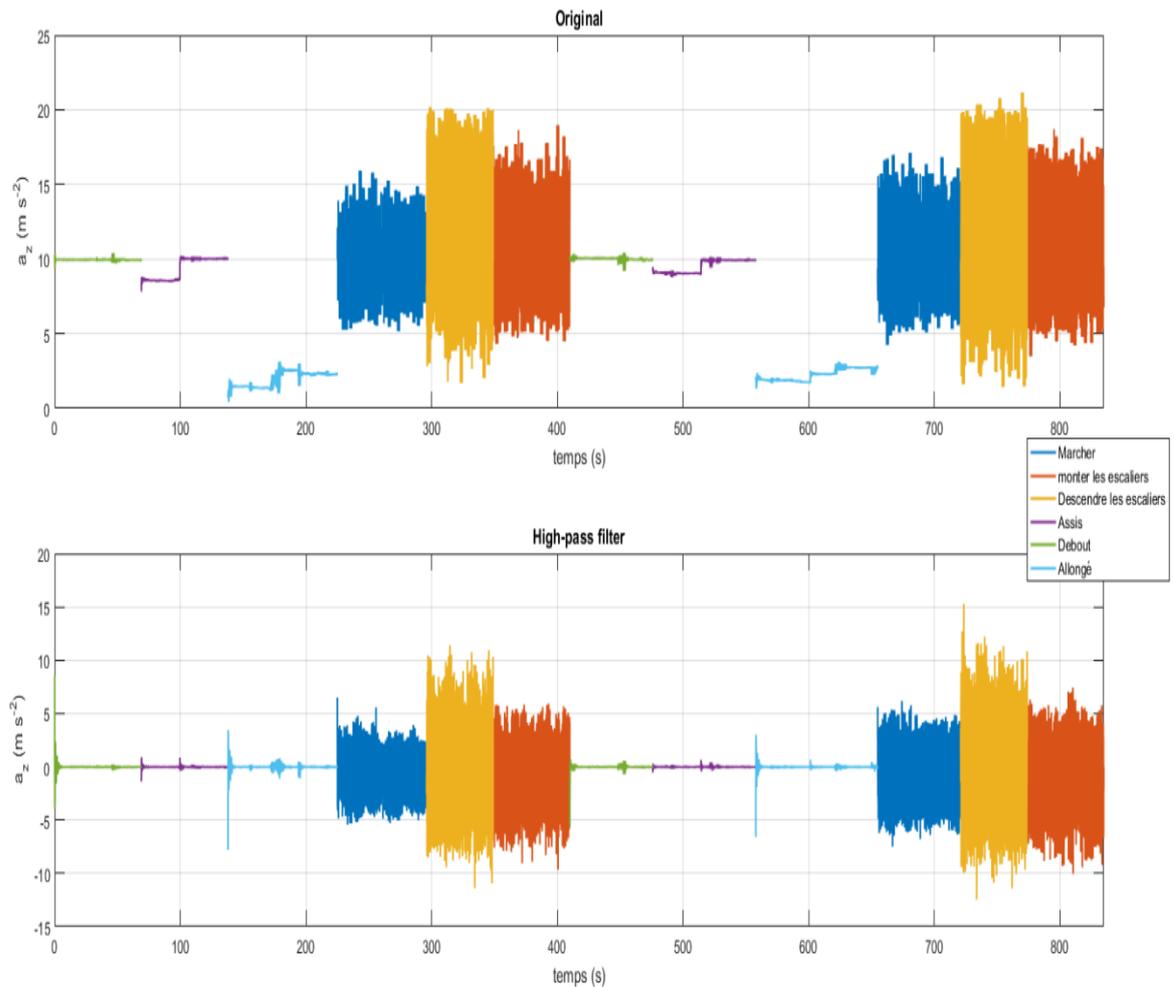


Figure 11 : Plot après filtrage numérique

3.7.7. Plot de Périodogramme

```

%%Sélectionnez et filtrer (high pass) la première partie du signal marcher
selw = (t < 18) & (aid == 1); % Marcher = 1
tw = t(selw);
accfw = accf(selw);
%% Sélectionnez et filtrer (high pass) la première partie du signal monter
%les escaliers
selwu = (t < 70) & (aid == 2); % Monter les escaliers = 2
twu = t(selwu);
accfwu = accf(selwu);
%% Plot periodogram Power Spectral Density
figure
subplot(2,1,1);
periodogram(accfw, [], 4096, fs); % Marcher
subplot(2,1,2);
periodogram(accfwu, [], 4096, fs); % Monter les escaliers

```

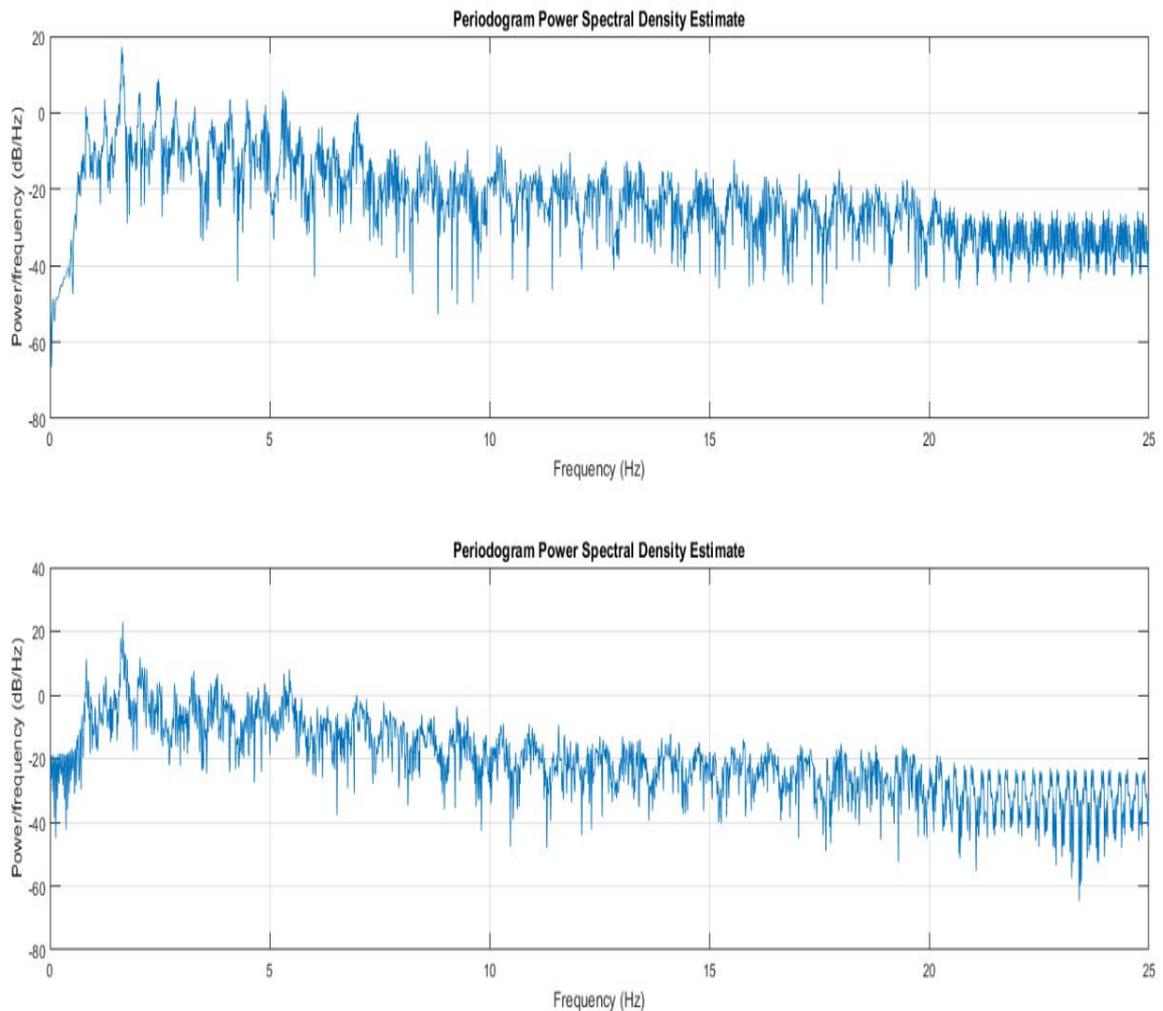


Figure 12 : Plot de Périodogramme

4. Extraction des caractéristiques

Les changements des valeurs de l'accéléromètre le long de chaque axe du Smartphone sont surveillés. Une longueur de fenêtre fixe de 50 données pour le domaine temporel est utilisée dans l'étape d'extraction de caractéristiques.

4.1. Caractéristiques du domaine temporel (Time-domain Features) [26] :

Pour chaque fenêtre, certains caractères statistiques de base peuvent être calculés à partir de données de capteur brut.

Mean. La valeur moyenne de chaque fenêtre le long de chaque axe.

Pearson Correlation coefficient. Le coefficient de corrélation est calculé le long de chaque paire d'axes des données du capteur. Mesure la relation entre deux vecteurs de signal s_1 et s_2 . Cette mesure statistique est bornée entre l'intervalle $[-1, 1]$ et indique une corrélation positive ou négative lorsque cette valeur est proche de 1 ou -1 respectivement, et aucune corrélation n'est proche de 0. Nous appliquons cette opération à des Signaux temporels de pair d'axe à partir de l'accéléromètre.

Root Mean Square (RMS): La moyenne quadratique d'un ensemble de nombres est la racine carrée de la moyenne arithmétique des carrés de ces nombres.

4.2. Caractéristiques du domaine fréquentiel (Frequency-domain Features)

Les caractéristiques du domaine fréquentiel ont été largement utilisées pour saisir la nature périodique des signaux. Un algorithme couramment utilisé pour extraire des fonctions de domaine fréquentiel est la transformée de Fourier rapide (FFT). Les caractéristiques typiques du domaine fréquentiel :

Periodogram: permet une estimation simple de la densité spectrale de puissance en prenant le carré de la transformée de Fourier, "**Spectral power of a frequency band**" : avec (periodogram PSD estimate) Cette fonction renvoie une mesure de puissance dans un intervalle du signal de fréquence. Puissance totale dans 6 bandes de fréquences adjacentes et prédéfinies.

Le tableau suivant (**Tableau 3**) contient les fonctions que nous avons utilisées afin de réussir l'extraction des caractéristiques de notre dataset.

Fonction	Déscription	Formulation
Mean(s)	Arithmetic mean	$\bar{s} = \frac{1}{N} \sum_{i=1}^N s_i$
RMS(s)	Root Mean Square	$\delta = \sqrt{\frac{1}{N} \sum_{i=1}^N s_i^2}$
Std (s)	Standard deviation	$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N s_i^2}$
Correlation (s1, s2)	Pearson correlation coefficient	$\rho = \frac{cov(s1, s2)}{\sigma(s1)\sigma(s2)}$
Periodogram (s, [],[],fs)	power spectral density (PSD) estimate	$\hat{S}_{per}(\omega) = \frac{1}{N} \left \sum_{n=1}^N x(n)exp(-j\omega n) \right ^2$ <p>N représente le nombre d'échantillons fixés ω représente la fréquence et varie</p>

** (N : longueur du vecteur signal)

Tableau 3 : Fonctions d'extractions

- **Caractéristiques obtenus** : Nous avons obtenu comme résultat 73 Caractéristiques, que nous allons détailler dans le **tableau 4** :

Nom	Mean	RMS	Correlation	power spectral (5 chaqu'un)
Total acc (x,y,z)	3	3	3	0
Body acc (x,y,z)	3	3	3	3x5=15
Body acc jerk (x,y,z)	3	3	3	3x5=15
Total acc magnitude	1	1	0	0
Body acc magnitude	1	1	0	5
Jerk acc magnitude	1	1	0	5
	12	12	9	40
Total :	73			

Tableau 4 : Caractéristiques obtenus

5. Sélection des caractéristiques (Features Selection) :

Dans notre expérimentation, les valeurs moyennes et corrélation dans le domaine du temps est une mesure de puissance dans un interval du signal de fréquence dans le domaine de la fréquence sont extraites. Certaines analyses de sensibilité ont été faites et ces caractéristiques sont jugées plus sensibles et utiles.

Les caractéristiques peuvent être sélectionnées à travers des méthodes de sélection, ces méthodes choisissent des fonctionnalités basées sur des critères de discrimination qui sont relativement indépendants de la classification.

