

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
Ministère de L'Enseignement Supérieur et de la Recherche Scientifique
Université Saad Dahleb Blida 1
Faculté des sciences



MÉMOIRE

Présenté pour l'obtention du **diplôme de Master**

En : Informatique

Option : Ingénierie des logiciels

Thème

**Développement d'un modèle basé sur le Machine Learning pour la reconnaissance d'activité humaine :
(Application au domaine de la santé)**

Réalisé par : FLITA Hana

ZENKHRI Meriem

Soutenu le 09/07/2023, devant le jury, composé de :

Mme.	N. Toubaline	à USDB	Présidente
Mme.	L. Ouahrani	à USDB	Examinatrice
Mme.	Soraya Cheriguene	à USDB	Promotrice
Mme.	Zeyneb Zerouali	à Mobilis	Encadrante

Année universitaire : 2022/2023

Remerciement

Nous remercions dieu de nous donner la force et la santé pour pouvoir terminer ce projet.

Nous remercions notre promotrice Mme **Cheriguene Soraya**, pour son encadrement professionnel, sa présence avec ses conseils, son aide apporté durant le parcours et la patience dont elle a fait preuve. Et notre Encadrante Mme **Zerouali Zeyneb** d'avoir accepté de nous encadrer et de nous proposer ce sujet.

Nous tenons à remercier les membres du jury de nous avoir consacré de leur temps pour juger et évaluer notre mémoire.

Enfin, nous souhaitons exprimer notre reconnaissance particulière à Morsli Hamza pour ses précieuses orientations.

Dedicace

A mes chers parents, je vous remercie jamais assez pour votre sacrifices, amour et soutien. Sans eux je serai jamais là ou je suis maintenant.

A mes soeurs : Amel, Zineb et Amina qui n'ont jamais cessé de me supporter.

A mes deux familles Zenkfri et Bouguerche.

A mes chères copines pour leurs soutien.

A mes enseignants en particulier Mme F.Boukerch

A mon binome, mon accompagnante durant le parcours universitaire.

Meriem

Dedicace

À mes chers parents.

À mon frère Fath Eddine et mes sœurs Ihssane, Farah et Nourhane.

À Anfel et à tous mes amis.

Ainsi qu'à tous ceux qui m'aiment...

Hana

Résumé

Avec l'avènement des objets connectés, les systèmes de reconnaissance d'activité humaine sont devenus un ensemble d'outils informatiques qui cherchent à identifier les activités quotidiennes d'un individu dans le but d'améliorer sa qualité de vie. Ceci constitue l'objectif de notre projet.

Ce mémoire se concentre sur la reconnaissance d'activité humaine en utilisant le dataset mHealth, qui fournit des données précieuses sur les signaux de capteurs portables pour différentes activités. L'objectif principal de cette étude est de développer un modèle de classification efficace pour prédire les activités humaines à partir de ces données, nous avons étudié et comparé deux approches principales : l'apprentissage automatique.

Pour l'apprentissage automatique nous avons utilisé quatre algorithmes populaires : les forêts aléatoires, machine à vecteurs de support, K-plus proches voisins, les arbres de décision ainsi que la combinaison des trois derniers. En ce qui concerne l'apprentissage profond, nous avons utilisé des réseaux neuronaux convolutifs (CNN) ainsi que des réseaux récurrents (BiLSTM et LSTM).

Mots Clés : Reconnaissance d'activité humaine , apprentissage automatique, apprentissage profond

Abstract

With the advent of connected objects, human activity recognition systems have become a set of computer tools that seek to identify the mboxactivities mboxdaily of an individual in order to improve his quality of life. This is mboxthe goal of our project.

This thesis focuses on human activity recognition using the mHealth dataset, which provides valuable data on portable sensor signals for different activities. The main objective of this study is to develop an effective classification model to predict human activities from this data, we studied and compared two main approaches: machine and deep learning.

For machine learning we used four popular algorithms: random forests, support vector machine, K-closest neighbors, decision trees as well as the combination of the last three. Regarding deep learning, we used convolutional neural networks (CNN) as well as recurrent networks (BiLSTM and LSTM)

Keywords: Human Activity Recognition, Machine Learning, Deep Learning.

ملخص

تتناول هذه الأطروحة التعرف على أنشطة الإنسان باستخدام التعلم الآلي والتعلم العميق. مع تقدم التكنولوجيا وظهور الأجهزة المتصلة، أصبحت أنظمة التعرف على أنشطة الإنسان مجالاً هاماً لتحسين جودة حياة الأفراد.

يهدف هذا البحث إلى تطوير نماذج فعالة للتصنيف تستخدم في توقع وتحليل الأنشطة البشرية باستخدام مجموعة من البيانات، التي تحتوي على إشارات من أجهزة الاستشعار القابلة للارتداء.

في إطار هاته الدراسة، تم استكشاف اثنين من الطرق الرئيسية: التعلم الآلي والتعلم العميق. فيما يتعلق بالتعلم الآلي، تم استخدام أربعة خوارزميات شائعة بالإضافة إلى استخدام تركيبية منها. أما فيما يتعلق بالتعلم العميق، فتم استخدام الشبكات العصبية المتسلسلة والشبكات المتكررة لتحقيق أداء متفوق في التصنيف والتنبؤ بالأنشطة البشرية. توصلت الدراسة إلى نتائج واعدة تؤكد قدرة النماذج المطورة على التعرف الدقيق على الأنشطة البشرية. يشير هذا البحث إلى إمكانية استخدام التعلم الآلي والتعلم العميق في مجال تحسين جودة حياة الأفراد من خلال التعرف على نمط الأنشطة اليومية.

الكلمات المفتاحية: رصد النشاط البشري، التعلم الآلي، التعلم العميق

Table des matières

Table des figures	i
Liste des tableaux	ii
Liste des abréviations	iii
Introduction Générale	2
1 Apprentissage automatique pour La reconnaissance des activités humaines.	3
1.1 Introduction	3
1.2 Aperçu sur la reconnaissance d'activités humaine	3
1.2.1 Activité humaine	3
1.2.2 Reconnaissance d'activité humaine	4
1.2.3 Processus de la reconnaissance d'activité humaine	4
1.2.4 Approches HAR	6
1.3 Apprentissage automatique	7
1.3.1 Définition de l'apprentissage automatique	7
1.3.2 Taxonomie de l'apprentissage automatique	7
1.3.3 Approches basés sur l'apprentissage supervisé	9
1.4 Apprentissage profond	10
1.4.1 Définition de l'apprentissage profond	10
1.4.2 Couches de l'apprentissage profond	13
1.5 Travaux connexes	14
1.6 Conclusion	16
2 Conception d'un système de reconnaissance d'activité humaine	17
2.1 Introduction	17
2.2 Présentation de l'organisme ATM MOBILIS	17
2.3 Datasets	20

2.4	Contribution de notre travail	20
2.5	Conception architecturale	21
2.6	Conception détaillée	22
2.6.1	Dataset utilisé	22
2.6.2	Classification avec des modèles ML	24
2.6.3	Classification avec des modèles DL	25
2.6.4	Module d'évaluation	28
2.7	Conclusion	29
3	Expérimentation et Discussion	30
3.1	Introduction	30
3.2	Environnement de travail	30
3.3	Implémentation de la solution	31
3.3.1	Prétraitement	31
3.3.2	Entraînement des modèles	33
3.4	Résultats et discussions	35
3.4.1	Résultats des modèles ML	35
3.4.2	Résultats des modèles DL	40
3.5	Conclusion	44
	Conclusion Générale	45
	Bibliographie	46

Table des figures

1.1	Pipeline du processus de la reconnaissance d'activité humaine	5
1.2	Classification d'un système HAR en fonction de leurs approches.	6
1.3	Types d'apprentissage automatique	8
1.4	Exemple d'un modèle SVM	9
1.5	Exemple d'un modèle RF	10
1.6	Structure d'un neurone	11
2.1	Organigramme de Mobilis	19
2.2	Architecture de la solution	21
2.3	Position des capteurs	22
2.4	Partitionnement de dataset	23
2.5	Architecture globale de la combinaison.	25
2.6	Schéma d'architecture du réseau CNN utilisé	26
3.1	Illustration des proportions des exemples pour chaque activité.	31
3.2	Visualisation des proportions des exemples de chaque activité après l'équi- librage.	32
3.3	L'historique de Colab durant la phase d'apprentissage.	34
3.4	Matrice de confusion du SVM	37
3.5	Matrice de confusion du RF	37
3.6	Matrice de confusion du 3NN	38
3.7	Matrice de confusion d'arbre de décision	38
3.8	Matrice de confusion de la combinaison (SVM, DT, 3NN)	39
3.9	Matrice de confusion du CNN	42
3.10	Matrice de confusion du LSTM	42
3.11	Matrice de confusion du BiLSTM	43

Liste des tableaux

1.1	Travaux recents sur HAR	15
2.1	Tableau récapulatif sur les datasets	20
2.2	Description des caractéristiques de jeux de données mHealth.	23
2.3	Exemple d'une matrice de confusion	28
3.1	Scores obtenus par les algorithmes de ML	35
3.2	Tableau récapulatif des activités	36
3.3	Evaluation des algorithmes de DL	40

Liste des abréviations

ADL	<i>Activity Daily Living</i>
HAR	<i>Human Activity Recognition</i>
DL	<i>Deep Learning</i>
ML	<i>Machine Learning</i>
NB	<i>Naive de Bayes</i>
LR	<i>Linear Regression</i>
RF	<i>Random Forest</i>
SVM	<i>Support vector Machine</i>
DT	<i>Decision Tree</i>
KNN	<i>K-Nearst Neighbors</i>
CNN	<i>Convolutional Neural Network</i>
LSTM	<i>Long Short Term Memory</i>
BiLSTM	<i>Bidirectional Long Short Term Memory</i>
RNN	<i>Recurrent Neural Network</i>
ANN	<i>Artificial Neural Network</i>

Introduction Générale

Contexte

La reconnaissance d'activité humaine (HAR) est devenue un domaine de recherche clé en interaction homme-machine (IHM) et en informatique. Son principal objectif est de reconnaître et de comprendre les activités humaines en temps réel ou à partir de données enregistrées. Cela peut être fait en utilisant une variété de capteurs, tels que des accéléromètres, gyroscopes, magnétomètres, ou même des caméras et des microphones. Ces capteurs captent des données liées au mouvement, à l'orientation et à d'autres caractéristiques pertinentes, qui peuvent ensuite être traitées et analysées pour déduire les activités exécutées.

HAR a des applications dans divers domaines, y compris les soins de santé, les sports, la surveillance et l'interaction homme-ordinateur. Dans les soins de santé, HAR peut être utilisé pour surveiller les mouvements des patients et détecter des anomalies ou des changements de comportement, aidant dans les soins aux personnes âgées ou des programmes de réadaptation. Dans le sport, il peut fournir des informations sur la performance des athlètes et aider à l'entraînement ou la prévention des blessures. En surveillance, HAR peut aider à identifier les activités suspectes ou le comportement anormal dans les espaces publics. Dans l'interaction homme-ordinateur, il peut permettre le contrôle gestuel ou améliorer les expériences de réalité virtuelle.

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

Problématique

La reconnaissance d'activité humaine joue un rôle crucial dans de nombreux domaines tels que la santé, la sécurité et les applications mobiles. Les capteurs embarqués dans les dispositifs portables, tels que les smartphones et les montres connectées, fournissent une source précieuse de données pour détecter et classifier différentes activités réalisées par

les individus. Cependant, il reste encore des défis à relever pour améliorer la précision de la reconnaissance d'activité et développer des modèles efficaces. Parmi les questions relatives à ce problème : Comment choisir le modèle d'apprentissage automatique le plus adapté à la reconnaissance d'activité humaine à partir des données des capteurs ? L'un des objectifs majeurs d'un système de reconnaissance d'activité humaine est de comprendre le comportement humain et permettre aussi la classification des activités.

Dans ce travail, nous avons tenté de générer des modèles d'apprentissage automatique et profond dans le but d'exploiter le domaine de la reconnaissance d'activité humaine.

Objectifs

Notre objectif principal est de développer une application pour le processus de la reconnaissance d'activité humaine.

Les objectifs suivants doivent être atteints lors de cette mission :

- * Présenter un aperçu sur l'apprentissage automatique et profond.
- * Prétraiter les données pour les rendre appropriées à l'entraînement des modèles de ML.
- * Explorer différentes techniques d'apprentissage automatique ainsi profond.
- * Évaluation et comparaison des performances.