Plusieurs méthodes utilisent des coefficients de corrélation simples semblables au critères discriminant de Fisher. D'autres adoptent des informations mutuelles ou des tests statistiques (test t, test F). Dans notre cas nous avons utilisé une métrique comme **Paired t test** [27] qui identifie les meilleurs caractéristiques qui aident à séparer les classes comme le montre la **figure 11**, tandis que les fonctions redondantes sont supprimées. De plus, ces caractéristiques sont relativement faciles et efficaces à calculer et à extraire.

Il y a 3 fonctions de domaine temporel et 1 fonction de domaine de fréquence pour chaque axe et 56 caractéristiques parmi 73 ont été choisies pour décrire chaque fenêtre d'activité et nous avons éliminé les fonctions suivantes :

Mean(Body acc (x,y,z))
 Mean(Body acc jerk (x,y,z))
 RMS(Body acc magnitude)
 power spectral(Body acc magnitude)
 power spectral(Jerk acc magnitude)

Nom	Mean	RMS	Correlation	power spectral (6 chaqu'un)
Total acc (x,y,z)	3	3	3	0
Body acc (x,y,z)	0	3	3	3x5=15
Body acc jerk (x,y,z)	0	3	3	3x5=15
Total acc magnitude	1	1	0	0
Body acc magnitude	1	0	0	0
Jerk acc magnitude	1	1	0	0
Sous -totaux	6	11	9	30
Total :	&	56		

Tableau 5 : Caractéristiques retenus après sélection

6. Classification :

Un certain nombre de classificateurs k-Nearest Neighbor (kNN), Support Vector Machines (SVM), Neural Networks (NN), Decision Tree (DT)) qui ont été examinés au premier chapitre sont étudiés sur les données collectées. Les fonctionnalités extraites des données brutes peuvent être des fonctions de domaine temporel ou de domaine fréquentiel.

Pour évaluer l'efficacité de différents classificateurs, la matrice de confusion est utilisée comme méthode d'évaluation.

Les paramètres de performance sélectionnés pour évaluer les méthodes de classification sont la précision et le rappel. La précision (P) est définie comme étant le nombre de vrais positifs (Tp) sur le nombre de vrais positifs et le nombre de faux positifs (Fp) :

$$P = \frac{T_p}{T_p + F_p} \quad \dots\dots\dots (1)$$

Le rappel (R) est défini comme le nombre de vrais positifs (Tp) sur le nombre de vrais positifs et le nombre de faux négatifs (Fn):

$$R = \frac{T_p}{T_p + F_n} \quad \dots\dots\dots (2)$$

La précision est également utilisée pour comparer les résultats de classification, qui est définie comme :

$$Accuracy = \frac{T_p + T_n}{T_p + T_n + F_p + F_n} \quad \dots\dots\dots (3)$$

Dataset a été réparti au hasard en deux ensembles indépendants, où 70% des données ont été sélectionnés pour training et les 30% restants pour les tests.

Voici le script de préparation du dataset des caractéristiques (dataset features) pour la répartition, afin d'entamer la classification :

❖ Script de de préparation du dataset des caractéristiques :

```

%% Effacer toutes les variables et charger les variables pré-
%enregistrées
% Effacer les variables non pertinentes
clear; clc

% Charger l'ensemble de vecteurs de caractéristiques (feat) et
% l'ensemble de cellules de noms de caractéristiques (featlabels)

load('Prep\Features.mat')

% Charger signales
load('Prep\Accelerations.mat')

% Data preparation
Activities = categorical(actid, (1:numel(actnames)), actnames);

Dataset_train = [array2table(feat) table(Activities)];

Dataset_train.Properties.VariableNames = [featlabels 'Activities'];

```

6.1. Résultat de classification :

DECISION TREES		
Preset	Précision (%)	Option
Complex Tree	90,6	Maximum number of splits: 100 Split criterion: Gini's diversity index Surrogate decision splits: Off
Medium Tree	85,2	Maximum number of splits: 20 Split criterion: Gini's diversity index Surrogate decision splits: Off
Simple Tree	75,4	Maximum number of splits: 4 Split criterion: Gini's diversity index Surrogate decision splits: Off

Tableau 6 : Précision DECISION TREES

NEAREST NEIGHBOR Classifier		
Preset	Précision (%)	Option
Fine KNN	91,4	Number of neighbors: 1 Distance metric: Euclidean Distance weight: Equal Standardize data: true
Medium KNN	90.9	Number of neighbors: 10 Distance metric: Euclidean Distance weight: Equal Standardize data: true
Coarse KNN	87.1	Number of neighbors: 100 Distance metric: Euclidean Distance weight: Equal Standardize data: true
Cosine KNN	91.9	Number of neighbors: 10 Distance metric: Cosine Distance weight: Equal Standardize data: true
Cubic KNN	90.1	Number of neighbors: 10 Distance metric: Minkowski (cubic) Distance weight: Equal Standardize data: true
Weighted KNN	92.0	Number of neighbors: 10 Distance metric: Euclidean Distance weight: Squared inverse Standardize data: true

Tableau 7 : Précision NEAREST NEIGHBOR Classifier

NEURAL NETWORK	
Précision (%)	Option
91.3	Number of Hidden Neurons : 18

Tableau 8 : Précision NEURAL NETWORK

SUPPORT VECTOR MACHINE		
Preset	Précision (%)	Option
Linear SVM	92,3	Kernel function: Linear Kernel scale: Automatic Box constraint level: 1 Multiclass method: One-vs-One Standardize data: true
Quadratic SVM	95,9	Kernel function: Quadratic Kernel scale: Automatic Box constraint level: 1 Multiclass method: One-vs-One Standardize data: true
Cubic SVM	96,3	Kernel function: Cubic Kernel scale: Automatic Box constraint level: 1 Multiclass method: One-vs-One Standardize data: true
	96,3	Box constraint level: 2
	96.4	Box constraint level: 3
Fine Gaussian SVM	87,8	Kernel function: Gaussian Kernel scale: 1.7 Box constraint level: 1 Multiclass method: One-vs-One Standardize data: true
Medium Gaussian SVM	94.5	Kernel function: Gaussian Kernel scale: 6.6 Box constraint level: 1 Multiclass method: One-vs-One Standardize data: true
Coarse Gaussian SVM	90.7	Kernel function: Gaussian Kernel scale: 27 Box constraint level: 1 Multiclass method: One-vs-One Standardize data: true

Tableau 9 : Précision SUPPORT VECTOR MACHINE

6.2. Evaluation des performances :

6.2.1. Matrice de confusion :

La méthode la plus commune est la *matrice de confusion* [28] qui laisse représenter l'exécution d'algorithme en identifiant clairement les types des erreurs (les faux positifs et les négatifs).

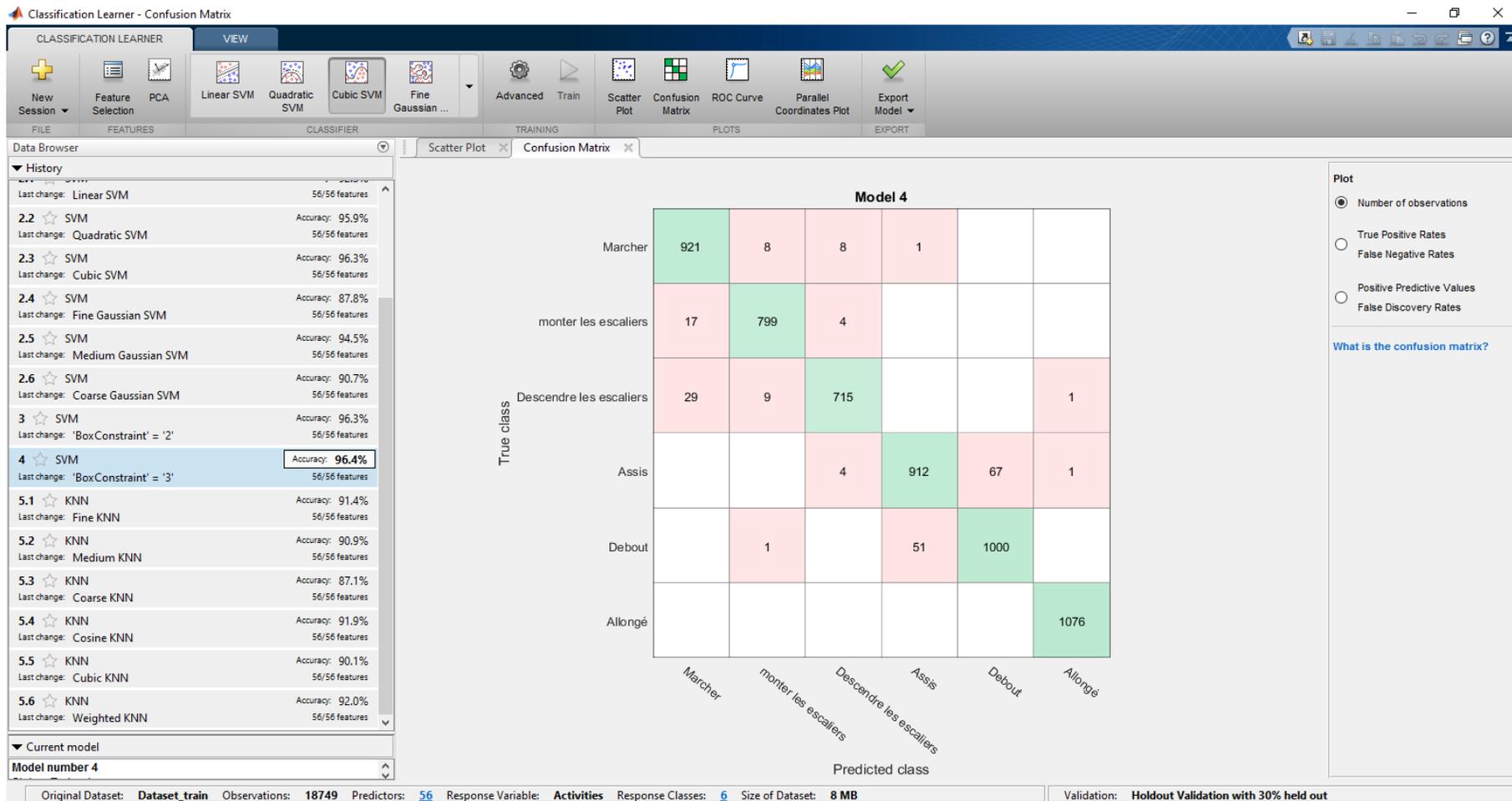


Figure 14 : Matrice de Confusion TC/PC

- Nous pouvons constater que quelque activité sont mal interprétées comme : monter l'escalier → marcher (17 cas sur 799) ou descendre l'escalier → marcher (29 cas sur 715). Il y a aussi une confusion entre les deux activités : assis-debout (67 cas sur 912 et 51 cas sur 1000).

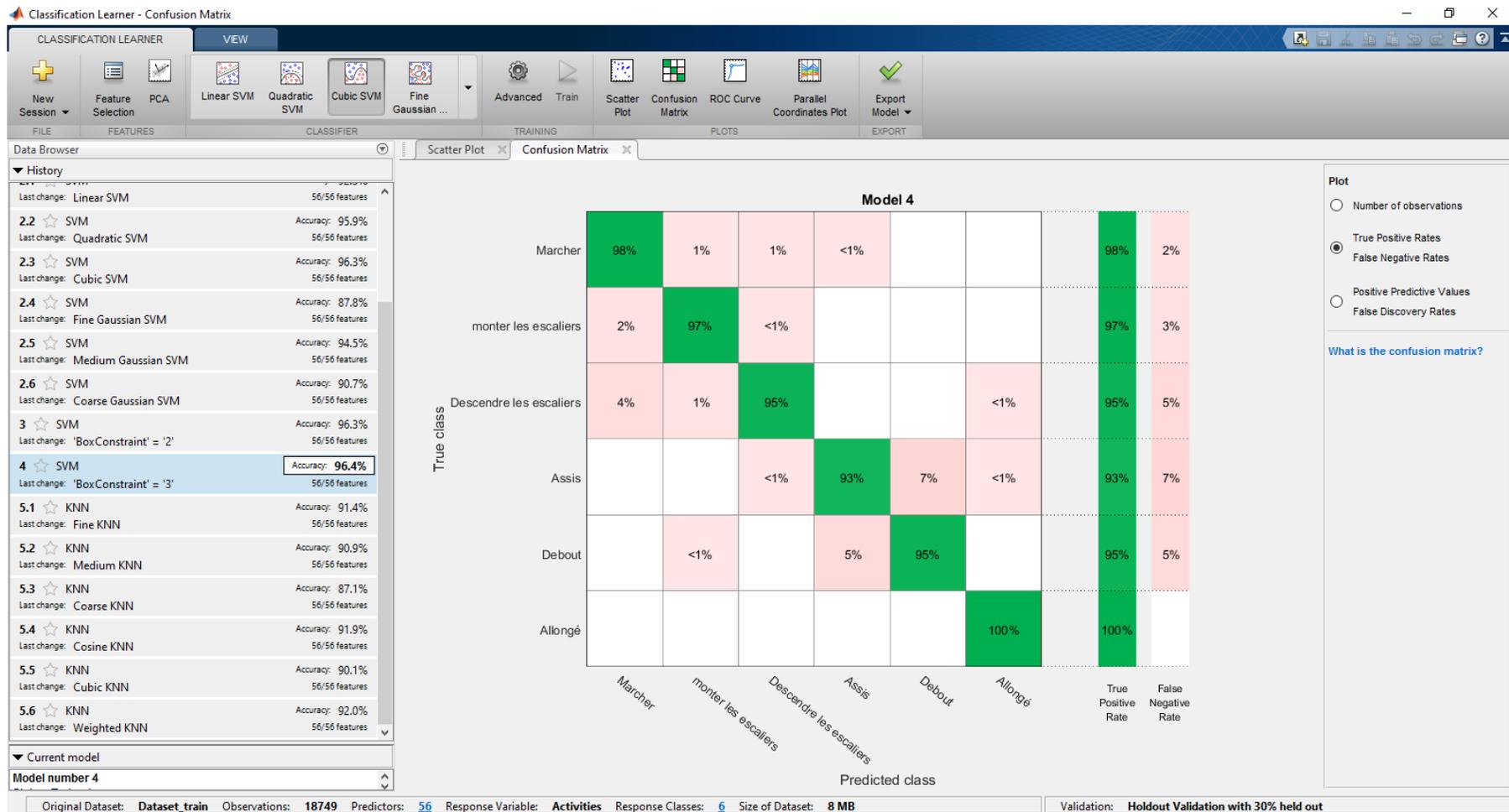


Figure 15 : Matrice de Confusion TPR/FNR

- Nous pouvons constater que le taux le plus élevé dans les faux négatifs concerne l'activité "Assis" : 7%.

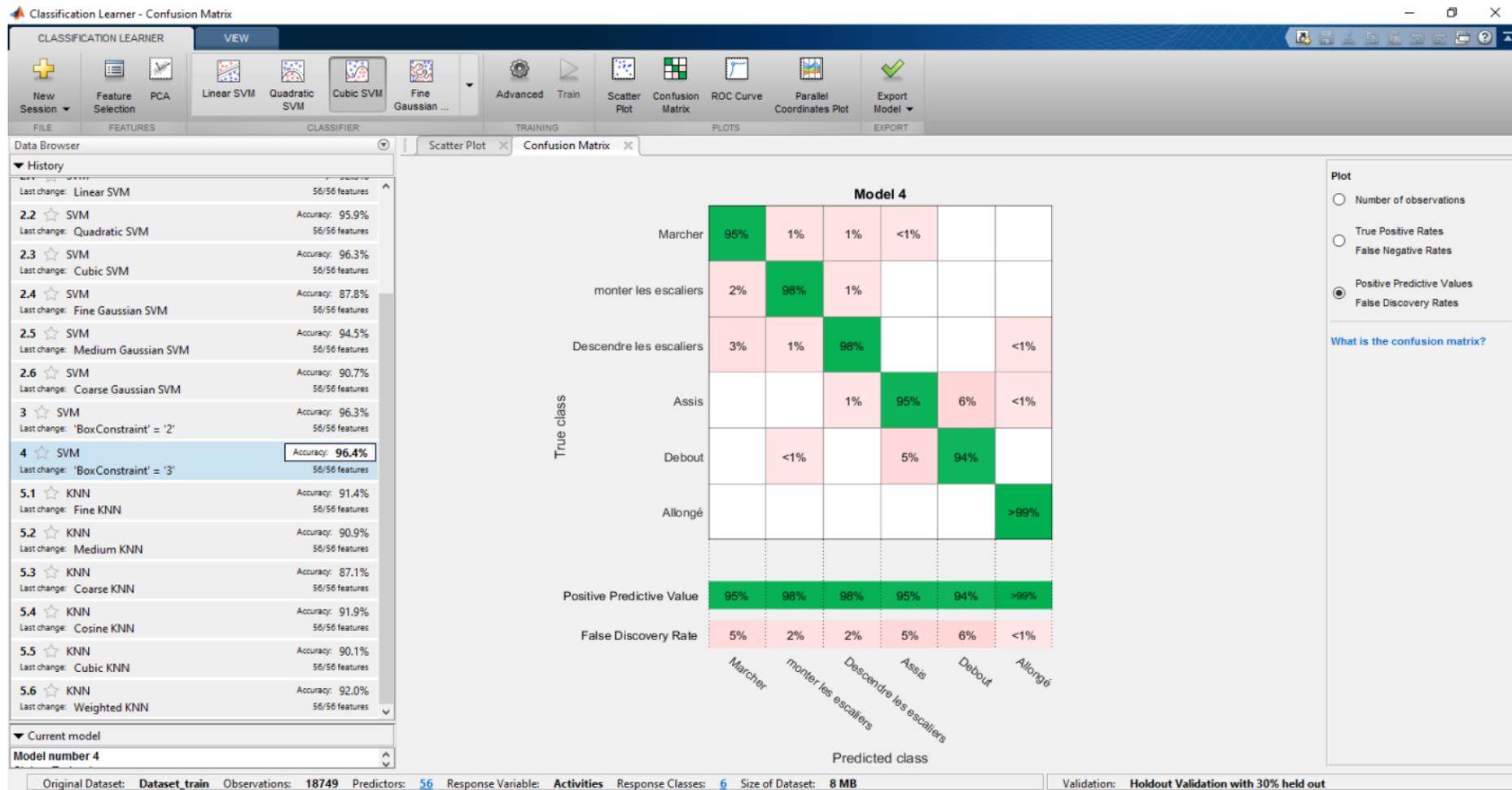


Figure 16 : Matrice de Confusion PDV/FDR

- Nous pouvons constater que le taux le plus élevé dans les faux positifs concerne l'activité "Debout" : 6 %.

Récapitulation :

1. Les activités les plus confuses entre eux sont :
 - Monter l'escalier → marcher (17 cas sur 799)
 - Descendre l'escalier → marcher (29 cas sur 715)
 - Assis → debout (67 cas sur 912).
 - debout → Assis (51 cas sur 1000).
2. Le taux le plus élevé dans les faux négatifs concerne l'activité "Assis" : 7%.
3. le taux le plus élevé dans les faux positifs concerne l'activité "Debout" : 6 %.
4. Les 56 caractéristiques sélectionnées sont :
 - Total acc (x,y,z) : Mean 3, RMS 3, Correlation 3
 - Body acc (x,y,z) : RMS 3, Correlation 3, power spectral 15
 - Body acc jerk (x,y,z) : RMS 3, Correlation 3, power spectral 15
 - Total acc magnitude : Mean 1, RMS 1
 - Body acc magnitude : Mean 1
 - Jerk acc magnitude : Mean 1, RMS 1
5. Suite aux résultats de performance : SVM est la méthode la plus adéquate dans notre cas avec une **Précision de 96.4%**.

SVM / Kernel function: Cubic / Kernel scale: Automatic / Box constraint level: 3

Multiclass method: One-vs-One / Standardize data: true / Standardize data: true

6.2.2. Test avec un Dataset certifié :

En plus de la validation des performances avec la matrice de confusion que nous avons terminée précédemment, nous allons tester notre sélection de caractéristiques avec un dataset certifié et publié.

a) Présentation du dataset :

▪ Publication du dataset

L'ensemble de données de HAR a été rendu disponible pour l'usage public. Il se compose de données brutes à inertie d'une expérience. Les détails de l'ensemble de données sont montrés dans la référence suivante [29].

▪ Validation du dataset :

Classifieur	Acronym	Précision (%)
Decision tree C4.5	DT	83.06
Random forest	RF	89.34
k-Nearest neighbors	k-NN	88.19
Naive bayes	NB	78.89
Logistic regression	LR	96.40
Multilayer perceptron	MLP	94.60
Support vector machine	SVM	96.50

Tableau10 : Comparaison Publié des Performances des classificateurs ML

▪ Caractéristiques :

Au total, 561 caractéristiques ont été extraites pour décrire chaque fenêtre d'activité de taille de 128 cycles.

b) Résultats obtenus :

Les résultats que nous avons obtenus avec les 56 caractéristiques retenus dans notre expérimentation sont présentés comme suit :

DECISION TREES		
Preset	Précision (%)	Option
Complex Tree	91.5	Maximum number of splits: 100 Split criterion: Gini's diversity index Surrogate decision splits: Off
Medium Tree	87.1	Maximum number of splits: 20 Split criterion: Gini's diversity index Surrogate decision splits: Off
Simple Tree	75,0	Maximum number of splits: 4 Split criterion: Gini's diversity index Surrogate decision splits: Off

Tableau 11 : Précision DECISION TREES – 2eme Test

NEAREST NEIGHBOR Classifier		
Preset	Précision (%)	Option
Fine KNN	94.9	Number of neighbors: 1 Distance metric: Euclidean Distance weight: Equal Standardize data: true
Medium KNN	94.3	Number of neighbors: 10 Distance metric: Euclidean Distance weight: Equal Standardize data: true
Coarse KNN	89.2	Number of neighbors: 100 Distance metric: Euclidean Distance weight: Equal Standardize data: true
Cosine KNN	94.3	Number of neighbors: 10 Distance metric: Cosine Distance weight: Equal Standardize data: true
Cubic KNN	93.0	Number of neighbors: 10 Distance metric: Minkowski (cubic) Distance weight: Equal Standardize data: true
Weighted KNN	95.3	Number of neighbors: 10 Distance metric: Euclidean Distance weight: Squared inverse Standardize data: true

Tableau 12 : Précision NEAREST NEIGHBOR Classifier – 2eme Test

SUPPORT VECTOR MACHINE		
Preset	Précision (%)	Option
Linear SVM	94.2	Kernel function: Linear Kernel scale: Automatic Box constraint level: 1 Multiclass method: One-vs-One Standardize data: true
Quadratic SVM	95,6	Kernel function: Quadratic Kernel scale: Automatic Box constraint level: 1 Multiclass method: One-vs-One Standardize data: true
Cubic SVM	96,7	Kernel function: Cubic Kernel scale: Automatic Box constraint level: 1 Multiclass method: One-vs-One Standardize data: true
	96.8	Box constraint level: 2
	96.6	Box constraint level: 3
Fine Gaussian SVM	87,4	Kernel function: Gaussian Kernel scale: 1.7 Box constraint level: 1 Multiclass method: One-vs-One Standardize data: true
Medium Gaussian SVM	95.1	Kernel function: Gaussian Kernel scale: 6.6 Box constraint level: 1 Multiclass method: One-vs-One Standardize data: true
Coarse Gaussian SVM	92.0	Kernel function: Gaussian Kernel scale: 27 Box constraint level: 1 Multiclass method: One-vs-One Standardize data: true