Plan du document

Notre travail est composé de trois chapitres structurés comme suit :

- Chapitre 1 : Dans ce chapitre, nous commençons par présenter une étude bibliographique sur la reconnaissance d'activité humaine, en mettant en avant les concepts fondamentaux de l'apprentissage automatique ainsi que les différents types utilisés. Nous aborderons également l'apprentissage profond.
- Chapitre 2 : Ce chapitre est dédié à la présentation de notre solution conceptuelle. Nous exposerons les principes de notre approche pour la reconnaissance d'activité humaine
- Chapitre 3 : Ce chapitre se concentre sur les détails de l'implémentation et de l'évaluation, ainsi que sur les résultats obtenus.

Enfin, nous concluons par une conclusion générale et des perspectives du projet.

Chapitre 1

Apprentissage automatique pour La reconnaissance des activités humaines.

1.1 Introduction

La reconnaissance des activités humaines (RAH) est un domaine en plein essor qui se concentre sur l'identification et la catégorisation automatiques des actions et des comportements humains en temps réel. L'apprentissage automatique joue un rôle central dans le HAR en fournissant les outils et les techniques permettant d'extraire des informations significatives des données collectées. Avec l'avènement des algorithmes, l'apprentissage automatique a permis aux systèmes HAR d'apprendre et de reconnaître automatiquement des schémas complexes et des dépendances dans les activités humaines.

Ce chapitre commence par introduire c'est quoi la reconnaissance d'activité humaine et son processus. Ensuite, L'apprentissage automatique, ses types et quelques-uns de ses algorithmes, ainsi que l'apprentissage profond, ses couches et ses algorithmes, seront abordés. Enfin, des travaux liés au projet.

1.2 Aperçu sur la reconnaissance d'activités humaine

1.2.1 Activité humaine

L'activité se réfère à une partie du corps ou à un mouvement du corps entier, consistant en plusieurs actions de base effectuées dans l'ordre chronologique.

Les types d'activités humaines qui ont été considérées en fonction de leur complexité, passant d'une action simple à une action plus complexe, sont comme suit : [1]

— Comportements : Ils décrivent l'ensemble des actions physiques et des réactions

des individus dans des situations spécifiques observables de l'extérieur et qui sont relatives à leurs émotions.

- Gestes : Typiquement, un geste est un langage ou une partie de la communication non verbale qui peut être utilisé pour exprimer des idées ou des ordres significatifs. Les gestes sont un deuxième type d'activités qui peuvent être conscientes comme « applaudir » et inconscientes comme « cacher le visage avec les mains quand on devient timide » ou « tirer la main quand on touche une matière chaude ».
- Interactions : Il s'agit d'actions ou d'échanges réciproques entre deux entités ou plus, qui modifient le comportement des individus ou des objets impliqués dans l'interaction. Il s'agit, en général, d'activités de deux types : d'humain à humain, comme le « câlin » ou de l'être humain à l'objet, comme la « lecture » dont l'objet est le livre.
- Actions de groupe : constituent les activités menées par un groupe de personnes comme les sports collectifs : football.
- Activité de la vie quotidienne (ADL = Activity of Daily Living) : des activités auxquelles les gens sont habitués tous les jours sans aucune aide. En général, il s'agit d'activités complexes, composées de nombreuses activités simples exécutées dans un environnement intérieur : cuisiner, dormir.

1.2.2 Reconnaissance d'activité humaine

La reconnaissance de l'activité humaine (HAR) est un domaine de recherche qui vise à identifier automatiquement les activités physiques via des techniques de l'intelligence artificielle (IA) à partir des données brutes d'activité. Ces données sont recueillies en utilisant diverses sources (dispositifs), tels que des accéléromètres, des gyroscopes et des caméras. L'objectif de HAR est de classer les actions physiques et les mouvements des individus dans des catégories prédéfinies, comme la marche, la course, le saut ou la position assise. Les applications de HAR peuvent être trouvées dans divers domaines, notamment la santé et le bien-être, les sports, le divertissement et les maisons intelligentes [2].

1.2.3 Processus de la reconnaissance d'activité humaine

Le processus de la reconnaissance d'activité humaine comporte plusieurs étapes clés, comme illustrer dans la figure 1.1

- a. **Collecte des données** : Cette étape consiste à recueillir des données à partir de capteurs ou des appareils de supervision tels que des accéléromètres, des gyroscopes,

les caméras et d'autres appareils portables.

- b. **Prétraitement** : Une fois les données sont collectés, elles doivent être prétraitées afin de supprimer toute information non pertinente, tout bruit ou toute valeur aberrante. Cette étape implique différentes techniques telles que la normalisation, le ré-échantillonnage (cas des données déséquilibrés) et autres pour préparer les données aux étapes suivantes [3].
- c. **Extraction des caractéristiques** : Cette étape consiste à filtrer les informations pertinentes en sélectionnant les caractéristiques les plus discriminantes, à partir des données pré-traitées, pour identifier l'activité humaine. Il existe deux types de vecteurs de caractéristiques pour la reconnaissance de l'activité humaine. Le premier concerne les caractéristiques statistiques comme la moyenne, la médiane, l'écart-type, le temps et le domaine de fréquence représentation. Ces caractéristiques sont basées sur les propriétés qualitatives des données acquises. D'autre part, les caractéristiques structurelles sont basées sur la relation entre les données du capteur mobile [4].
- d. **Classification** : Les informations extraites à l'étape précédente seront utiliser pour entraîner un modèle d'apprentissage automatique approprié pour la tâche de reconnaissance de l'activité humaine. Les modèles couramment utilisés comprennent les arbres de décision, les forêts aléatoires(RF), machine à vecteurs de support (SVM) et les réseaux de neurones profonds.

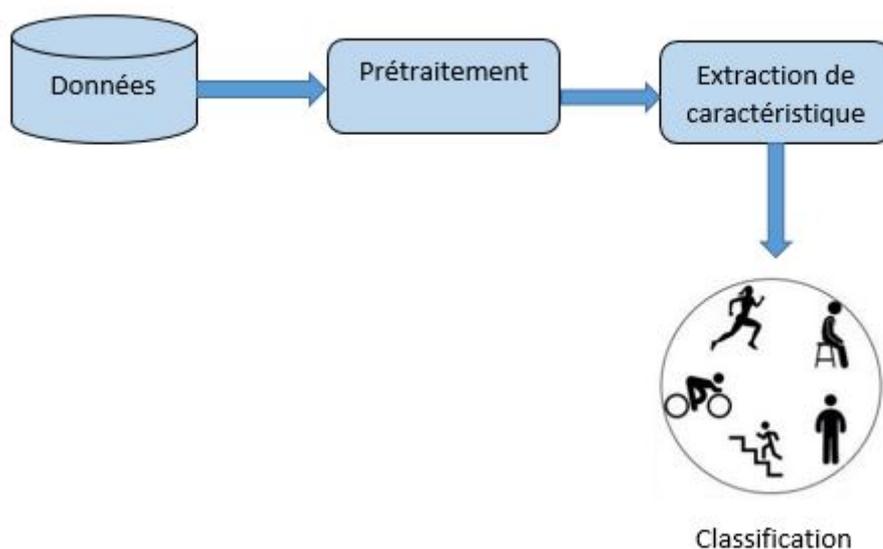


Figure 1.1: Pipeline du processus de la reconnaissance d'activité humaine

1.2.4 Approches HAR

Pour atteindre l'objectif de la reconnaissance d'activité humaine, des systèmes dotés de détection sont indispensables pour identifier l'activité. A cet effet, les approches HAR proposés dans la littérature peuvent être catégorisés en deux familles en fonction de la méthode de collection des données en entrée [5].

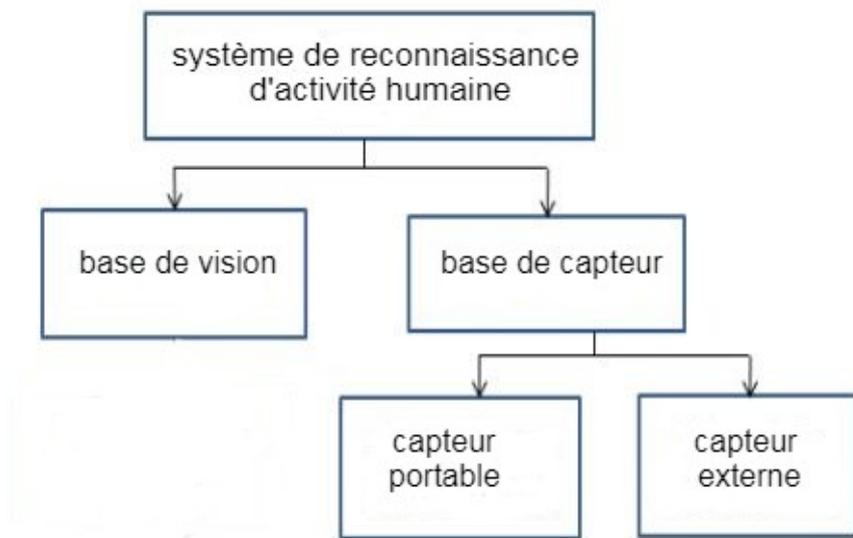


Figure 1.2: Classification d'un système HAR en fonction de leurs approches.

- **Basée sur la vision :** Où le système peut reconnaître des activités par l'analyse des données visuelles saisies par des caméras ou d'autres capteurs de vision. Ce type des méthodes HAR est basé sur l'utilisation des techniques de vision par ordinateurs et traitement d'images pour identifier les activités humaines à partir de séquences de vidéos. l'analyse des images s'appuient sur plusieurs paramètres comme la résolution, l'éclairage de l'image. Les auteurs de [6] sont intéressés par cette approche. Certaines activités ne peuvent être reconnues uniquement par une seule image statique actuelle. Il est nécessaire d'avoir accès à les informations relatives aux événements antérieurs et postérieurs, dans ce cas les séquences vidéo sont plus précises [7], [8], [9].
- **Basée sur capteur :** Un capteur est généralement défini comme un dispositif capable de percevoir et de mesurer un type particulier d'entrée (propriété ou caractéristique physique). Il fournit ainsi une sortie qui peut être transformée en informations lisibles [10].

Les approches basées sur les capteurs peuvent être classées en capteurs portables et capteurs externes.

Un capteur est dit **portable** s'il comprend un accéléromètre, un magnétomètre et un gyroscope, qui peuvent être portés par les utilisateurs ou intégrés à des appareils portatifs, comme des téléphones intelligents, des montres intelligentes [11].

Les capteurs d'objets (externe) font référence aux capteurs attachés à un objet particulier qui détectent leurs mouvements. Ils sont utilisés moins souvent que les capteurs portables en raison des coûts élevés et des difficultés d'installation [11].

1.3 Apprentissage automatique

1.3.1 Définition de l'apprentissage automatique

L'apprentissage automatique (Machine Learning en anglais) est une branche en évolution constante de l'informatique qui vise à reproduire l'intelligence humaine en apprenant de l'environnement. Elle fait référence à la conception et l'implémentation des algorithmes qui permettent à une machine d'évoluer à partir des données complexes grâce à un modèle d'apprentissage. De même, les algorithmes de l'apprentissage sont capables de s'améliorer automatiquement et de prendre de meilleures décisions sur des nouvelles données sans avoir besoin d'être explicitement programmé [12].

Les méthodes d'apprentissage automatique reposent sur trois éléments principaux sont l'algorithme, le modèle et data.

L'apprentissage automatique peut servir à résoudre des problèmes comme :

- que l'on ne sait pas résoudre.
- que l'on sait résoudre, mais dont on ne sait formaliser en termes algorithmiques comment nous les résolvons.
- que l'on sait résoudre, mais avec des procédures beaucoup trop gourmandes en ressources informatiques.

1.3.2 Taxonomie de l'apprentissage automatique

Les méthodes de l'apprentissage automatique peuvent être regroupées en différents types, la figure suivante représente le typage tel que L'axe des x représente la certitude des étiquettes et l'axe des y le pourcentage de données de formation étiquetées [13].

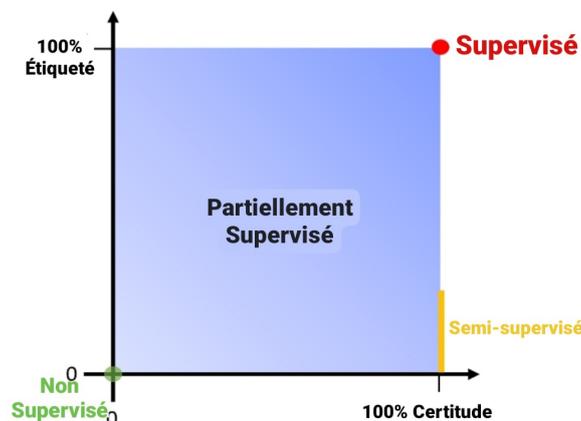


Figure 1.3: Types d'apprentissage automatique

1.3.2.1 Apprentissage supervisé :

Dans ce cas, 100% des données d'entraînement sont étiquetées et les étiquettes sont certaines à 100%. Pour chaque image utilisée pour entraîner le système, l'étiquette correspondante est connue et il n'y a aucune incertitude quant à celle-ci. Cela implique une étape supplémentaire par rapport aux autres types, à savoir l'étiquetage des données, qui est généralement effectué manuellement par des experts du domaine [13].

On peut distinguer 2 types de l'apprentissage supervisé sont :

- **classification** : lorsque les données proviennent d'un ensemble fini de valeurs, et les valeurs de sortie sont discrètes.
- **régression** : Dans ce type, l'étiquette est continue, cette technique est utilisée dans la prédiction des quantités, tailles, valeurs...etc. Les valeurs de sortie sont continues.

1.3.2.2 Apprentissage non-supervisé :

Le type d'apprentissage non-supervisé est en constante évolution par rapport à l'apprentissage supervisé, car il est jugé plus efficace vu que la plupart des données dans la nature sont généralement non étiquetées. Ce type d'apprentissage vise dans sa globalité à claudrer des données non étiquettes en créant ainsi des groupes contenant les données les plus similaires.

1.3.2.3 Apprentissage semi-supervisé :

Dans certaines sources, l'apprentissage semi-supervisé n'est pas toujours catégorisée comme un type distinct à part entière, car il combine en effet des éléments de l'apprentis-

sage supervisé et de l'apprentissage non-supervisé, L'apprentissage semi-supervisé utilise un ensemble de données non étiquetées. Il utilise d'abord l'apprentissage non-supervisé sur une partie des données pour construire les classes de sortie. Ensuite, il applique l'apprentissage supervisé pour prédire les classes de sortie des données restantes en se basant sur les classes établies par l'apprentissage non-supervisé [14].

1.3.2.4 Apprentissage par renforcement :

Ce type ne repose pas sur des exemples de données comme les autres mais sur des stimuli provenant de l'environnement d'un agent, à tout moment l'agent peut effectuer des actions pour les mènera à un nouvel état où l'objectif principal est de trouver le comportement idéal de ce agent [13].

1.3.3 Approches basés sur l'apprentissage supervisé

Dans notre travail, on s'intéresse à l'apprentissage supervisé, dans ce qui suit, une description des algorithmes du ML est présentée :

Machine à vecteurs de support (SVM) : est un algorithme d'apprentissage automatique supervisé utilisé pour la classification et la régression. Son objectif est de trouver un hyperplan dans un espace à N dimensions qui classifie distinctement les points de données. La dimension de l'hyperplan dépend du nombre de caractéristiques. Si le nombre de caractéristiques d'entrée est de deux, alors l'hyperplan est juste une ligne. Si le nombre de caractéristiques d'entrée est de trois, l'hyperplan devient un plan bidimensionnel. Il devient difficile à imaginer lorsque le nombre de caractéristiques dépasse trois [15], voici une figure 1.4 le représente :

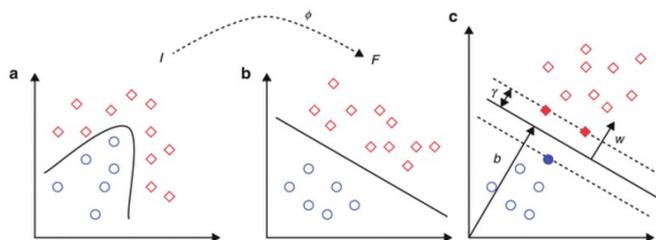


Figure 1.4: Exemple d'un modèle SVM

K plus proches voisins (KNN) : est l'une des méthodes de classification les plus fondamentales et les plus simples. Elle devrait être l'un des premiers choix pour une étude de classification lorsqu'il y a peu ou pas de connaissances préalables sur la distribution des

données. La classification par K plus proche voisin a été développée à partir de la nécessité d'effectuer une analyse discriminante lorsque des estimations paramétriques fiables des densités de probabilité sont inconnues ou difficiles à déterminer. Certaines propriétés formelles de la règle du k-plus proche voisin ont été élaborées par exemple, il a été montré que pour $k=1$ et $n \rightarrow \infty$ l'erreur de classification du k-plus proche voisin est limitée par deux fois le taux d'erreur de Bayes [16].

Arbre de décision : Les arbres de décision sont des modèles séquentiels qui utilisent des tests simples pour prendre des décisions. Un point de données est classifié par un arbre de décision en fonction de la classe la plus fréquente dans la région correspondante. Le taux d'erreur est le rapport entre les points mal classés et le nombre total de points de données [17].

Les forêts aléatoires (Random forests) : Les forêts aléatoires sont des prédicteurs bien connus et puissants, composés d'un ensemble d'arbres de décision récursifs binaires. Les forêts aléatoires sont facilement adaptables à la fois à la classification et à la régression, peuvent être parallélisées sans difficulté, et peuvent traiter différents types de variables (continues et catégorielles). Elles sont simples à utiliser, produisent de bons résultats avec peu ou pas de réglage, et peuvent être appliquées à une grande variété de domaines [18].

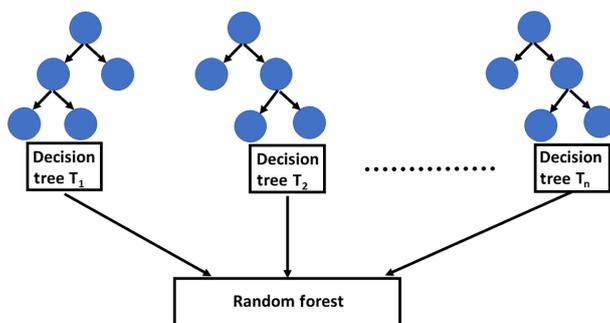


Figure 1.5: Exemple d'un modèle RF

1.4 Apprentissage profond

1.4.1 Définition de l'apprentissage profond

L'apprentissage profond est une branche spécifique de l'apprentissage automatique qui se base plus spécifiquement sur la notion de réseau de neurones artificiel. L'algorithme d'apprentissage profond est conçu comme une structure en couches qui tente de reproduire la structure du cerveau humain. En utilisant la composition de nombreuses fonctions non

linéaires, l'apprentissage profond permet de modéliser les dépendances complexes entre les caractéristiques d'entrée et les étiquettes [19] [20].

Les réseaux neuronaux profonds sont composés de plusieurs couches ou chacune recevant et interprétant les données de la couche précédente, d'où la notion de profondeur. Cette particularité leur permet de détecter des propriétés précises à partir de données non structurées. Le développement de ce type de réseaux a été conçu pour effectuer une extraction automatique de primitives à partir d'un ensemble de données brutes volumineux, sans recourir à des techniques préétablies de l'extraction de caractéristiques. Ceci a permis de résoudre des problèmes d'apprentissage beaucoup plus complexes que ne le font les réseaux de neurones moins profonds et représente une différence majeure par rapport à l'apprentissage automatique traditionnel.

Le réseau de neurones artificiels (ANN) : La notion de "neurone formel" (ou simplement "neurone") désigne une fonction algébrique non linéaire et bornée, dont la valeur est déterminée par des paramètres appelés coefficients ou poids. Les variables de cette fonction sont généralement désignées comme les "entrées" du neurone, tandis que la valeur produite par la fonction est appelée sa "sortie" (Voir la figure 1.6). Il est capable de modéliser des relations complexes, de plus, il présente une excellente tolérance aux erreurs, une grande rapidité et une grande évolutivité grâce au traitement parallèle. Les réseaux de neurones artificiels sont des modèles informatiques composés de nœuds inter-connectés appelés neurones artificiels, qui sont organisés en couches successives. Les ANNs sont utilisés pour des tâches d'apprentissage automatique et d'intelligence artificielle. Ils sont capables d'apprendre à partir de données en ajustant les poids et les biais des connexions entre les neurones qui donne une efficacité pour les tâches telles que la classification des images et la reconnaissance des paroles.

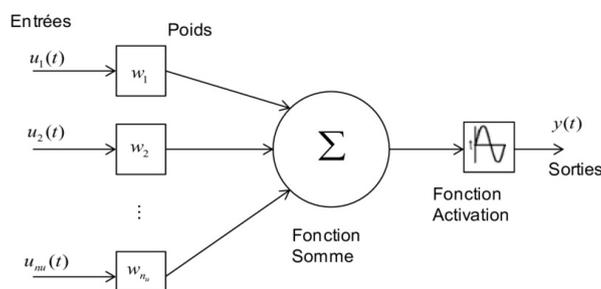


Figure 1.6: Structure d'un neurone

ANN présente un modèle de transformation de données constitué d'un ensemble d'unités (représentant les neurones), superposées en couches et inter-connectées via des connecteurs. Pour rendre la machine capable d'apprendre à partir des données fournies en entrée,

un réseau de neurones repose sur un grand nombre de neurones fonctionnant en parallèle et organisés en couches. La première couche reçoit des informations brutes en entrée, puis, chaque couche reçoit les sorties d'informations des couches précédentes. Cette organisation en couches permet un traitement progressif des données et des transformations complexes.

Différents types de réseaux de neurones sont distingués en fonction de divers critères. La variance des réseaux de neurones est déterminée selon la façon de transmission des informations entre les différentes couches de neurones. Il existe de nombreuses architectures de réseaux de neurones différentes qui peuvent être utilisées pour simuler cette propagation de l'information [21] [22].

Les réseaux de neurones récurrents (RNNs) : Un Réseau de Neurones Récurrent est un type de réseau de neurones qui traite des données séquentielles en maintenant une mémoire interne des entrées passées. Il possède des connexions qui permettent à l'information de circuler en boucle récurrente qui génère une sorte de mémoire. Cette flexibilité confère au RNN la capacité de capturer les dépendances dans les données et de conserver les informations des étapes précédentes de la séquence.

Mémoire à Long Terme (LSTM) : LSTM (Long Short-Term Memory) est une variante spéciale des RNNs qui est capable de se souvenir d'informations à plus long terme. LSTM est adapté pour traiter des problèmes liés aux séries temporelles, il peut apprendre et prédire efficacement des schémas dans des données séquentielles, même lorsque les dépendances s'étendent sur une longue plage, grâce à ses cellules et mécanismes [23].

Le LSTM bidirectionnel (BiLSTM) : Le BiLSTM (Bidirectional LSTM) est une extension de l'architecture traditionnelle LSTM qui introduit une couche de complexité supplémentaire. Tandis que le LSTM traite les données séquentielles dans une direction avant, le BiLSTM intègre à la fois un flux d'informations avant et arrière, ce qui lui permet de bénéficier des informations disponibles dans le futur pour réaliser des prédictions plus précises. En prenant en compte le contexte futur, le BiLSTM est capable de mieux gérer les situations où les observations à venir ont un impact sur la prédiction actuelle [24].

Réseau neuronal convolutif (CNN) : CNN est l'acronyme en anglais pour "Convolutional Neural Network", il est l'un des réseaux neuronaux profonds les plus populaires. Les CNNs comprennent plusieurs couches, notamment la couche de convolution, la couche de non-linéarité, la couche de pooling et la couche entièrement connectée. Les couches de convolution et entièrement connectée possèdent des paramètres, tandis

que les couches de pooling et de non-linéarité n'en possèdent pas. Il présente d'excellentes performances dans les problèmes d'apprentissage automatique [25].