Tableau 13 : Précision SUPPORT VECTOR MACHINE- 2eme Test

NEURAL NETWORK	
Précision (%)	Option
94.0	Number of Hidden Neurons : 18

Tableau 14 : Précision NEURAL NETWORK - 2eme Test

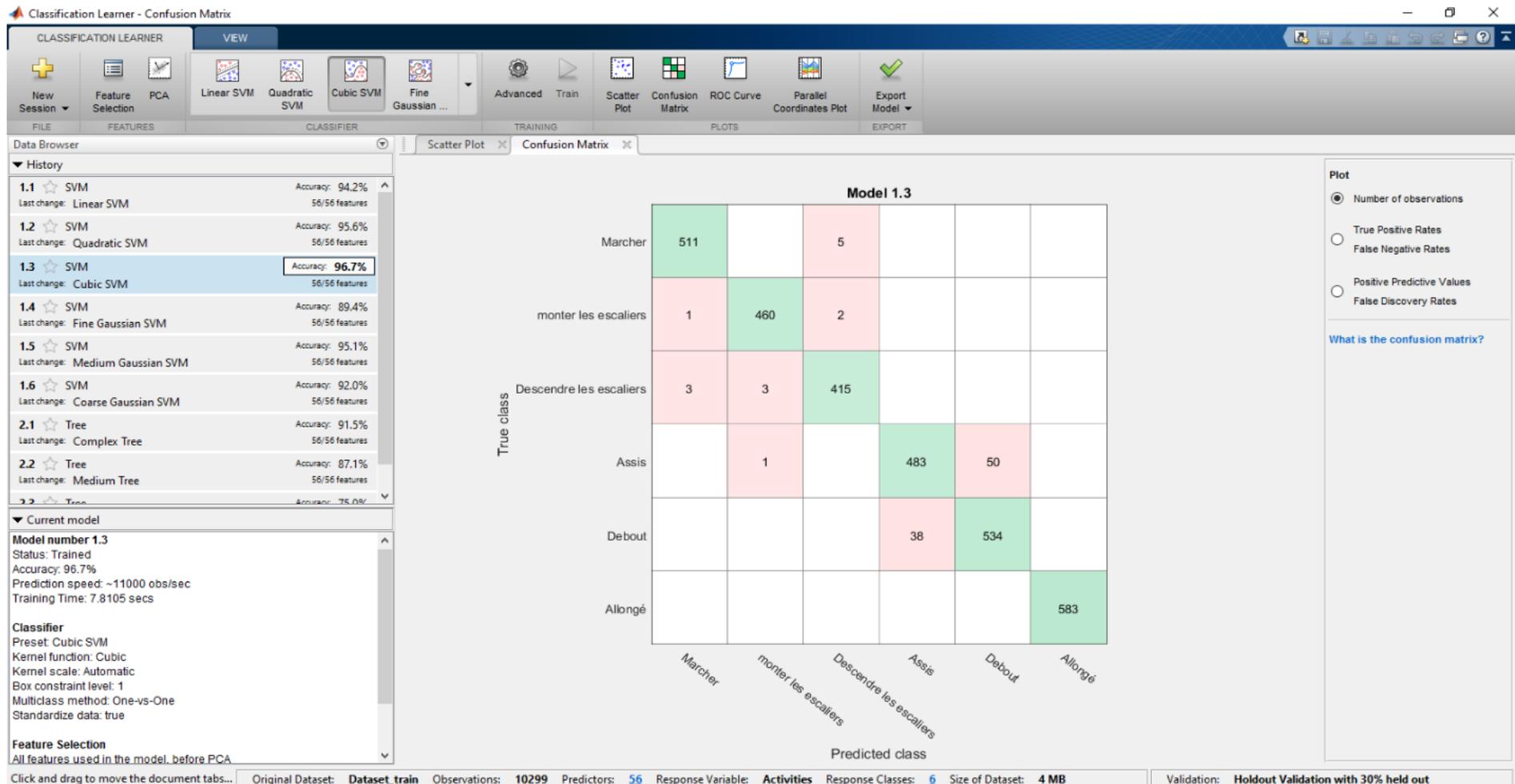


Figure 17 : Matrice de Confusion TC / PC – Zeme Test

- Nous pouvons constater qu'il y a seulement une confusion entre les deux activités : assis-debout (50 cas sur 483 et 38 cas sur 534).

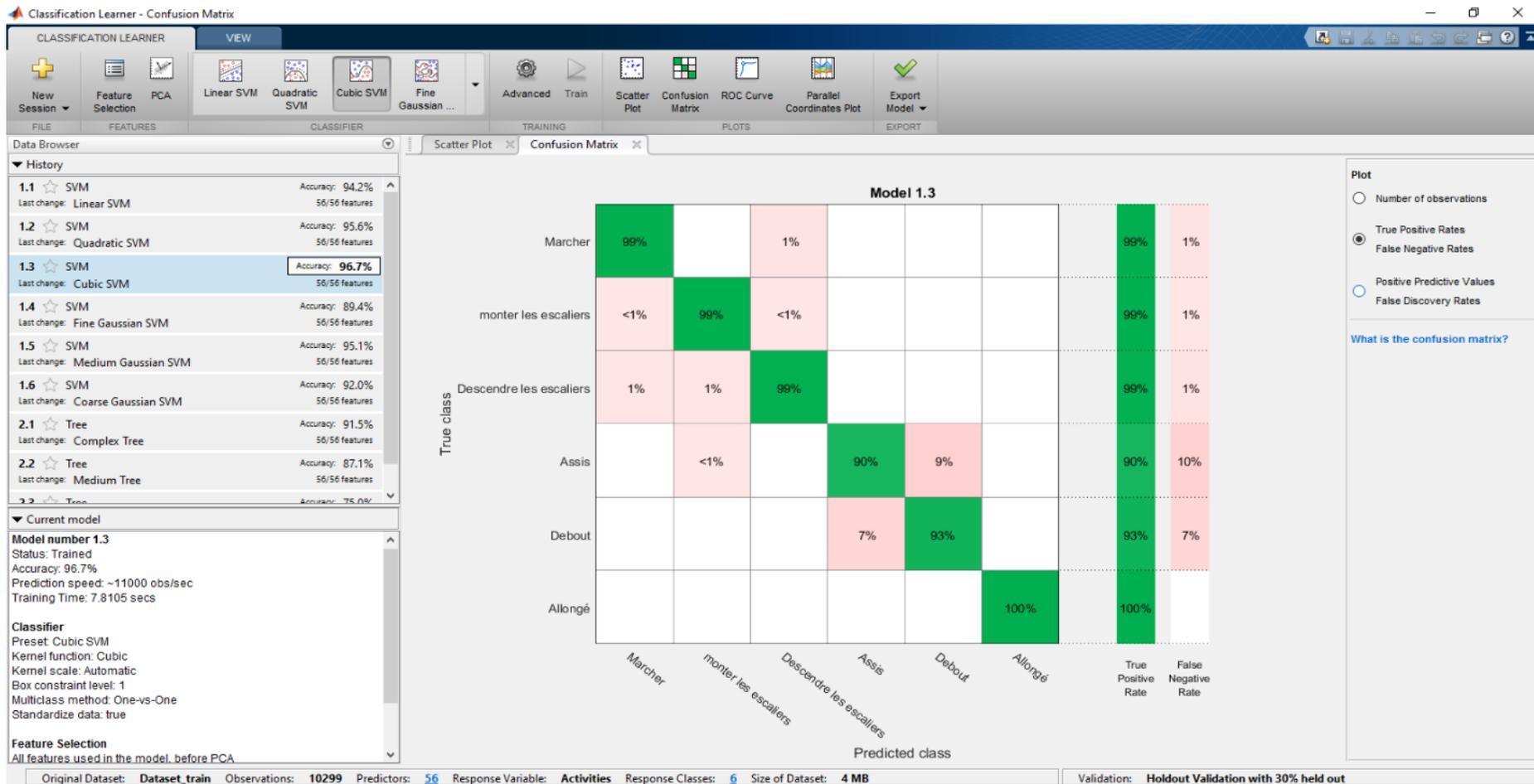


Figure 18 : Matrice de Confusion TPR/FNR – 2eme Test

- Nous pouvons constater que le taux le plus élevé dans les faux positifs concerne l'activité " Assis " : 10 %.

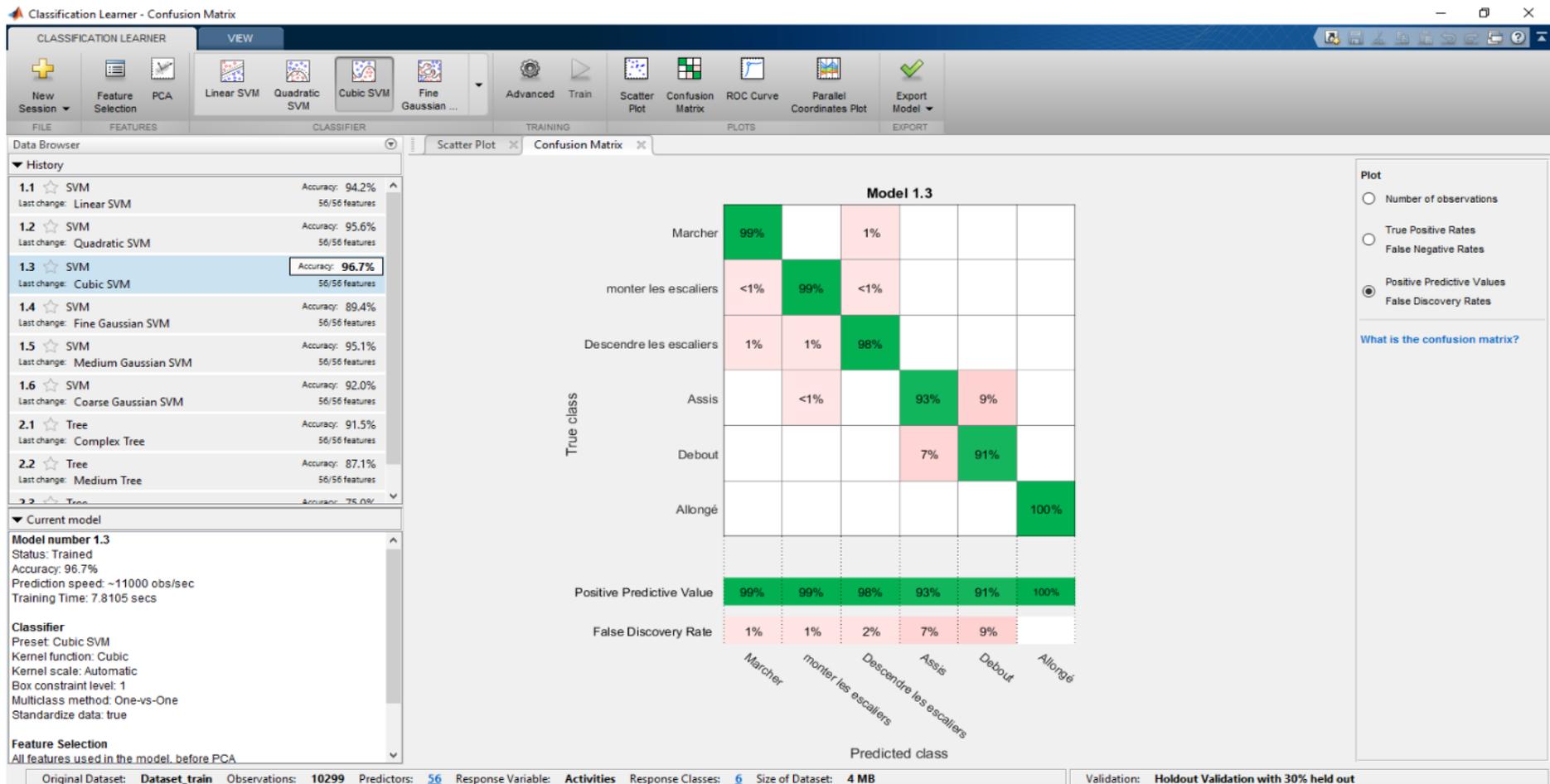


Figure 19 : Matrice de Confusion PPV/FDR – 2eme Test

- Nous pouvons constater que le taux le plus élevé dans les faux négatifs concerne l'activité "Debout" : 9%.

c) Comparaison des résultats :

- Nous allons comparer les résultats obtenus, en utilisant notre dataset, avec les résultats obtenus en utilisant le dataset certifié :

Classifieurs Utilisés	Précision (%) (Dataset certifié)	Précision (%) (Dataset "Centre A")
Decision tree	91.5	90.6
k-Nearest neighbors	95.3	92.0
Neural network	94.0	91.3
Support vector machine	96.8	96.4
Nbre Caractéristiques Utilisées	56	56
Segmentation (window)	128	50
Senseurs Utilisés	Accéléromètre	Accéléromètre

Tableau15 : Comparaison N°1 des Performances des classificateurs ML – 2eme Test

- Nous allons comparer aussi les résultats obtenus, en utilisant notre modele (Centre A), avec les résultats figurant dans l'expérimentation du dataset certifié (modele Certifié) :

Classifieurs Utilisés	Précision (%) (Modele certifié)	Précision (%) (Modele "Centre A")
Decision tree	83.06	91.5
Random forest	89.34	/
k-Nearest neighbors	88.19	95.3
Naive bayes	78.89	/
Logistic regression	96.40	/
Neural network	94.60	94.0
Support vector machine	96.50	96.8
Nbre Caractéristiques Utilisées	561	56
Segmentation (window)	128	128
Senseurs Utilisés	Accéléromètre Gyroscope	Accéléromètre /

Tableau16 : Comparaison N°2 des Performances des classificateurs ML – 2eme Test

Récapitulation :

1. Les activités les plus confuses entre eux sont :
 - Assis → debout (50 cas sur l'ensemble de 483).
 - debout → Assis (38 cas sur l'ensemble de 524).
2. Le taux le plus élevé dans les faux négatifs concerne l'activité "Assis" : 10
3. le taux le plus élevé dans les faux positifs concerne l'activité "Debout" : 9 %.
4. Suite aux résultats de performance : SVM est toujours la méthode la plus adéquate pour notre cas avec une **Précision** de : **96.8% et 96.4% .**

SVM / Kernel function: Cubic /Kernel scale: Automatic / Box constraint level: 2
Multiclass method: One-vs-One /Standardize data: true /Standardize data: true

Conclusion :

Dans notre expérimentation nous avons choisi 21 sujets qui ont participé aux exercices (activités), ce qui nous a permis de collecter 21 fichiers ; En conséquent nous avons entamé la première phase de prétraitement réalisée manuellement, en suite la deuxième phase incluant la segmentation et la préparation de données pour le traitement. La troisième phase a été consacrée pour le filtrage et le conditionnement des signaux en appliquant des filtres (fonctions Matlab).