1.4.2 Couches de l'apprentissage profond

- a. **Couche de convolution** : cette couche est un composant fondamental des réseaux de neurones convolutifs qui applique un ensemble de filtres ajustables aux données d'entrée. Ces filtres glissent ou se convolvent à travers les données d'entrée, effectuant des multiplications et des sommes élémentaires. Ce processus capture les motifs ou caractéristiques locales dans les données d'entrée, tels que les contours, les textures ou les formes.
- b. **Couche Pooling** : est responsable de la réduction de la taille spatiale de la sortie de la fonction convolutive. Cela permet de diminuer la puissance de calcul nécessaire pour traiter les données en effectuant une réduction de dimensionnalité. De plus, elle est utile pour extraire les caractéristiques dominantes qui sont invariants par rotation et par position, maintenant ainsi un processus efficace d'entraînement du modèle [26].
- c. **Couche Dropout** : est une technique utilisée pour prévenir le surapprentissage d'un modèle. Le Dropout fonctionne en fixant de manière aléatoire à zéro les sorties des unités cachées (neurones qui composent les couches cachées) à chaque mise à jour de la phase d'entraînement [27].
- d. **Couche Flatten** : est un type de couche de réseau neuronal qui réduit les dimensions spatiales de l'entrée en une seule dimension. Cela est généralement fait pour préparer l'entrée à une couche entièrement connectée, qui attend une entrée en un vecteur unidimensionnel [28].
- e. **Couche Dense** : ou bien couche entièrement connectée est un type de couche de réseau neuronal dans lequel chaque neurone de la couche est connecté à tous les neurones de la couche précédente. Les couches entièrement connectées sont souvent utilisées en tant que dernière couche d'un réseau neuronal, car elles permettent au réseau de faire des prédictions ou des classifications en se basant sur toutes les caractéristiques apprises par les couches précédentes.

1.5 Travaux connexes

Pour situer notre travail dans son contexte, nous présentons d'abord quelques travaux antérieurs pertinents qui ont été réalisés dans le contexte de la reconnaissance d'activité humaine.

Dans [29], plusieurs modèles de l'apprentissage ont été étudiés, où les auteurs ont mené une comparaison entre différents algorithmes de classification tels que : DT, RF, SVM, LR. La comparaison a été faite en termes de taux de classification, F-mesure, rappel et la précision. Les résultats obtenus montrent que l'algorithme SVM offre les meilleures performances par rapport aux autres algorithmes de classification.

Une autre étude comparative a été faite par [30]. Les auteurs ont utilisé aussi une variété des modèles d'apprentissage automatique et une diversité des caractéristiques telles que la moyenne, la variance et la corrélation. Les résultats expérimentaux ont indiqué que les classificateurs NB et KNN ont des valeurs concurrentes.

Junhuai et al. [31] ont évalué l'algorithme multi-classe support vecteur machine à l'aide des critères de performance tels que la précision, le rappel et taux de classification correcte. Les auteurs ont conclu que cet algorithme a amélioré la performance de la prédiction.

L'étude [32] a impliqué 25 bénévoles, dont 15 ont été formés pour l'entraînement de données et le reste pour le test. Le système reçoit les données d'accéléromètre à partir des smartphones et montres intelligentes (smartwatch). La classification a été effectuée en utilisant le modèle KNN (avec $k=1$, $k=3$, $k=5$, $k=7$). La répartition des données entre l'entraînement et le test a été réalisée à l'aide de la méthode cross validation. Les meilleurs résultats ont été obtenus avec $k=3$.

Les auteurs de [33] ont évalué l'algorithme SVM. Cette étude était effectuée sur 5 étudiants (3 mâles, 2 femelles) le nœud de collecte de données est attaché à la taille de chaque participant pour reconnaître 06 activités qui sont : marcher, courir, monter, descendre, sauter et se lever. Pour l'évaluation des performances, plusieurs caractéristiques sont utilisées comme la moyenne, maximum, minimum et la corrélation. Le taux de classification obtenu était 96,2%.

D'autres chercheurs ont utilisé des approches d'apprentissage profond pour réaliser leurs systèmes comme le travail [34], où les auteurs ont proposé une comparaison entre 2 datasets UCI HAR et Pamap2 en appliquant les modèles CNN, LSTM et BiLSTM. Pour UCI HAR ont classifié 6 activités et 12 activités pour Pamap2. Ils ont utilisés 4 indicateurs d'évaluation : taux d'exactitude, précision, rappel et f-mesure. A travers les résultats obtenus on remarque que CNN atteint un taux d'exactitude le plus élevé et les performances du LSTM et de BiLSTM étaient similaires.

D'après avoir une vue globale sur les travaux antérieurs, quelques critères ont été sélectionnés pour comparer ces études précédents. Le tableau 1.1 présente une comparaison des travaux présentés.

référence	capteurs	dataset	activités	méthodes	Accuracy
[29]	accéléromètre gyroscope	UCI ML Repository	marcher, monter, descendre, assis, debout et allongé.	DT SVM RF LR	80,3% 96,33% 90,05% 96,01%
[30]	Smartphone (Accéléromètre gyroscope)	UCI	marcher, monter, descendre, debout	SVM RF KNN NB	90,63% 92,71% 97,92% 98,96%
[31]	accéléromètre gyroscope magnétomètre	UCI ML Repository	marche, assis, debout, descendre et monter.	SVM	98,57%
[32]	smartwatch accéléromètre	-	pompes, s'assois, squat saut.	KNN	76,67% 80% 96,67%
[33]	accéléromètre gyroscope	-	marcher, courir, sauter, descendre et monter.	SVM	92,5%
[34]	accéléromètre gyroscope magnétomètre	Pamap2 UCI HAR	s'allonger, s'assois, marcher, courir, cyclisme, monter, descendre, sauter, nettoyer, repasser.	CNN LSTM BLSTM	91% 92% 85% 89% 89% 89,4%

Tableau 1.1: Travaux recents sur HAR

D'après l'analyse effectuée sur les différents travaux de la littérature, nous pouvons constater que :

- La majorité des travaux ont utilisé des datasets collectées de UCI Machine Learning Repository grâce à la diversité des ensembles de données pour la reconnaissance d'activité humaine

- La méthode qui donne les meilleurs résultats est généralement Machine à Vecteur Support et pour l'apprentissage profond c'est CNN.

- L'apprentissage profond n'est pas largement utilisé dans la reconnaissance d'activité humaine. Pour cet effet, nous l'avons décidé de l'intégrer dans notre projet pour pouvoir comparer ses résultats avec les méthodes de l'apprentissage automatique décrites prochainement.

1.6 Conclusion

Dans ce chapitre, nous avons abordé divers aspects liés à la reconnaissance d'activité humaine. Nous avons commencé par présenter une classification des méthodes utilisées pour acquérir des données dans ce domaine. Ensuite, nous avons examiné les techniques d'apprentissage automatique et d'apprentissage profond appliquées à la reconnaissance d'activité humaine, en mettant l'accent sur les travaux connexes les plus récents.

Chapitre 2

Conception d'un système de reconnaissance d'activité humaine

2.1 Introduction

Notre projet s'inscrit dans un cadre de recherche visant à développer un modèle pour la reconnaissance d'activité humaine . Pour cela, nous avons mené une étude sur l'apprentissage automatique appliqué à HAR. Dans cette phase, nous commençons par présenter l'organisme d'accueil, puis nous abordons la conception du modèle ainsi que les algorithmes mis en jeu.

2.2 Présentation de l'organisme ATM MOBILIS

Filiale du Groupe Algérie Télécom, Mobilis est le premier opérateur mobile en Algérie, devenu autonome en août 2003, elle siège actuellement au Quartier d'Affaires, Bab Ezzouar, Alger.



Depuis sa création, Mobilis s'est fixé des objectifs principaux qui sont : la satisfaction client, la fidélité client, l'innovation et le progrès technologique, qui lui ont permis de faire des profits et d'acquérir près de 20 Million d'abonnés en un temps record.

Optant pour une politique de changement et d'innovation, Mobilis travaille en permanence sur son image de marque et veille constamment à offrir le meilleur à ses clients.

En déployant un réseau de haute qualité, en assurant un service client satisfaisant, et en créant des produits et services innovants, Mobilis est positionné comme étant un opérateur proche de ses partenaires et de ses clients, renforcé par sa signature institutionnelle : « Partout avec vous ». Son slogan est une promesse d'écoute et un signe de son engagement

à assumer son rôle dans le développement durable grâce à sa participation dans le progrès économique, son respect de la diversité culturelle, son engagement d'assumer son rôle social et sa participation à la protection de l'environnement.

Se munissant des valeurs : Transparence, Loyauté, Dynamisme et Innovation. Mobilis optimise sa qualité de service et veille à fidéliser ses clients.

« Mobilis, s'affirme une entreprise dynamique, innovante, loyale et transparente et continue sa compétition saine, dans un marché très concurrentiel ou le sérieux, la crédibilité et la proximité sont les clés de la réussite.

Comme le montre la figure au dessous 2.1 [35], L'organisation est structurée en deux grandes orientations générales, chacune étant composée de deux divisions qui rassemblent plusieurs directions. Pour notre stage, nous travaillons principalement au sein de la Direction de système d'information (DSI). Cette direction veut développer un système de reconnaissance d'activité humaine qui permet aux clients de suivre automatiquement leurs activités physiques quotidiennes, qualité de sommeil ou d'autres paramètres liés à la santé à travers de l'application.



Figure 2.1: Organigramme de Mobilis

2.3 Datasets

Le tableau ci-dessous 2.1 présente une sélection d'ensemble de données pertinents pour la reconnaissance d'activité humaine. Ces ensembles ont été soigneusement choisis pour leur diversité d'activités et de modalités de capture de données. Chaque jeu de données est décrit en termes de sa source, du nombre d'activités enregistrées et de l'approche adoptée.

Dataset	Description	Acquisition	Lien
Pampa2 (2012)	Contient 18 activités exécutés par 9 sujets	Capteur	https://archive-beta.ics.uci.edu/dataset/231/pamap2+physical+activity+monitoring
Opportunity (2009)	Contient 9 activités comprend des capteurs portable, d'objets et ambiants.	Capteur	https://archivebeta.ics.uci.edu/dataset/226/opportunity+activity+recognition
mHealth (2014)	Contient 12 activités pour 10 sujets	Capteur	https://www.kaggle.com/datasets/nimalsankalana/mhealth-dataset-data-set-csv
ActivityNet (2016)	Contient 200 types d'activités et un total de 849 heures de vidéos recueillies sur Youtube	Vision	https://paperswithcode.com/dataset/activitynet
WISDM (2011)	se compose de données recueillies auprès de 36 utilisateurs effectuant 6 activités	Capteur	https://www.kaggle.com/datasets/die9origephit/human-activity-recognition
Daily and sport Activities	Contient 19 activités est effectuée par 8 sujets	Capteur	https://www.kaggle.com/datasets/obirgul/daily-and-sports-activities
UCI-HAR (2013)	Contient 6 activités recueillies par 30 bénévoles	Capteur	https://huggingface.co/datasets/udayl/UCI_HAR
UCF101 (2012)	Se compose de 13320 clips vidéo pour 101 activités	Vision	https://paperswithcode.com/dataset/ucf101
Kinetics (2019)	Comprend environ 500 000 clips vidéo	Vision	https://paperswithcode.com/dataset/kinetics

Tableau 2.1: Tableau récapulatif sur les datasets

2.4 Contribution de notre travail

Le but de notre projet consiste à développer un modèle d'apprentissage automatique pour la reconnaissance d'activité humaine en appliquant l'approche basé sur des capteurs tels que l'accéléromètre et le gyroscope.

Nous avons également appliqué des techniques d'apprentissage profond pour améliorer les performances du modèle. Les résultats obtenus démontrent l'efficacité de notre approche et ouvrent la voie à de futures améliorations dans le domaine de la reconnaissance d'activité humaine.

2.5 Conception architecturale

La figure 2.2 représente une conception architecturale de la solution proposée pour la classification d'activités humaines, nous détaillons par la suite chaque module ainsi les algorithmes utilisés.

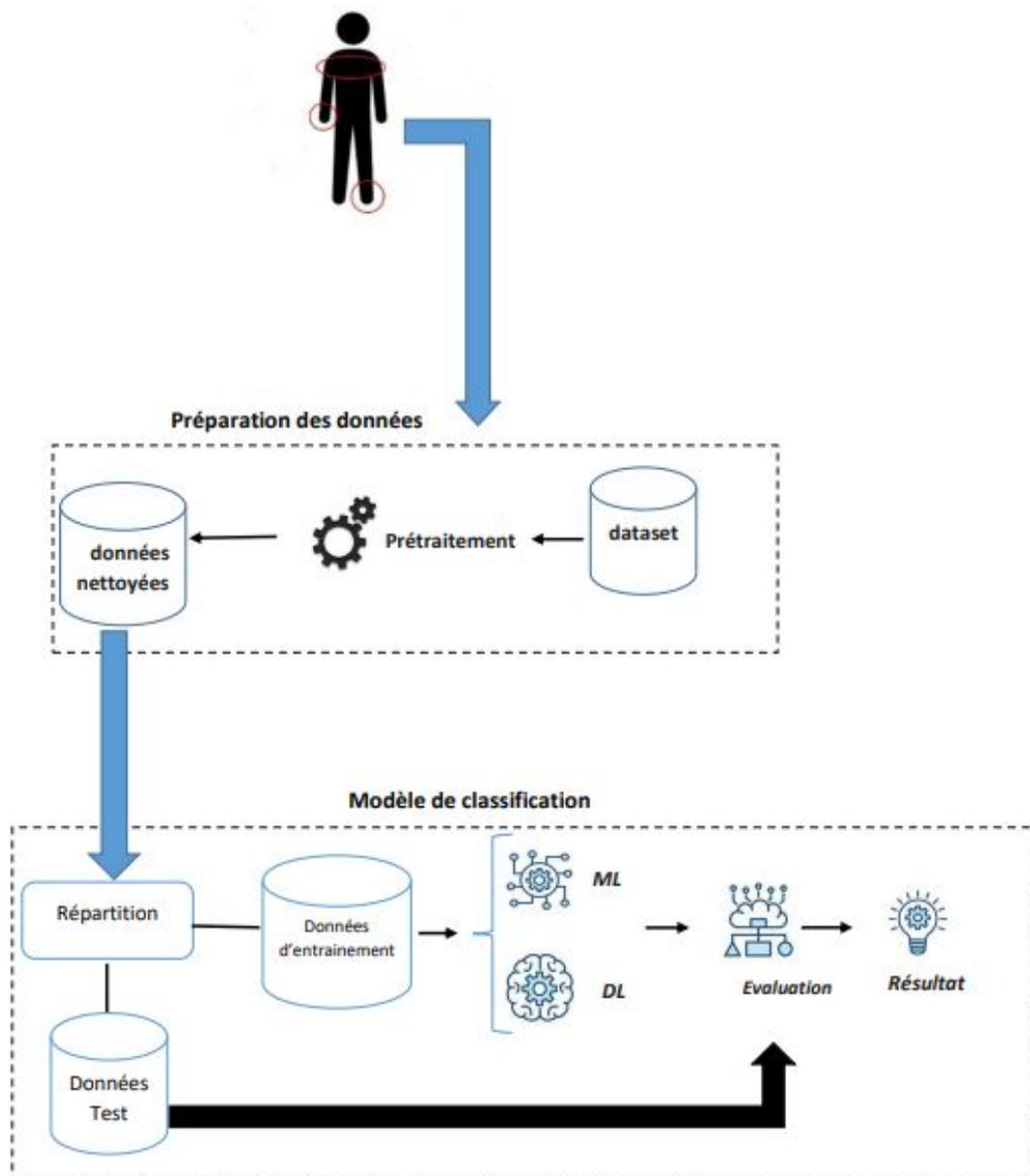


Figure 2.2: Architecture de la solution

2.6 Conception détaillée

Dans cette section, Nous mettons en avant les détails de la conception. L'objectif principal de cette phase est de fournir une vision claire et détaillée de la manière dont notre projet est réalisé.

2.6.1 Dataset utilisé

Pour l'évaluation des algorithmes cités auparavant, nous avons opté pour l'ensemble de données mHealth (Mobile Health) qui comprend des enregistrements de mouvements corporels pour dix bénévoles de divers profils tout en effectuant 12 activités physiques : se lever, s'asseoir, marcher, monter, courir, sauter devant et arrière, s'allonger, pédaler (cyclisme), s'accroupir, se pencher, footing, lever les bras vers l'avant. Des capteurs placés sur la poitrine, le poignet droit et la cheville gauche du sujet, comme illustre la figure 2.3, sont utilisés pour mesurer le mouvement de diverses parties du corps, à savoir l'accélération, la vitesse de rotation et l'orientation du champ magnétique. Toutes les modalités de détection sont enregistrées à un taux d'échantillonnage de 50 Hz, ce qui est considéré comme suffisant pour capturer l'activité humaine.

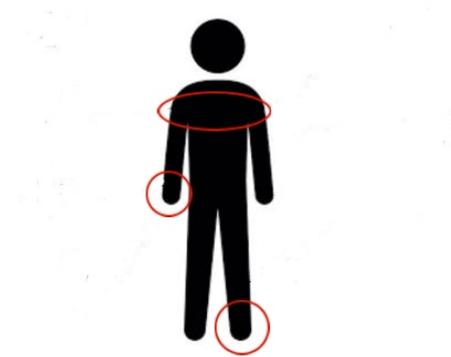


Figure 2.3: Position des capteurs

Les caractéristiques extraites de l'expérience ont été dérivées selon un processus élaboré et sont présentées dans le tableau 2.2

Caractéristique	Explication
acc_ch_XYZ	accélération du capteur thoracique(poitrine) sur les axes x,y,z
elec_1	signal d'électrocardiogramme (ECG) 1
elec_2	signal d'électrocardiogramme (ECG) 2
acc_la_XYZ	accélération du capteur de la cheville gauche sur les axes x,y,z
gyro_la_XYZ	gyroscope du capteur de la cheville gauche sur les axes x,y,z
mag_la_XYZ	magnétomètre du capteur de la cheville gauche sur les axes x,y,z
acc_rla_XYZ	accélération à partir du capteur du poignet droit sur les axes x,y,z
gyro_rla_XYZ	gyroscope du capteur du poignet droit sur x,y,z
mag_rla_XYZ	magnétomètre du capteur du poignet droit sur x,y,z

Tableau 2.2: Description des caractéristiques de jeux de données mHealth. [36]

2.6.1.1 Partitionnement des données

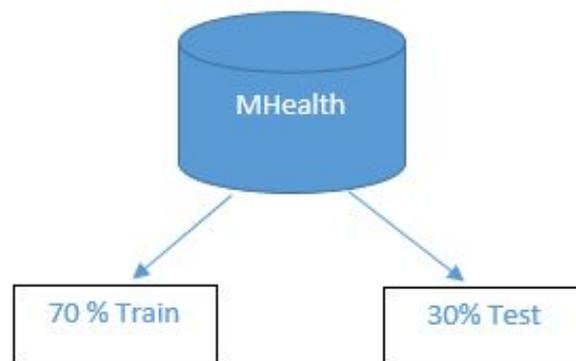


Figure 2.4: Partitionnement de dataset

L'ensemble de données obtenu a été partitionné en ensembles de données d'apprentissage et de test. Chaque classifieur sera entraîné sur l'ensemble de données d'apprentissage puis évalué par rapport à l'ensemble de test. La répartition des données utilisée dans ce travail est de 70% et 30% pour les ensembles d'apprentissage et de test respectivement comme le montre la figure 2.4.

2.6.2 Classification avec des modèles ML

Dans cette phase, l'objectif est d'entraîner les algorithmes de classification afin de créer des modèles sous forme de règles qui seront utilisés ultérieurement pour classer de nouveaux échantillons. Nous examinerons maintenant en détail les divers algorithmes d'apprentissage automatique employés pour générer ces modèles

1. **Machine à Vecteur Support** : Dans cette approche nous avons utilisé machine à vecteur de support. Les SVM sont des modèles d'apprentissage supervisé qui peuvent être employés pour effectuer des tâches de classification. Ils sont particulièrement adaptés à la reconnaissance d'activité humaine car ils sont capables de séparer efficacement les différentes classes d'activités dans un espace multidimensionnel. Nous avons procédé à l'entraînement de SVM en utilisant la bibliothèque scikit-learn et le classifieur SVC. Cette dernière offre des fonctionnalités complètes permettant d'entraîner des SVM avec divers noyaux, d'ajuster les hyper-paramètres et d'évaluer les performances du modèle.
2. **K plus proches voisins** : Nous avons appliqué l'algorithme K plus proches voisins pour la tâche de la reconnaissance d'activité humaine. Cet algorithme a été généré avec le classifieur `KNeighborsClassifier()`. Le choix du paramètre pour le nombre de voisins dans l'algorithme des k plus proches voisins n'est pas régi par des règles ou critères généraux. Cela nécessite donc de mener plusieurs expérimentations afin de déterminer le nombre optimal de voisins à considérer. Pour cela nous avons testé avec $k=3$ et $k=5$ le meilleur résultat a été obtenu par $k=3$.
3. **Arbre de décision** : Nous avons également utilisé l'arbre de décision dans le processus de reconnaissance d'activité humaine. En utilisant la fonction `DecisionTreeClassifier()` de Scikit-learn.
4. **Forêt aléatoire** : est un algorithme d'apprentissage supervisé qui combine plusieurs arbres de décision pour effectuer la classification. Après la préparation des données d'entraînement et de test, nous avons créé ce classifieur en utilisant la classe `RandomForestClassifier` de scikit-learn. Lors de la prédiction d'une nouvelle activité, le modèle combine les prédictions de chaque arbre individuel, ce qui permet d'obtenir une prédiction globale plus fiable.
5. **Combinaison** : Ces dernières années, plusieurs travaux ont prouvé la supériorité des systèmes ensemblistes par rapport à l'utilisation d'un seul modèle [37] [38]. Dans un système de combinaison basé sur le vote, chaque membre de l'ensemble participe de manière indépendante et les sorties de tous les classifieurs sont combinées dans une dernière étape à une décision unique. La prédiction de chaque candidat

de l'ensemble D_i , pour un exemple X est interprétée comme un vote pour l'une des classes possibles. La classe avec la majorité des votes est retenue comme décision finale [39]. La figure 2.5 résume cette explication.

Dans notre projet, Nous avons appliqué la combinaison des 3 modèles pour bénéficier de leurs forces respectives. Ces modèles sont entraînés sur les mêmes données d'apprentissage.

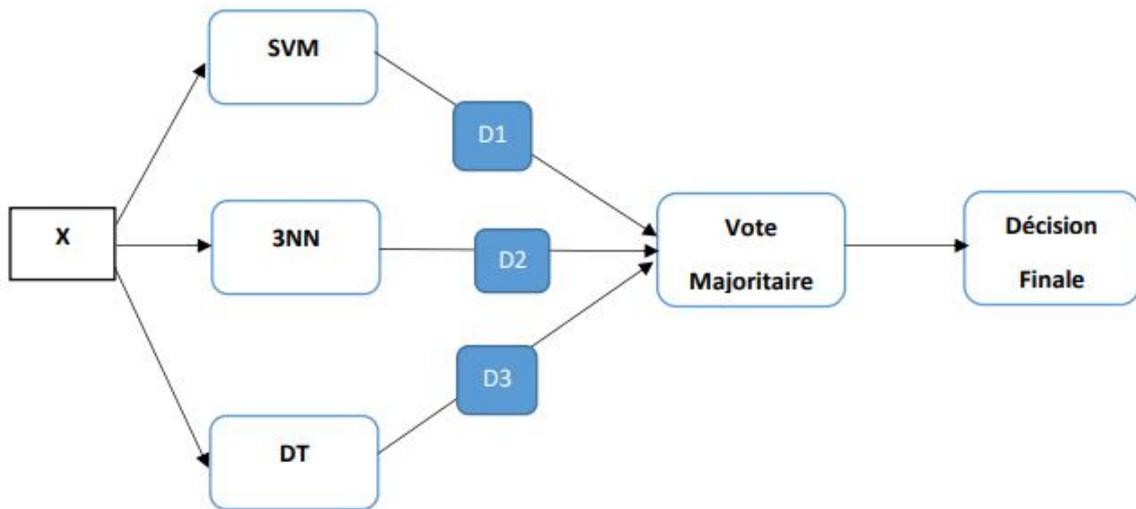


Figure 2.5: Architecture globale de la combinaison.

2.6.3 Classification avec des modèles DL

Pour la classification en utilisant le Deep Learning, nous avons utilisé les APIs Keras et TensorFlow pour développer et entraîner les modèles, une fois que le modèle est construit, nous procédons à l'entraînement en spécifiant l'optimiseur "adam", la fonction de perte "categorical_crossentropy" et la métrique d'évaluation "accuracy", avec un taux d'entraînement de 80%. Ensuite, nous lançons la boucle d'entraînement sur 10 époques en utilisant les données d'entraînement préalablement préparées pour tous les modèles que nous l'avons.