Pour la quatrième phase nous avons ploté, visualiser et analyser les données. Les phases (2, 3, 4) ont été réalisées par des scripts Matlab.

Dans la partie de l'extraction des caractéristiques nous avons discuté dans les deux domaines temporel et fréquentiel, les fonctions qui ont été jugées plus sensibles et utiles, guidés par une métrique (paired t-test), et nous avons sélectionnés 56 caractéristiques parmi 73 qui ont été retenues au démarrage.

CHAPITRE III

IMPLÉMENTATION

Dans ce chapitre, nous présentons les détails de conception de notre système de reconnaissance d'activités humaines en ligne et de ses fonctionnalités (**figure 20**) ; Alors nous décrivons l'environnement de création du système, la mise en œuvre des différents programmes, les interfaces et les bases de données, qui servent à sa constitution, ensuite nous présentons quelques interfaces résultantes.

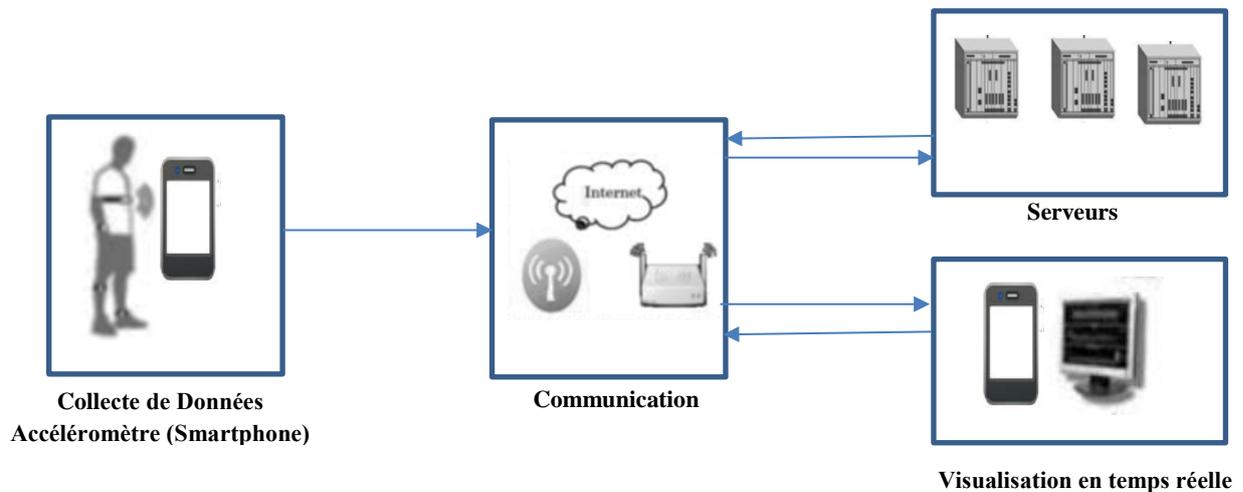


Figure 20 : Architecture générale

1. Architecture détaillée de système :

Notre système comporte Cinq composants principaux : le client/Smartphone, le serveur central (intermédiaire), le serveur base de données, serveur Web et le serveur de traitement (machine Learning), Comme illustrée à la **figure 21**.

L'application du smartphone collecte les données du capteur (accéléromètre) et les envoie au serveur central (intermédiaire) via Wifi. Par conséquent, elle peut être divisée en 3 activités : la 1ère est l'identification de l'utilisateur (Login), la 2ème est l'enregistrement le cas échéant (RegisterActivity) et la 3ème est de démarrer/stopper le service de collecte de données (de taille fixe : window=50 sample) et aussi leurs transmissions (données + nom d'utilisateur) (RecordService).

Nous utilisons le Protocole http pour établir la connexion et pour transférer les données.

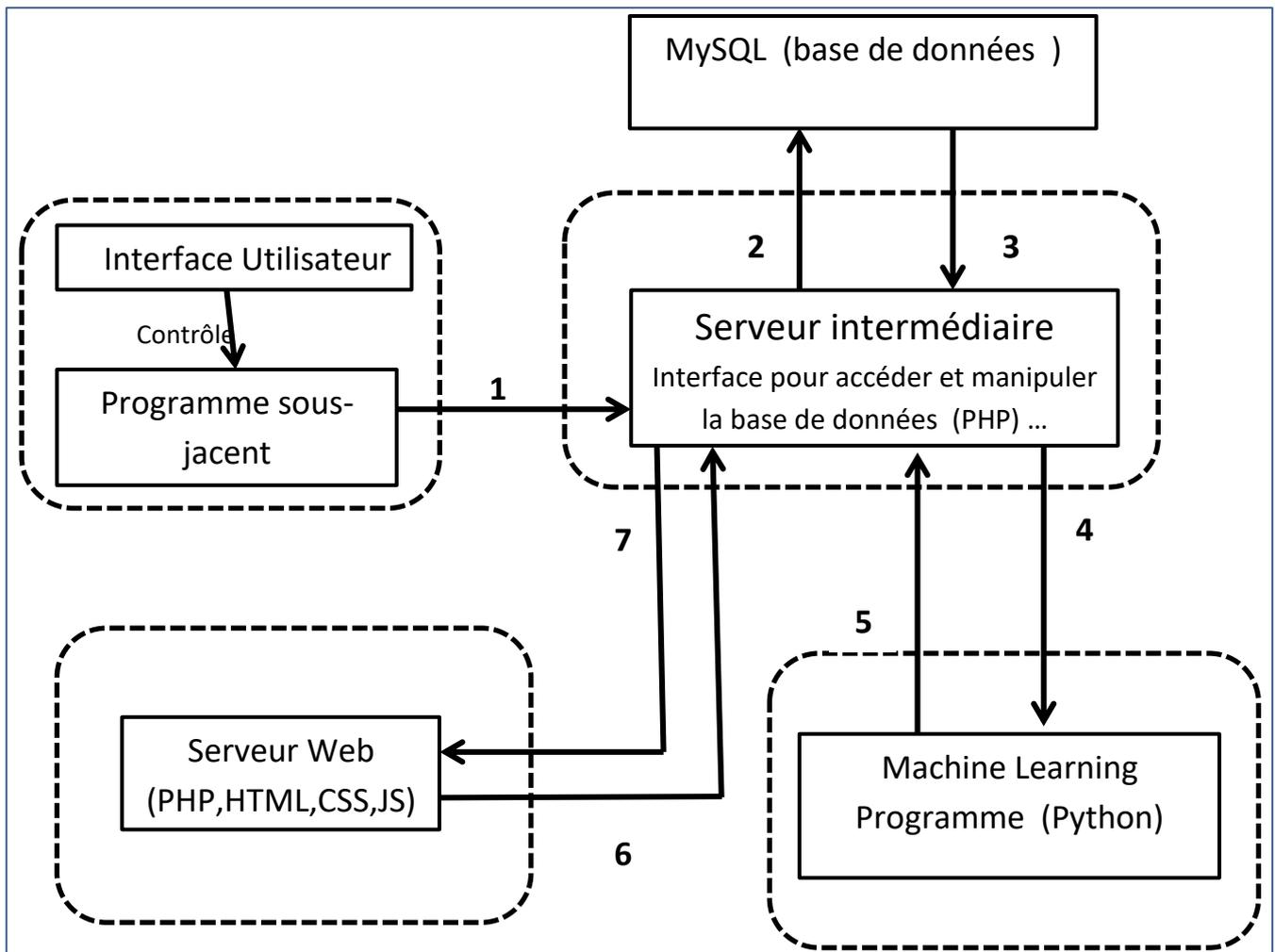


Figure 21 : Architecture détaillée

Légende :

- (1) : Envoi des données d'accéléromètre segmentées (de taille fixe window =50).
- (2) : Enregistrement des données dans la base de données.
- (3) : Importer les données demandées par ML ou Web Server de la BDD.
- (4) : Transmettre les données vers ML.
- (5) : Renvoi le résultat de reconnaissance.
- (6) : Demande périodique de données à afficher.
- (7) : Envoi les résultat à afficher au Serveur Web.

2. Configuration utilisée dans l'implémentation :

Dans le système proposé, nous avons utilisé des environnements virtuels :

Wampserver : Apache http server, MySQL serveur BDD, php.

Virtualenv : Un outil pour créer des environnements Python isolés. Il crée un dossier contenant tous les exécutables nécessaires pour utiliser les packages dont un projet Python aurait besoin (**figure 22**), et pour le développement nous utilisons l'environnement **Eclipse EE PyDev** (**figure 23**)

Le tableau suivant récapitule la configuration utilisée dans l'implémentation du système :

Smartphone :	Oppo A37f / QualcommMSM8916 Quad Cœur / Ram 2Go
Capteur :	Accéléromètre : lis3dh-accel de STMicroelectronics 0.01m/sec ² , max range 156.8, min delay : 5000 microseconds
PC Portable :	Core I3 , Ram 8 Go, DD 500 Go, Radeon GPU Core I7, Ram 8 Go, DD 600 Go, nVidia GEFORCE with CUDA, Wi-Fi 802.11b/g/n
Modem Wifi	TP-Link /Wireless N ADSL2+
Plateforme :	Android 4.1 Windows 10
Logiciel et Environnement :	Matlab R 2016 a (64bit) Pathon 3.6.0, Pathon library package (Numpy, Pandas, Scikit_learn, Sipy, Joblib, Virtualenv,) Android Studio 2.2.3 Eclipse EE PyDev 5.7.0 Wamp server 2.5 (Apache 2.4.9, PHP 5.5.12, MySQL 5.6.17) ArgoUML
Activités :	Marche, Assis, Debout, Allongé, Monte l'escalier, Descend l'escalier
Dataset :	Collecté à partir de 21 sujets "Centre d'accueil OEB"
La fréquence d'échantillonnage :	50Hz / Sensor speed : Game / Sensor unit : meters
Caractéristique Utilisé :	56 Caractéristiques (sélectionnées dans l'expérimentation)
Classificateur :	SVM

Tableau 17 : Configuration Utilisée dans l'implémentation

```

C:\WINDOWS\system32\cmd.exe

C:\env>cd scripts

C:\env\Scripts>activate.bat
(env) C:\env\Scripts>cd..

(env) C:\env>cd server_m1

(env) C:\env\server_m1>dir
Le volume dans le lecteur C n'a pas de nom.
Le numéro de série du volume est 963A-BEEC

Répertoire de C:\env\server_m1

16/05/2017  23:31    <DIR>        .
16/05/2017  23:31    <DIR>        ..
15/05/2017  20:53            380 .project
15/05/2017  20:53            431 .pydevproject
16/05/2017  21:25    <DIR>        data
16/05/2017  22:07            6 239 Features.py
16/05/2017  21:37            2 065 Model.py
16/05/2017  22:03            1 979 Recon.py
16/05/2017  21:56            2 689 855 svm.pkl
16/05/2017  22:06    <DIR>        __pycache__
                6 fichier(s)        2 700 949 octets
                4 Rép(s)  47 779 299 328 octets libres

(env) C:\env\server_m1>python Recon.py
    
```

Figure 22 : Environnement virtuel Virtualenv

The screenshot shows the Eclipse IDE with the PyDev plugin. The left sidebar displays the project structure for 'server_c', including folders for 'data', 'svm', and 'svmops', and Python files like 'Features.py', 'Model.py', and 'Recognition.py'. The main editor window shows the code for 'Recognition.py', which includes imports for threading, time, urllib, pandas, and numpy. The code defines a 'process' function that handles incoming requests, decodes the data, and processes it using a pre-trained SVM model. The console at the bottom shows the output of the program running on an Android device.

```

7@ import threading
8 import time
9 from urllib.request import urlopen
10 from urllib.parse import urlencode
11 import pandas as pd
12 import numpy as np
13
14
15
16 map = {'1.0': 'WALKING',
17        '2.0': 'UPSTAIRS',
18        '3.0': 'DOWNSTAIRS',
19        '4.0': 'SITTING',
20        '5.0': 'STANDING',
21        '6.0': 'LYING'}
22 fs = 50
23 def process(ID):#, ID1, ID2, ID3, ID4, ID5):
24
25     data = urlencode({"usercookie": ID})
26     conn = urlopen("http://117.0.0.1/har2/get.php", data.encode(encoding='utf_0', errors='strict'))#ID.encode('ascii')) #('ascii'))
27     #print(conn)
28     msg = conn.read().decode('utf-8-sig')
29     #print(msg)
30     info = msg.split("\n")
31     #print(info)
32
33     xyz = pd.read_json(info[0])
34     print(len(xyz))
35     axis = pd.DataFrame.as_matrix(xyz)
36
37     if len(axis) != 50:
38
39         return
40
41     feat = np.zeros((1,44))
42
43     import Features as fea
44
    
```

Figure 23 : Eclipse EE Pydev

3. Cycle de vie d'une activité Androïde :

Le système, pour des raisons de priorisation d'activités (coup de téléphone), peut tuer une activité quand il a besoin de ressources. Pour cette raison, aucune activité ne peut penser pouvoir vivre jusqu'au bout de son traitement.