1. CNN :

Dans notre étude, nous avons examiné l'utilisation des réseaux de neurones convolutifs (CNN). une architecture très populaire grâce à son efficacité dans la tâche de classification et à son capacité d'extraction automatiquement des caractéristiques pertinentes à partir des données d'entrée. Le modèle a été construit en utilisant les couches suivantes (voir la figure 2.6) :

- Une couche Input qui prend en compte le nombre de pas de temps et le nombre de caractéristiques de nos données.
- Deux couches Convolution1D pour une représentation hiérarchique des caractéristiques et une augmentation de sa capacité.
- Une couche Dropout pour réduire le surapprentissage.
- Une couche MaxPooling1D pour réduire la dimensionnalité.
- Une couche Flatten pour aplatir les sorties de la couche de mise en commun.
- Des couches entièrement connectées sont la couche Dense avec 128 neurones.
- Une couche Dense pour la classification multi-classes.

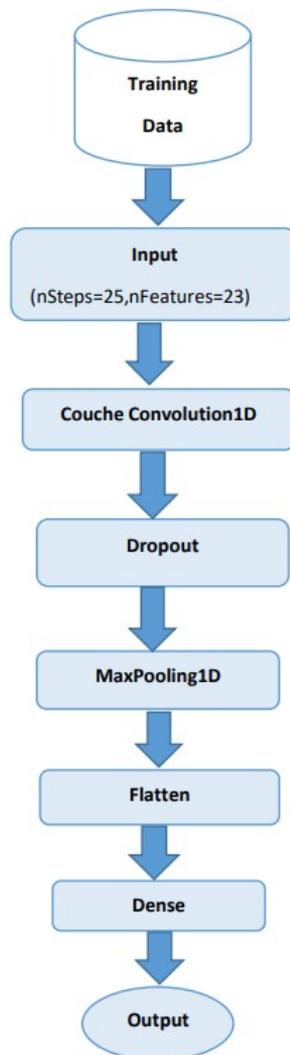


Figure 2.6: Schéma d'architecture du réseau CNN utilisé

2. LSTM :

LSTM est une architecture de réseau de neurones récurrents (RNN) capable de capturer et de mémoriser les dépendances à long terme dans les séquences, ce qui en fait une méthode efficace pour le traitement des données séquentielles, y compris la classification. Dans l'architecture du modèle, nous avons utilisé les APIs keras et tensorflow et comme couches nous avons :

- Une couche d'entrée (Input) qui est définie en prenant en compte le nombre de pas de temps et le nombre de caractéristiques de nos données. Cela permet au modèle de savoir comment traiter et interpréter les séquences temporelles avec les caractéristiques spécifiées.
- Une couche LSTM pour prendre en compte les dépendances à long terme dans les séquences d'entrée.
- Une couche Dropout Pour régulariser le modèle et éviter le surapprentissage.
- Une couche Flatten est utilisée pour transformer les sorties de la couche Dropout en un vecteur unidimensionnel à être connectées à la couche suivante.
- Une couche Dense a été ajoutée pour introduire de la non-linéarité dans le modèle.
- Une couche de sortie (outputs) pour la classification multi-classes.

Enfin le modèle a été compilé.

3. BiLSTM :

Le BiLSTM est une architecture de réseau de neurones récurrents qui intègre à la fois un flux d'informations avant et arrière pour capturer les dépendances bidirectionnelles dans les données séquentielles. L'architecture est similaire à celle de LSTM, elle comprend les couches suivantes : – Une couche d'entrée qui prend en compte la forme des données d'entrée.

- Deux couches LSTM bidirectionnelles sont utilisées pour capturer les dépendances bidirectionnelles.
- Des couches supplémentaires pour améliorer la régularisation, une couche Dropout est ajoutée avec Dropout(0.5). Ensuite, une couche Flatten est utilisée pour aplatir les sorties de la couche LSTM avec Flatten().
- Une couche Dense est ajoutée pour introduire des connexions entièrement connectées les neurones et la fonction d'activation.
- une couche Dense est utilisée comme couche de sortie.

2.6.4 Module d'évaluation

Il n'est pas toujours simple de choisir la meilleure métrique. La performance des algorithmes varie en fonction des activités prises en compte et du jeu de données utilisé. L'évaluation de la qualité d'un système de classification nécessite une certaine comparaison basée sur des métriques de mesure de performance.

- **matrice de confusion** : connue sous le nom de tableau de confusion, est une représentation des performances d'un modèle de classification. Elle est utilisée pour évaluer la précision d'un modèle en comparant ses prédictions qu'il fait avec les valeurs de l'échantillon de réelles.

Pour le cas d'une classification multi-classes où chaque ligne de la matrice représente la classe réelle des données, tandis que chaque colonne représente la classe prédite par le modèle.

La structure de base d'une matrice de confusion est la suivante :

	Prédit positif	Prédit négatif
Réel positif	Vrai positif (VP)	Faux négatif (FN)
Réel négatif	Faux positif (FP)	Vrai négatif (VN)

Tableau 2.3: Exemple d'une matrice de confusion [40]

- **F-mesure** : F-mesure permet de combiner la précision et le rappel en une seule métrique, est calculé comme suit :

$$\text{F-mesure} = \frac{2 * \text{Precision} * \text{Rappel}}{\text{Precision} + \text{Rappel}} \quad [41]$$

- **Taux d'exactitude (Accuracy)** : L'exactitude est la mesure primaire pour évaluer nos modèles, elle est calculée le nombre de bonnes réponses prédictions par le nombre total de prédictions. Sa formule est :

$$\text{Taux d'exactitude} = \frac{VP + VN}{VP + VN + FP + FN} \quad [42]$$

- **Rappel** : le rappel est le nombre d'exemples identifiés positivement correctement divisé par le nombre total positifs réellement . sa définition est :

$$\text{Rappel} = \frac{VP}{VP + FN} \quad [34]$$

- **Précision** : La précision peut être considérée comme une mesure de la qualité ou

de l'exactitude. C'est la fraction des instances pertinentes c'est à dire le nombre d'exemples positifs sur le nombre d'exemples étiquetés comme positifs.

$$\text{Précision} = \frac{VP}{VP+FP} \quad [43]$$

2.7 Conclusion

Nous avons présenté dans ce chapitre la conception de notre système, les méthodes utilisées et les objectifs visés. Nous avons exposé les étapes de conception qui nous permettront de réaliser une solution pour aborder notre sujet. Ensuite, nous avons décrit le choix des algorithmes d'apprentissage automatique, les architectures des modèles d'apprentissage profond, ainsi que les métriques d'évaluation utilisées.

Dans le prochain chapitre, nous présenterons les démarches d'implémentation de notre système ainsi que les résultats obtenus.

Chapitre 3

Expérimentation et Discussion

3.1 Introduction

Après avoir présenté notre solution de manière architecturale dans le chapitre précédent, nous présentons maintenant les technologies et les outils adoptés pour l'implémentation, ainsi que le pré-traitement effectué sur la base de données. Nous exposons ensuite les résultats des différentes méthodes de classification étudiées en réalisant une étude de comparaison entre eux afin d'en analyser les performances.

3.2 Environnement de travail

Dans cette partie nous présentons les différents outils utilisés pour l'implémentation de notre système.

- **Google Colab** : ou Colaboratory est un service cloud gratuit, offert par Google, basé sur Jupyter Notebook et destiné à la formation et la recherche en l'apprentissage automatique. Cette plateforme permet d'entraîner des modèles de Machine Learning directement dans le cloud. Sans donc avoir besoin d'installer quoi que ce soit sur notre ordinateur, à l'exception d'un navigateur.

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

- **Scikit-learn** : est une bibliothèque libre Python destinée l'apprentissage automatique. Elle est développée par de nombreux contributeurs notamment dans le monde académique

par des instituts français d'enseignement supérieur et de recherche comme Inria.

– **Nump** : est une bibliothèque permettant d'effectuer des calculs numériques avec Python. Elle introduit une gestion plus facile des tableaux de nombres et des fonctions.

– **Matplotlib** : est une bibliothèque de traçage pour le langage de programmation Python. Elle fournit une API orientée objet pour intégrer des graphiques dans des applications utilisant des kits d'outils d'interface utilisateur graphique.

3.3 Implémentation de la solution

3.3.1 Prétraitement

Rappelons que le dataset utilisé intitulé Mhealth, les données ont été collectées auprès de dix personnes effectuant 12 activités. Lors de l'exploration, nous avons remarqué que l'ensemble présente un problème des données déséquilibrées. Une situation déséquilibrée se produit lorsque le nombre des exemples d'une classe est nettement plus grand que celui des autres classes. La figure 3.1 présente une illustration claire de la distribution des exemples de chaque activité.

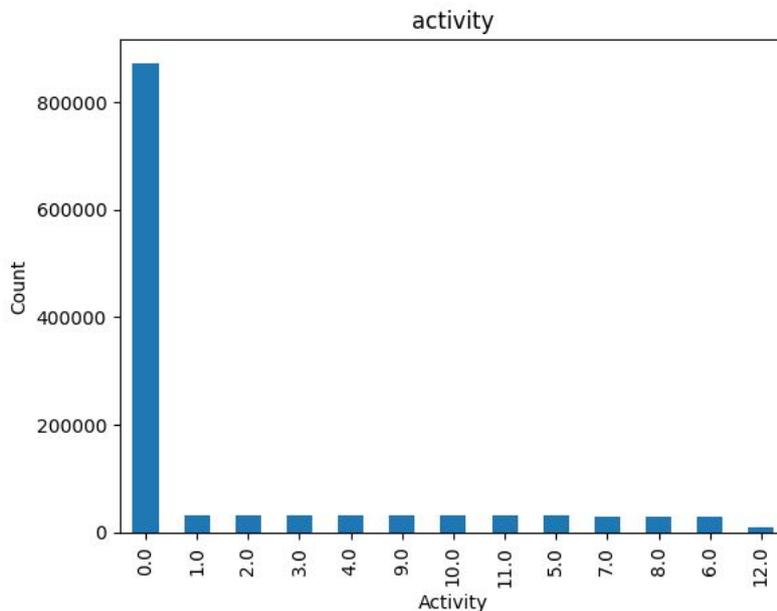


Figure 3.1: Illustration des proportions des exemples pour chaque activité.

Lorsque les données d'entraînement sont déséquilibrées, les modèles de classification privilégient généralement les classe majoritaires, et donc échouent de classer correctement les exemple des classes minoritaires, ce qui dégrade la performance de classification [44].

De ce fait, nous avons appliqué un ré-échantillonnage (resampling) avec deux méthodes : random resampling et undersampling (sous-échantillonnage). dans le cas de sous-échantillonnage, la classe majoritaire est réduite ou "sous-échantillonnée" pour correspondre au nombre d'instances dans la ou les classes minoritaires. Ceci est fait en sélectionnant un sous-ensemble d'instances de la classe majoritaire afin que la distribution de classe devienne plus équilibrée.

Au final nous avons choisi la deuxième méthode car elle a bien équilibré les classes comme le montre cette figure 3.2.