Une activité possède quatre états [30] que sont :

« Active » : l'activité est lancée par l'utilisateur, elle s'exécute au premier plan ;

« En Pause » : l'activité est lancée par l'utilisateur, elle s'exécute et est visible, mais elle n'est plus au premier plan. Une notification ou une autre activité lui a volé le focus et une partie du premier plan ;

« Stoppée » : l'activité a été lancée par l'utilisateur, mais n'est plus au premier plan et est invisible. L'activité ne peut interagir avec l'utilisateur qu'avec une notification ;

« Morte » : l'activité n'est pas lancée.

Le schéma suivant indique le cycle de vie d'une activité et les méthodes appelées lors des changements d'état :

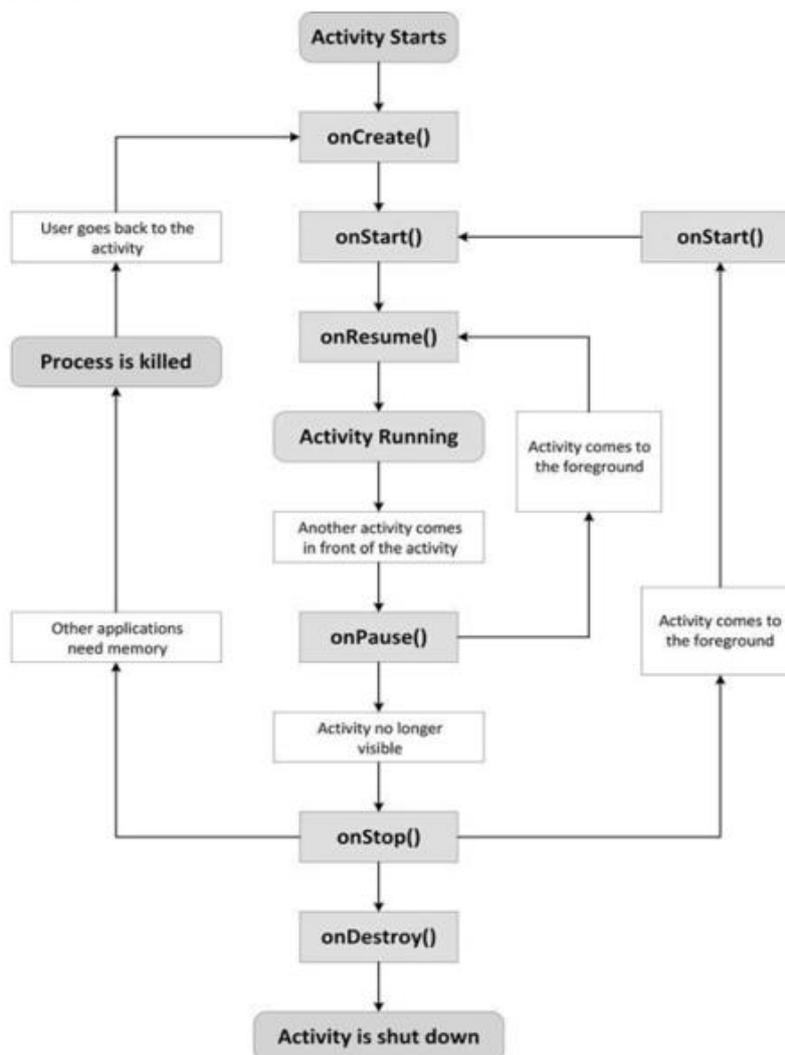


Figure 24 : Cycle de vie d'une activité Android [31]

4. Diagramme UML :

4.1. Cas d'utilisations :

Nous avons jugé que les deux cas d'utilisations les plus importants à présenter concernent l'application Androïde et le serveur Central.

4.1.1. Application Androïde

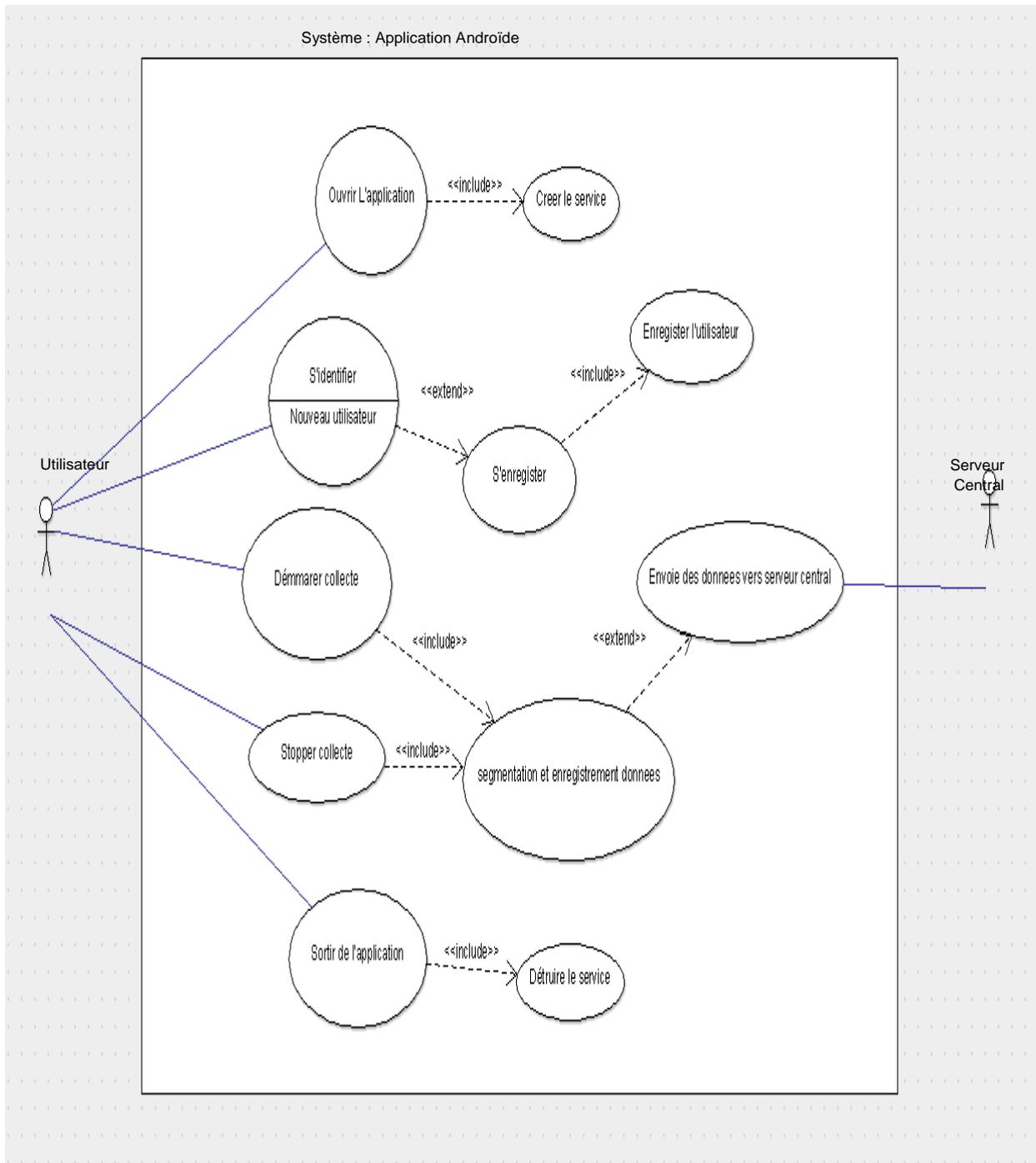
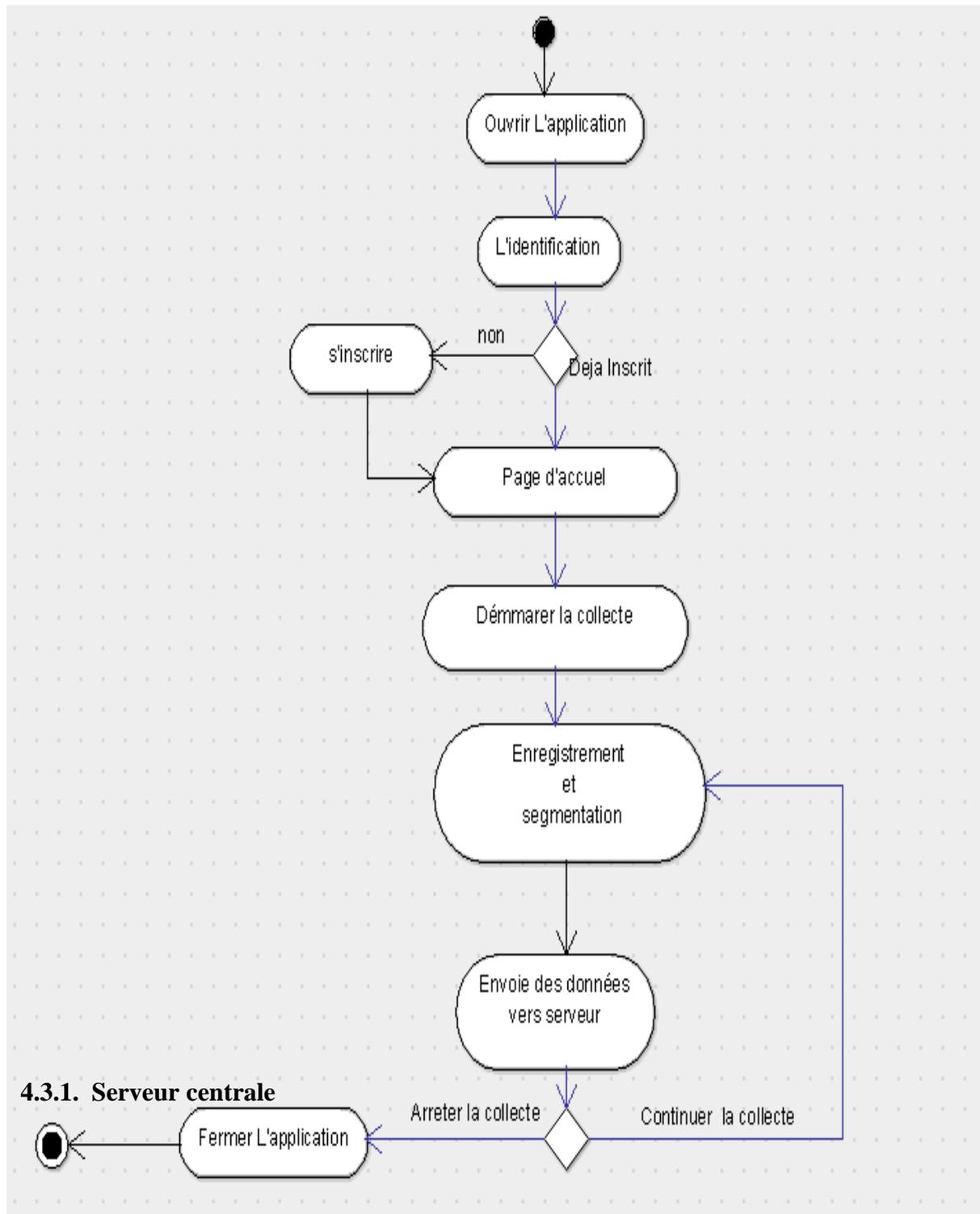


Figure 25 : Diagramme Cas d'utilisation : application Androïde

4.3. Diagramme d'activités :



4.3.1. Serveur centrale

Figure 28 : Diagramme d'activités - Application Androïde

5. Modules et Fichiers :

5.1. Python :

Pour la création du modèle de reconnaissance d'activités avec le classifieur SVM à partir du dataset, nous utilisons deux modules **model.py** (Création de modèle) + **features.py** (extractions des caractéristiques). Pour la reconnaissance des activités à partir des données transmis online nous utilisons deux modules **Recon.py** (Reconnaissance d'activité en cours). *(Le code source des trois modules annexé en annexe 2)*

5.2. PHP :

Nous présentons ci-après les fichiers php que nous avons utilisé au niveau du serveur central :

- **connection.php** : Pour créer une connexion avec la BDD MySQL.
- **insert.php** : Pour insérer dans la BDD les données poster par android.
- **get.php** : Pour récupérer les données d'accéléromètre des utilisateurs dans la BDD et les envoyer au ML.
- **getUsernames.php** : Pour récupérer les noms des utilisateurs dans la BDD et les envoyer au ML.
- **result.php** : Pour insérer dans la BDD les données (résultat de reconnaissance) poster par ML.
- **get2.php** : pour récupérer les données (résultat de reconnaissance) des utilisateurs dans la BDD et les envoyer au web Server.

5.3. Tables BDD :

Colonne	Type	Null	Défaut	Commentaires
ID_t1	int(20)	Non		Id table axes d'accéléromètre
IDuser	int(20)	Non		Id d'utilisateur
x	float	Non		Axe x
y	float	Non		Axe y
z	float	Non		Axe z

Tableau 18 : Table user_t1

Colonne	Type	Null	Défaut	Commentaires
IDuser	int(20)	Non		Id d'utilisateur
username	varchar(20)	Non		Nom d'utilisateur
value	varchar(15)	Non		Valeur de l'activité

Tableau 19 : Table user_t2

5.4.Interface Application Androïde :

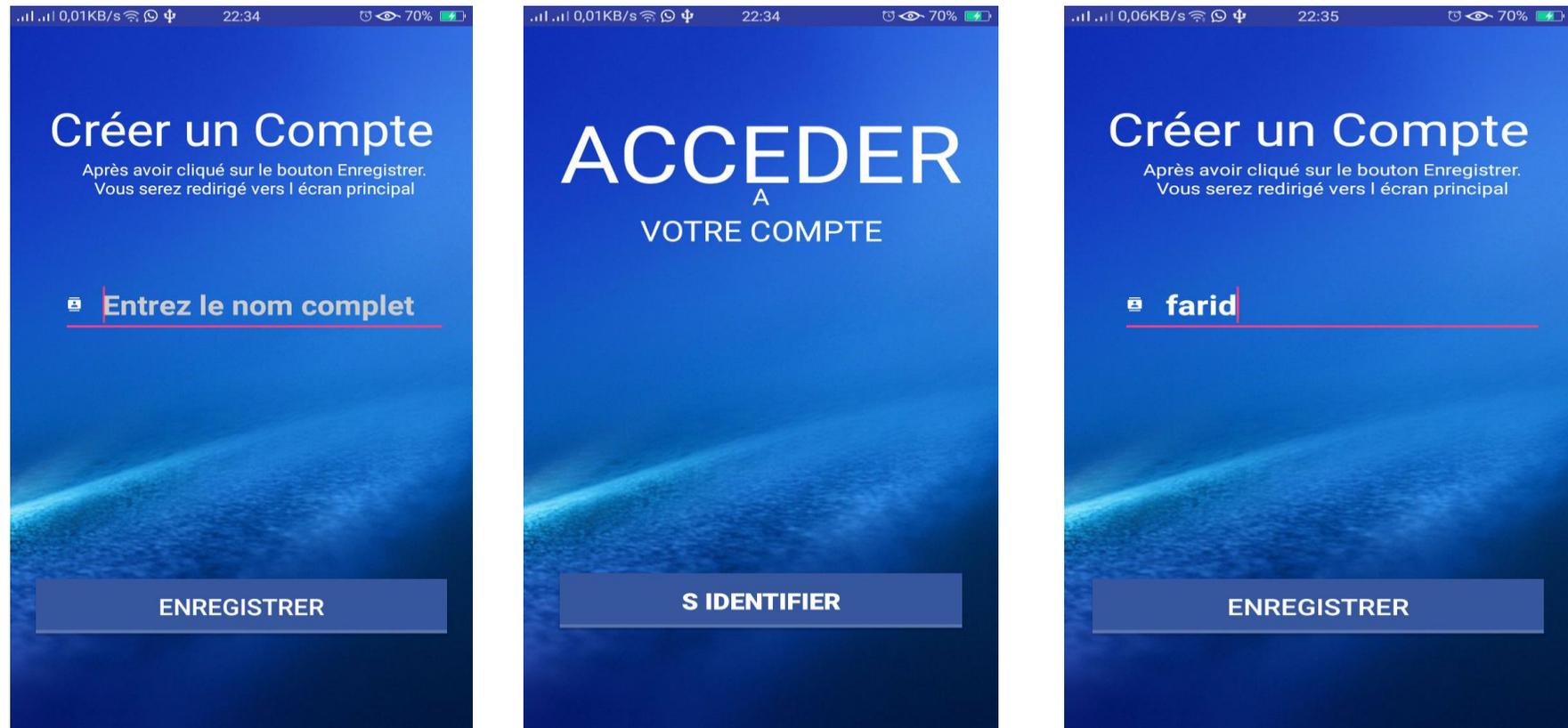


Figure 29 : Interface application Androïde 1 - 2 - 3

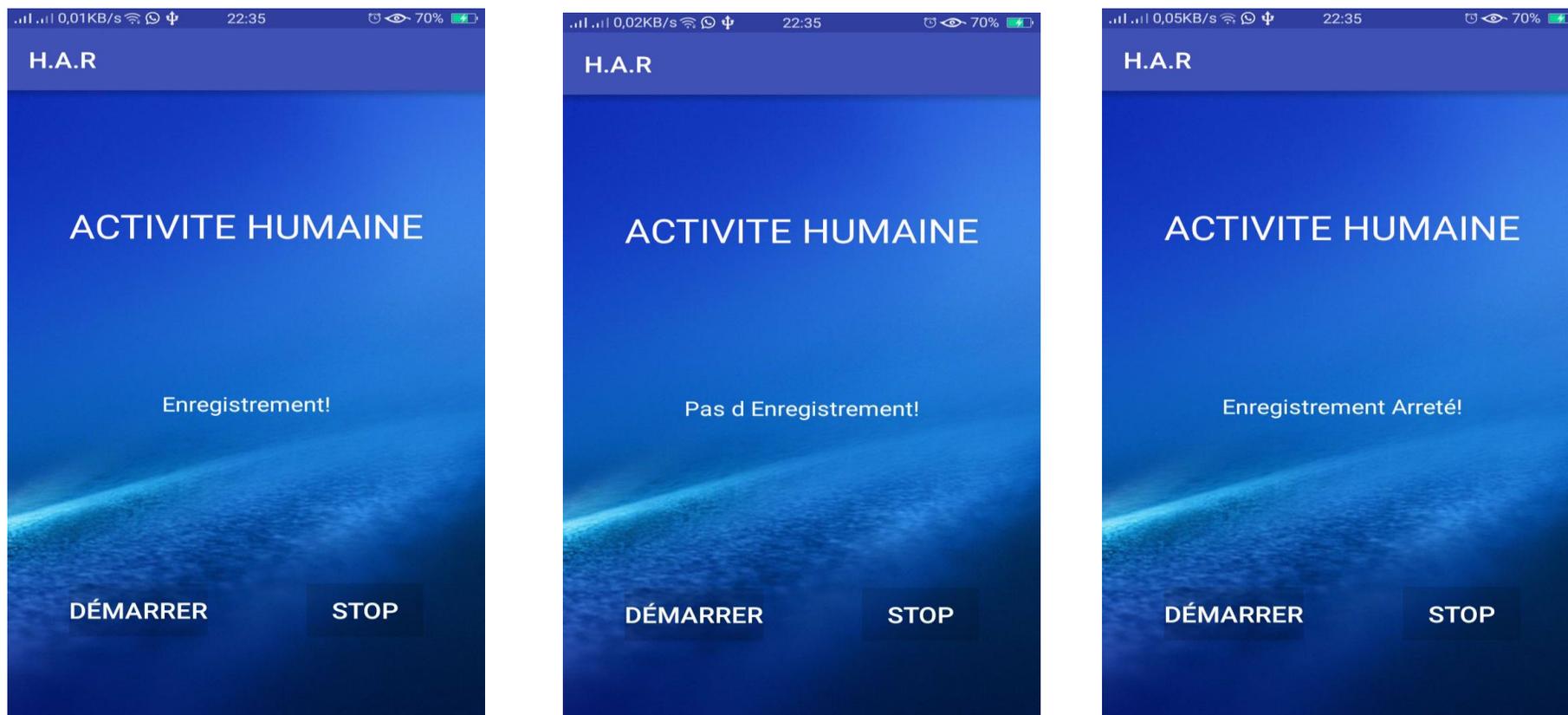


Figure 30 : Interface application Androide 4 - 5 - 6

5.5.Interface Page Web :

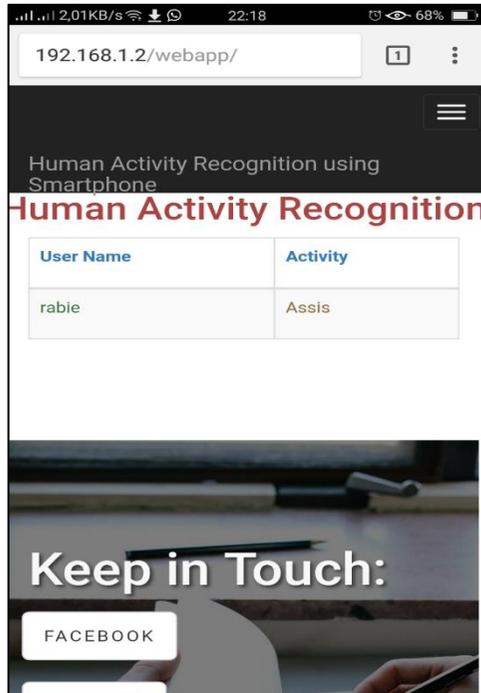


Figure 31 : Interface Page Web - Version Smartphone

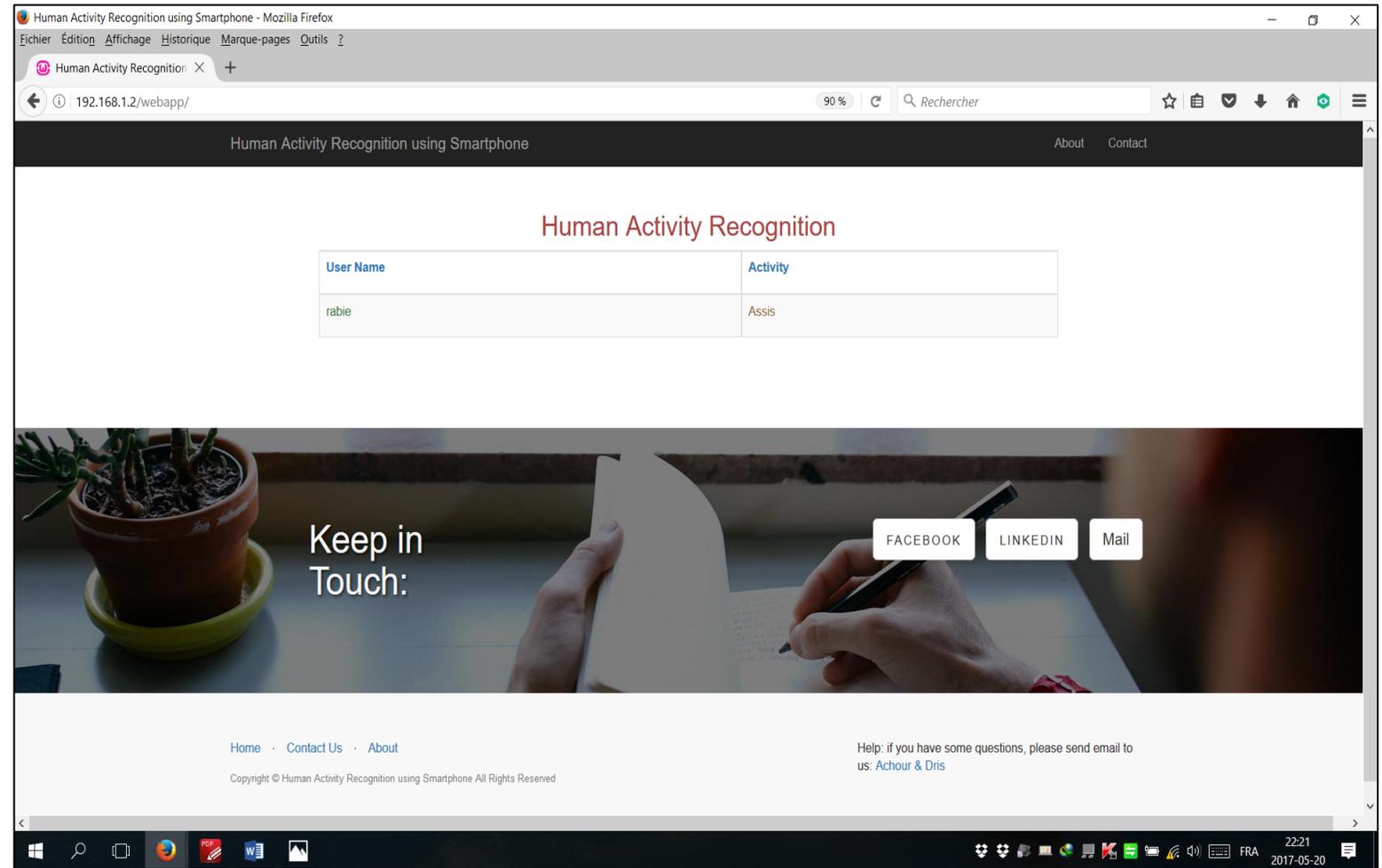


Figure 32 : Interface Page Web - version PC

Conclusion :

Dans ce chapitre, nous avons en premier lieu, présenté les architectures de notre système : générale et détaillée, la configuration utilisée dans l'implémentation de notre application. Par la suite, nous avons présenté quelques diagrammes UML, les interfaces et la base de données. L'application fonctionne correctement et offre toutes les fonctions utiles et nécessaires à la prédiction, mais nous espérons encore l'améliorer pour la rendre plus attractive.