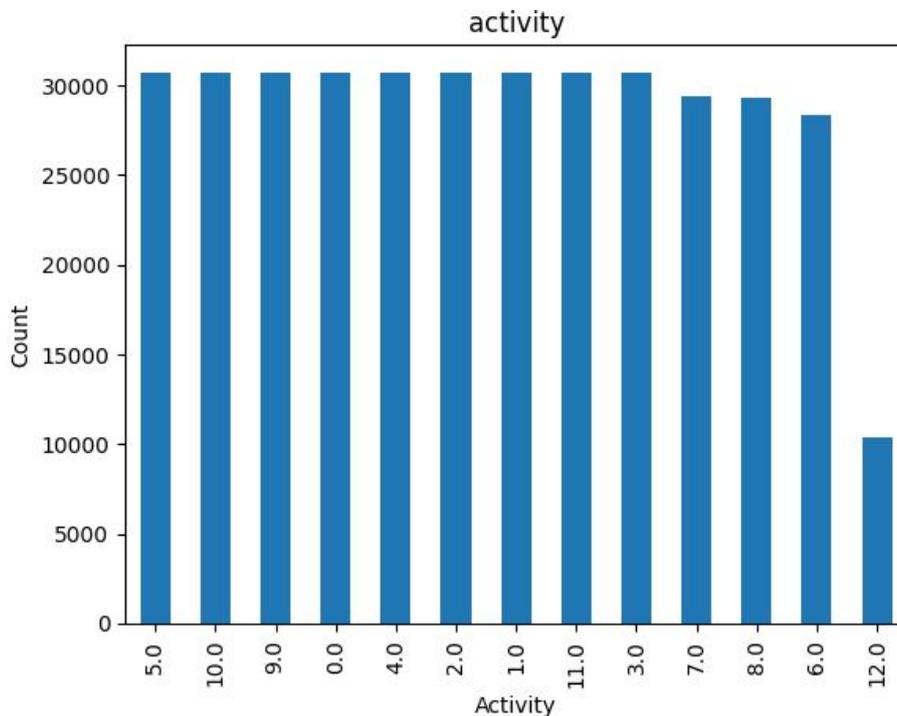


Figure 3.2: Visualisation des proportions des exemples de chaque activité après l'équilibrage.

En ce qui concerne le pré-traitement des données pour l'entraînement des modèles deep learning, nous avons ajouté des étapes supplémentaires plus qu'au prétraitement du machine learning qui sont :

- Mise à l'échelle de l'ensemble complet des prédicteurs : Nous utilisons la méthode `StandardScaler()` pour mettre les valeurs des prédicteurs à une échelle commune. Cela permet de normaliser leurs valeurs afin qu'elles se situent dans une plage commune, favorisant ainsi la stabilité et la convergence du modèle.

- Application d'une fonction de transformation de séquence : Nous appliquons une fonction de transformation de séquence pour adapter les données à un format spécifique. Cela peut inclure le découpage des données en séquences de longueur fixe ou l'ajout de certaines transformations spécifiques aux séquences.

– Division de l’ensemble d’entraînement en séquences : Nous divisons également l’ensemble d’entraînement en séquences pour faciliter l’apprentissage du modèle. Par exemple, nous pouvons utiliser une longueur de séquence fixe de n pas, comme $n = 25$, pour former des séquences cohérentes à partir des données.

– Conversion des variables de sortie en catégorielles : Pour les problèmes de classification, nous convertissons les variables de sortie en variables catégorielles. Cela permet au modèle de comprendre les différentes classes et d’effectuer des prédictions catégoriques plutôt que continues.

– Encodage One-Hot des étiquettes entières : Nous convertissons les étiquettes entières en une représentation encodée One-Hot. Cette représentation permet d’attribuer un vecteur binaire unique à chaque classe, facilitant ainsi l’apprentissage et l’évaluation du modèle.

3.3.2 Entraînement des modèles

La stratégie d’entraînement utilisée pour chaque classificateur (RF, SVM, KNN,DT) est expliquée comme suit :

Le dataset Mhealth est divisé en deux parties en utilisant la fonction `train_test_split`. Nous avons entraîné les modèles avec un paramètre `test_size` de 0.3, ce qui nous a donné 261740 données d’entraînement et 112175 données de test.

Les modèles sont créés et définis à l’aide des fonctions prédéfinies. Pour la combinaison des modèles : SVM, KNN, DT nous avons utilisé le `VotingClassifier` pour créer un classificateur par vote. Les modèles sont entraînés en utilisant la fonction `fit`.

Pour le deep learning (CNN, LSTM, BiLSTM), nous avons suivi la procédure suivante : Nous avons commencé par diviser le dataset en utilisant une division différente. Cette fois-ci, nous avons utilisé un pourcentage différent et un `test_size` de 0.2.

Nous avons obtenu des formes indiquant que `X_train` et `X_test` sont des tenseurs 3D, avec une dimension de 299108 pour les échantillons d’entraînement et 74759 pour les échantillons de test. Chaque échantillon est une séquence de longueur 25 avec 23 caractéristiques.

Après nous avons converti les variables de sortie en utilisant une représentation catégorielle encodée, en transformant les étiquettes entières en une représentation correspondante avec la fonction `to_categorical()`.

Les modèles de deep learning (CNN, LSTM, BiLSTM) sont construits en utilisant différentes couches citées précédemment qui jouent un rôle clé dans l’apprentissage et la représentation des données. Ces couches permettent aux modèles d’extraire des caractéristiques significatives et de capturer les dépendances complexes présentes dans les données.

Couches de convolution : Les modèles CNN utilisent des couches de convolution pour

effectuer des opérations de convolution sur les données d'entrée.

Couches récurrentes : Les modèles LSTM et BiLSTM utilisent des couches récurrentes pour capturer les dépendances séquentielles dans les données. Ces couches sont conçues pour traiter des données séquentielles.

Les prédictions sont effectuées sur les données de test, puis évaluées en utilisant le taux d'exactitude, la précision, le rappel, la F-mesure et la matrice de confusion. Les résultats de cette évaluation seront présentés dans la prochaine section.

la précision et la perte en fonction de l'époque sont illustrés dans la figure 3.3.

```
Epoch 1/10
9348/9348 [=====] - 750s 79ms/step - loss: 0.5142 - accuracy: 0.8326
Epoch 2/10
9348/9348 [=====] - 724s 77ms/step - loss: 0.2439 - accuracy: 0.9255
Epoch 3/10
9348/9348 [=====] - 719s 77ms/step - loss: 0.1946 - accuracy: 0.9422
Epoch 4/10
9348/9348 [=====] - 711s 76ms/step - loss: 0.1704 - accuracy: 0.9507
Epoch 5/10
9348/9348 [=====] - 713s 76ms/step - loss: 0.1556 - accuracy: 0.9556
Epoch 6/10
9348/9348 [=====] - 710s 76ms/step - loss: 0.1444 - accuracy: 0.9589
Epoch 7/10
9348/9348 [=====] - 708s 76ms/step - loss: 0.1364 - accuracy: 0.9612
Epoch 8/10
9348/9348 [=====] - 706s 76ms/step - loss: 0.1289 - accuracy: 0.9642
Epoch 9/10
9348/9348 [=====] - 708s 76ms/step - loss: 0.1235 - accuracy: 0.9656
Epoch 10/10
9348/9348 [=====] - 711s 76ms/step - loss: 0.1196 - accuracy: 0.9670
```

Figure 3.3: L'historique de Colab durant la phase d'apprentissage.

3.4 Résultats et discussions

Dans ce qui suit nous présentons les résultats obtenus pour évaluer les performances des méthodes de classification.

3.4.1 Résultats des modèles ML

Algorithme	Taux de classification (%)	Précision (%)	rappel (%)	F-mesure (%)
SVM	84,33	83,68	83,53	83,14
KNN(k=3)	86,29	85,80	84,34	84,50
KNN(k=5)	85,88	85,36	83,83	83,75
DT	93,01	92,65	92,52	92,58
RF	97,99	97,93	98	97,88
Combinaison	90,96	90,91	89,80	90,12

Tableau 3.1: Scores obtenus par les algorithmes de ML

Le tableau 3.1 représente les résultats des algorithmes SVM, KNN, DT, RF ainsi que la combinaison des trois approches individuelles, en termes de performances.

Pour l'algorithme SVM, les résultats montrent une exactitude de 84,33%, une précision de 83,68 %, un rappel de 83,53% et une F-mesure de 83,14%. Ces valeurs indiquent que SVM obtient de bonnes performances globales, mais pas les meilleures parmi les algorithmes évalués.

En ce qui concerne KNN avec un voisinage de $k=3$, on observe une amélioration des performances avec une exactitude de 86,29%, une précision de 85,80%, un rappel de 84,34% et une F-mesure de 84,50%. Cela suggère que KNN avec $k=3$ a tendance à mieux classer les données que SVM. Lorsque le voisinage de KNN est augmenté à $k=5$, les performances diminuent légèrement avec une exactitude de 85,88%, une précision de 85,30%, un rappel de 83,83% et une mesure F de 83,75%. Cependant, ces valeurs restent relativement élevées, montrant que KNN avec $k=5$ est toujours compétitif en termes de performances.

Le Random Forest (RF) se démarque avec des performances impressionnantes, affichant une exactitude de 97.99%, une précision de 97.93%, un rappel de 98% et une F-mesure de 97,78%. Ces résultats suggèrent que RF est très efficace dans la classification des données, dépassant les autres algorithmes évalués.

l'arbre de décision (DT) montre les performances suivants : taux d'exactitude 93,01% ,une précision de 92,65% , un rappel de 92,52% et une F-mesure 92,58%.

Enfin, la combinaison des trois algorithmes (SVM, KNN et DT) montre des performances solides avec une exactitude de 90,96%, une précision de 90,91%, un rappel de 89,80% et une F-mesure de 90,12%. Cela indique que la combinaison des différents algorithmes a permis d'améliorer les performances par rapport à chaque algorithme pris individuellement.

Les matrices de confusion pour les modèles ML :

Dans cette partie, nous présentons les matrices de confusion pour les classifieurs machine à vecteur support, arbre de décision, forêt aléatoire, K plus proches voisins ainsi que la combinaison des trois (SVM, DT, KNN).

Le tableau 3.2 montre une description des activités selon les numéros (0,12) :

Numéro	Signification
0	aucune activité
1	se lever (debout)
2	s'asseoir
3	allonger
4	marcher
5	monter
6	se pencher
7	lever les bras
8	s'accroupir
9	pédaler (cyclisme)
10	footing
11	courir
12	sauter en avant et en arrière