CONCLUSION

GENERALE

Conclusion générale

Initialement le choix de ce thème était un challenge, pour apporter un plus à un volet de recherche innovant et en pleine évolution, notamment qu'il est pris comme sujet de recherche par des pionniers du domaine.

Nous avons proposé un système pour la reconnaissance de l'activité humaine à l'aide d'un capteur de smartphone au sein d'un centre d'accueil des personnes âgées.

Comme nous avons dévoilé dans le chapitre deux, la classification des caractéristiques a été réalisée par quatre classificateurs qui sont largement utilisés dans le domaine d'apprentissage supervisé. Nous les avons appliqués sur deux datasets ; la première est celle que nous avons collectée au sein du centre d'accueil des personnes âgées, la deuxième est certifiée et publiée et issue d'une expérimentation.

Les résultats que nous avons obtenus nous ont permis de choisir et de réduire au maximum le nombre de filtres à côté d'un taux de précision élevé. Nous avons également abouti, après deux comparaisons, au meilleur classificateur de notre expérimentation.

Sans oublier que nous avons obtenu des résultats proches à ceux généralement obtenus dans des cas similaires avec seulement un capteur (accéléromètre), ce qui donne une optimisation globale des ressources et un grand avantage en matière de : temps de traitement, volume du flux de données, économie d'énergie et simplicité de traitement.

Malgré ces résultats concrets ont été mis en face à des difficultés et diverses contraintes citons : notre manque de recul dans ce domaine déplacé par rapport au nôtre, la période courte de recherche, manque de dataset fiable et gratuit et la difficulté de collecte.

Notre perspective est d'ajouter une amélioration à notre travail en intégrant des capteurs multi-modaux : capteurs de pouls cardiaque, températures, tensions artérielles qui seront bénéfiques pour le bien-être et une meilleure prise en charge des personnes âgées.

Il est important à noter que la réalisation de ce mémoire nous a été bénéfique sur tous les plans. Nous avons développé nos connaissances en matière de recherche, d'analyse et d'expérimentation, ça nous a donné une bonne occasion pour découvrir et maîtriser plusieurs techniques dans le Machine Learning.

BIBLIOGRAPHIE

Bibliographie

- [1] A. Bulling, U. Blanke, and B. Schiele, "A tutorial on human activity recognition using body-worn inertial sensors,"
- [2], [18] "Android api description,"
https://developer.android.com/guide/topics/sensors/sensors_overview.html [20-04-2017]
- [3] C.-C. Yang and Y.-L. Hsu, "A Review of Accelerometry-Based Wearable Motion Detectors for Physical Activity Monitoring,"
- [4] A. Mannini and A. M. Sabatini, "Machine learning methods for classifying human physical activity from on-body accelerometers," *Sensors*, 2010.
- [5], [24] J. Yin, Q. Yang, and J. J. Pan, "Sensor-based abnormal human-activity detection," *Knowledge and Data Engineering, IEEE* 2010.
- [6] Département des statistiques de santé et de l'informatique risques sanitaires globaux dans information « : la mortalité et le fardeau de la maladie attribuables au major choisi risque », Genève, l'organisation mondiale de la santé, 2009]
- [7] «De l'activité physique et de la santé» un rapport du chirurgien General, Centres pour le contrôle et la prévention des maladies, du centre national pour l'empêchement de maladie chronique et la promotion de santé, Atlanta, 1996.
- [8], [9],[10] E. Biernat , M. Lutz "Data science : "Fondamentaux et études de cas Machine learning avec Python et R", Eyrolles,2015,ISBN:978-2-212-14243-3 .
- [11] Diane J. Cook, Narayanan C. Krishnan: "Activity Learning Discovering, Recognizing, and Predicting Human Behavior from Sensor Data ", 2015 ,John Wiley & Sons, Inc
- [12], [13],[15],[17] Miguel A. Labrador, Oscar D. Lara Yejas : "Human Activity Recognition Using Wearable Sensors and Smartphones", CRC Press ISBN 13: 978-1-4665-8828-8
- [14], [16] Yin, Xizhe, "Leveraging Smartphone Sensor Data for Human Activity Recognition" (2016). *Electronic Thesis and Dissertation Repository*. 4292.
- [19] B. Najafi, K. Aminian, A. Paraschiv-Ionescu, F. Loew, C. J. Bula, and P. Robert, "Ambulatory system for human motion analysis using a kinematic sensor: monitoring of daily physical activity in the elderly," *Biomedical Engineering, IEEE*
- [20] Jatoba, L.C.,Grossmann, U., Kunze, C.,Ottenbacher, J., Stok, W.(2008)Context aware mobile health monitoring: Evaluation of different pattern recognition methods for classification of physical activity. In *Engineering in medicine and Biology Society*, 2008 EMBS.
- [21] Maurer, U., Smailagic, A.,Siewiorek,D.P.Deisher, M (2006, April). Activity recognition and monitoring using multiple sensors on different body positions. In *wearable and Implantablebody sensor networks*,2006. BSN 2006.IEEE
- [22] J. Parkka, M. Ermes, P. Korpiäa, J. Mantyjarvi, J. Peltola, and I. Korhonen, "Activity classification using realistic data from wearable sensors," *Information Technology in Biomedicine, IEEE*
- [23] Guiry, J.J., van de Ven, P., Nelson, J., Warmerdam, L., Riper, H.(2014). Activity recognition with smartphone support. *Medical engineering & physics*, 36(6), 670-675.
- [25], [26] Jorge-Luis Reyes-Ortiz, Davide Anguita, Alessandro Ghio, Luca Oneto, and Xavier Parra. "Human activity recognition using smartphones", *doctoral thesis, Springer theses* 2015. ISBN:978-3-319-142-73-9
- [27] L.Ladha, T.Deepa, "Feature selection methods and algorithms", *International Journal on Computer Science and Engineering (IJCSE)* 2011.
- [28], [29] <http://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones> [05-05-2017]
- [30], [31] <http://mathias-seguy.developpez.com/tutoriels/android/comprendre-cyclevie-activite/> [05-05-2017]

ANNEXES

Annexe 1 :

❖ Installation et configuration de WampServer

L'installaion de **WampServer** est facile et son utilisation très intuitive permet de le configurer très rapidement.

Utilisation WampServer :

- Création automatique d'un répertoire « www » lors de l'installation (typiquement c:\wamp\www).
- Créer un sous répertoire pour le projet et déposer les fichiers PHP (copier les 2 dossiers au répertoire wamp\www)
- Cliquer sur le lien « Localhost » du menu de WampServer ou ouvrir le navigateur préféré, et aller à l'adresse <http://localhost>

Rendre WAMP accessible :

Voici les étapes pour rendre WAMP accessible depuis LAN ou WAN :

Étape 1

D'abord, configurer le service WAMP sur le serveur et sur les services requis. Et assurer que le serveur WAMP est en ligne et qu'il fonctionne sur le navigateur en utilisant `http://localhost`.

Étape 2

Maintenant, rendre le serveur WMAP accessible dans le réseau local ou le WAN, obtenir `wamp\bin\apache\Apache2.4.9\conf\ httpd.conf` et l'ouvrir et trouver le code suivant:

```
<Directory "cgi-bin">
    AllowOverride None
    Options None
    Order allow,deny
    Deny from all
</Directory>
```

Et modifier le code comme ci-dessus:

```
<Directory "cgi-bin">
    AllowOverride None
    Options None
    Order allow,deny
    Allow from all
</Directory>
```

Étape 3

L'étape suivante consiste à **ouvrir le port** (80) du serveur afin que tous puissent accéder au serveur. Cela dépend du système d'exploitation qu' on utilise. suivre les étapes ci-dessous.

Ouvrir **Pare-feu Windows**, puis cliquer sur "**Paramètres avancés**", puis sélectionner "**Règles entrantes**" dans le panneau de gauche, puis cliquer sur "**Ajouter une règle ...**". Sélectionner "**PORT**" comme option dans la liste, puis dans l'écran suivant, sélectionner le protocole "**TCP**" et entrer le numéro de port "**80**" sous "**Port local spécifique**" puis cliquer sur le bouton "**Suivant**" et sélectionner "**Autoriser la connexion**" et Puis donner le nom général et la description à ce port et cliquer sur Terminé. Ensuite, "Redémarrer tous les services" de WAMP et accéder à votre machine dans LAN ou WAN.

❖ Installation de Python

Python aussi s'installe facilement.

Installation des packages

1- Exigences pour l'installation des packages

Les étapes à suivre avant d'installer d'autres packages Python

1-1 Installer pip, setuptools

Si Python 2 >= 2.7.9 ou Python 3 >= 3.4 installer à partir de python.org , alors déjà pip et setuptools sont installés , mais il faut passer à la dernière version:

```
Python - m pip installation - U pip setuptools
```

1-2 Optionnelle, Créer un environnement virtuel

Utilisation de virtualenv:

```
pip install virtualenv
```

```
virtualenv <DIR>
```

```
source <DIR>/bin/activate
```

Utilisation de venv:

```
python3 -m venv <DIR>
```

```
source <DIR>/bin/activate
```

1-3 Utiliser pip pour installer

pip est l'installateur recommandé.

Pour installer la dernière version de "SomeProject":

```
pip install 'SomeProject'
```

Pour installer une version spécifique:

```
pip install 'SomeProject == 1.4'
```

Pour installer une version supérieure ou égale à une et inférieure à une autre:

```
pip install 'SomeProject >= 1, <2'
```

Pour installer une version qui est "**Compatible**" avec une certaine version:

```
pip install 'SomeProject ~ = 1.4.2'
```

Dans ce cas, cela implique l'installation de toute version ">= 1.4.*" Qui est également ">= 1.4.2".

Dans notre cas nous avons installés :

```
pip install sklearn
```

```
pip install joblib
```

```
pip install matplotlib
```

```
pip install pandas
```

```
pip install virtuelenv
```

1-4 Source Distributions vs Wheels

Pip peut être installé à partir de Distributions de source (sdist) ou de Wheels, mais si les deux sont présents sur PyPI, Pip préférera une Wheel compatible.

Wheels sont un format de distribution pré- construit qui fournit une installation plus rapide par rapport aux distributions de source (sdist) , surtout lorsqu'un projet contient des extensions compilées.

pour le package numpy il faut installer numpy+mkl avec source Wheel (télécharger <https://pypi.python.org/pypi/numpy>) :

```
pip install numpy-1.12.1+mkl-cp36-cp36m-win_amd64.whl
```

pour le package scipy il faut l'installer avec source Wheel (télécharger <https://pypi.python.org/pypi/scipy>) :

```
pip install scipy-0.19.0-cp36-cp36m-win
```

Annexe 2

- Attestation de stage au " Centre d'accueil des personnes âgées à OEB "
- CD contenant (les répertoires et les fichiers suivant):
 - Le répertoire HAR contient code source de l'application.
 - Le Mémoire au format Word 2013 (Mémoire HAR2017.docx)
 - Le Mémoire au format PDF (Mémoire HAR2017.pdf)