Tableau 3.2: Tableau récapitulatif des activités

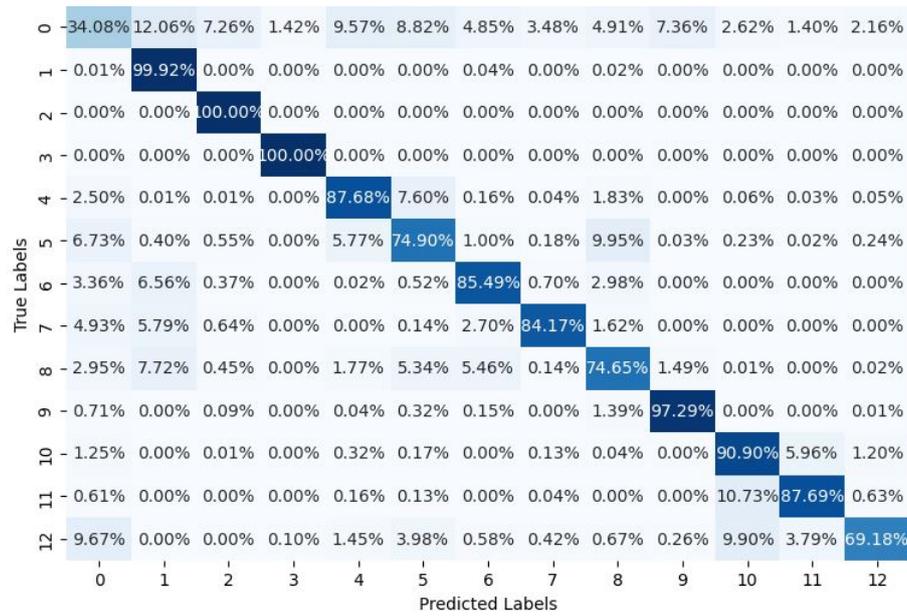


Figure 3.4: Matrice de confusion du SVM

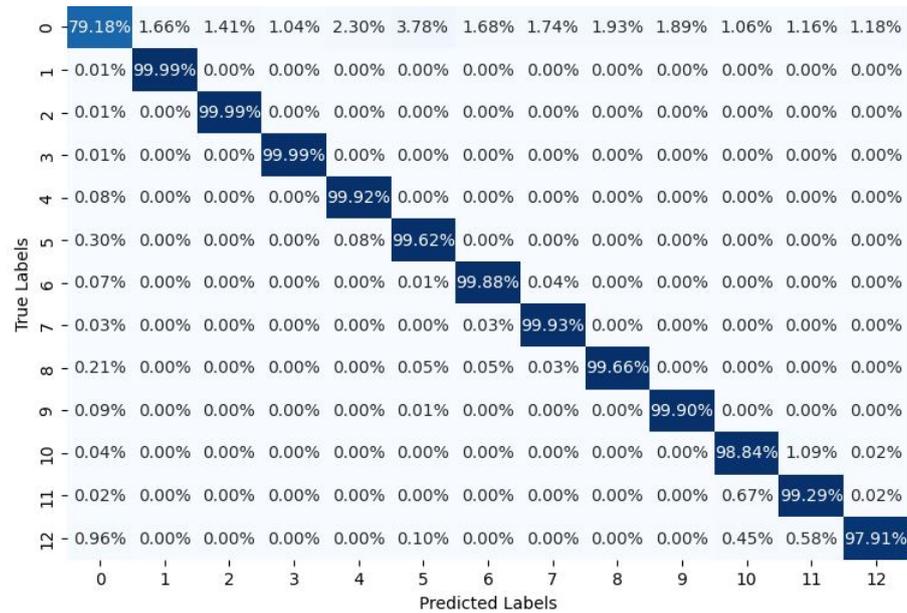


Figure 3.5: Matrice de confusion du RF

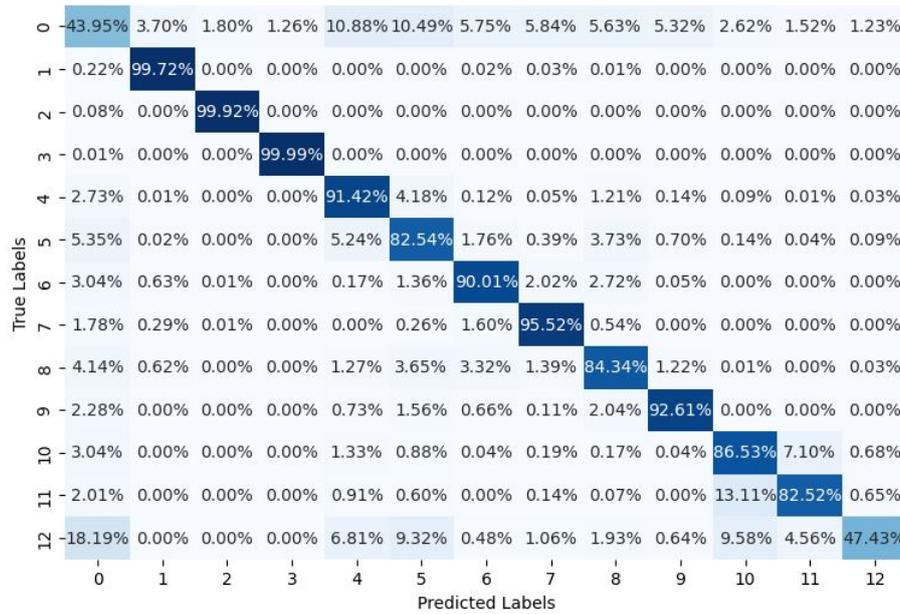


Figure 3.6: Matrice de confusion du 3NN

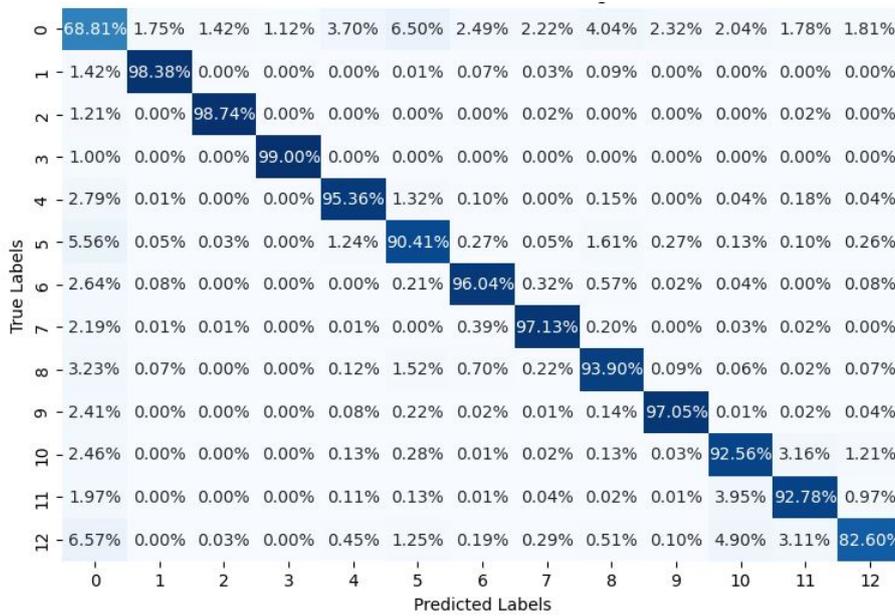


Figure 3.7: Matrice de confusion d'arbre de décision

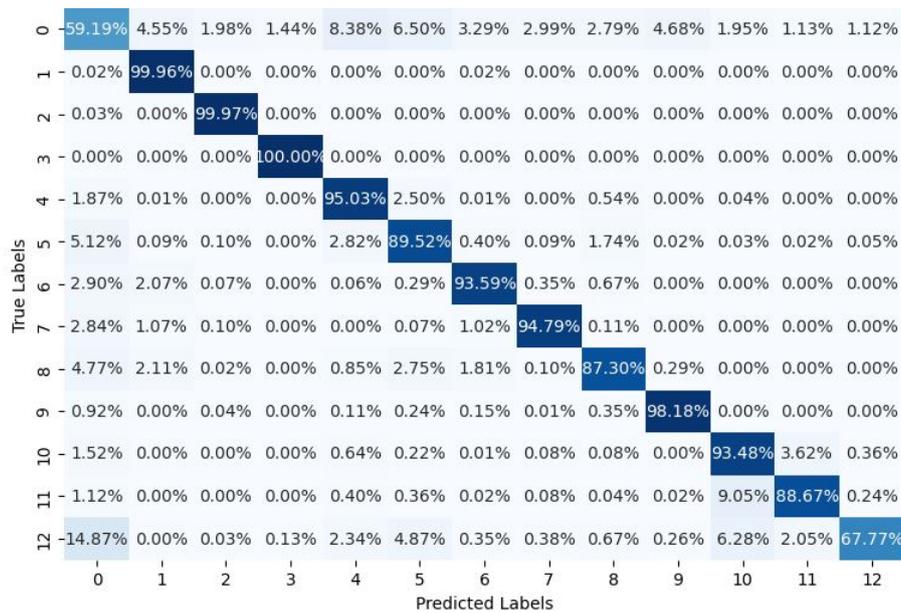


Figure 3.8: Matrice de confusion de la combinaison (SVM, DT, 3NN)

Les matrices de confusion sont un outil essentiel pour évaluer les performances des modèles d'apprentissage automatique, en particulier dans les problèmes de classification. Elles fournissent une représentation visuelle des résultats de classification en comparant les prédictions du modèle avec les vraies étiquettes des données.

1. Performance du SVM :

- Le SVM semble bien performer pour les activités 2 (s'asseoir) et 3 (allonger) avec un taux de 100%.

- Cependant, il a du mal avec l'activité 0, avec un taux de seulement 34.08%.

2. Performance du Random Forest (RF) :

- Le RF obtient de très bons résultats pour les activités 1 (se lever), 2 (s'asseoir) et 3 (allonger) avec un taux de 99.99%.

- L'activité 0 présente toujours des difficultés avec un taux de 79.18%.

3. Performance du 3NN :

- Le 3NN performe bien pour l'activité 3 (allonger) avec un taux de 99.99%.

- Cependant, l'activité 0 a un taux assez bas de 43.95%.

4. Performance de l'arbre de décision :

- L'arbre de décision a une bonne performance pour l'activité 3 avec un taux de 99%.

- Comme les autres modèles, il a du mal avec l'activité 0, avec un taux de 68.81%.

5. Modèle combiné (SVM, 3NN et arbre de décision) :

- La combinaison de modèles semble améliorer la performance globale, en particulier pour les activités 2 (s'asseoir), 1 (se lever) et 3 (allonger).

- L'activité 0 présente toujours des difficultés, mais le taux a augmenté à 59.19%.

En résumé, les modèles semblent bien classer les activités 1 (se lever), 2 (s’asseoir) et 3 (allonger) avec des taux élevés. L’activité 0 semble être la plus difficile à classer pour tous les modèles, montrant ainsi une classe déséquilibrée ou des caractéristiques problématiques. Et pour les résultats des faux positifs et des faux négatifs dans les matrices montrent une tendance vers des valeurs relativement faibles, ce qui indique que le modèle commet moins d’erreurs de classification de ces types.

3.4.2 Résultats des modèles DL

Algorithme	Taux d’exactitude (%)	Précision (%)	Rappel (%)	F-mesure (%)
CNN	91,16	91,53	91,16	90,49
LSTM	93,87	93,93	93,90	93,26
BiLSTM	96,5	96,47	96,5	96,3

Tableau 3.3: Evaluation des algorithmes de DL

Les résultats décrites dans le tableau 3.3 démontrent des performances significatives pour les trois modèles de deep learning évalués : CNN, LSTM et BiLSTM.

Le modèle CNN a atteint un taux d’exactitude de 91,16%, une précision de 91,53%, un rappel de 91,16% et une F-mesure de 90,49%. Ces résultats indiquent que le modèle CNN a été capable de classifier avec précision les données d’entrée, obtenant de bons scores dans les différentes mesures d’évaluation.

Le modèle LSTM a présenté des performances encore meilleures, avec un taux d’exactitude de 93,87%, une précision de 93,93%, un rappel de 93,90% et une F-mesure de 93,26%. Ces résultats soulignent l’efficacité du modèle LSTM dans la capture des dépendances à long terme dans les séquences temporelles, ce qui lui permet de réaliser des classifications précises.

Le modèle BiLSTM a obtenu les meilleurs résultats parmi les trois modèles évalués, avec un taux d’exactitude de 96,5%, une précision de 96,47%, un rappel de 96,5% et une F-mesure de 96,3%. Ces résultats mettent en évidence les avantages de l’utilisation de la bidirectionnalité dans le modèle BiLSTM, qui lui permet d’exploiter à la fois les informations passées et futures pour une meilleure compréhension des séquences et des classifications plus précises.

La comparaison des performances entre les modèles révèle que le modèle BiLSTM a surpassé les modèles CNN et LSTM en termes de précision, de rappel, de taux d’exactitude et de F-mesure. Cela suggère que l’incorporation de la bidirectionnalité dans le modèle a permis d’améliorer significativement sa capacité de classification.

Cependant, il convient de noter que le modèle BiLSTM a également nécessité une plus grande complexité et une plus grande consommation de ressources en comparaison avec les autres modèles. Cela soulève la question de l'équilibre entre la performance et l'efficacité du modèle, et de la nécessité d'adapter l'approche en fonction des contraintes spécifiques de chaque application.

En conclusion, les résultats obtenus démontrent l'efficacité des modèles de deep learning, en particulier du modèle BiLSTM, dans la classification des données.

Dans nos expérimentations, nous avons examiné l'impact des fonctions d'activation sur les performances des modèles. Nous avons spécifiquement comparé l'utilisation de la fonction d'activation tanh, qui était utilisée précédemment, avec la fonction d'activation ReLU voici les résultats :

le modèle BiLSTM a obtenu un taux d'exactitude de 95,75%, une précision de 95,65%, un rappel de 95,75% et une F-mesure de 95,60%. Pour le modèle LSTM, les résultats étaient légèrement inférieurs, avec un taux d'exactitude de 89,84%, une précision de 91,18%, un rappel de 89,84% et une F-mesure de 89,12%. Enfin, le modèle CNN a atteint un taux d'exactitude de 91,06%, une précision de 91,14%, un rappel de 91,06% et une F-mesure de 90,27%.

Les résultats de nos expérimentations indiquent que l'utilisation de la fonction d'activation ReLU a généralement conduit à de bonnes performances pour les trois modèles, mais la fonction d'activation tanh a donné des meilleurs résultats dans ce type de données.

Cependant, le choix de la fonction d'activation peut avoir un impact significatif sur les performances des modèles d'apprentissage automatique, et il est important de considérer différentes options et d'expérimenter pour trouver la configuration optimale.

En conclusion, nos résultats suggèrent que l'utilisation de la fonction d'activation ReLU peut être bénéfique dans les modèles de deep learning, en particulier pour les tâches de classification. Néanmoins, il est important de prendre en compte les particularités du modèle et des données, et d'expérimenter différentes fonctions d'activation pour obtenir les meilleures performances possibles.

3.4.2.1 Les matrices de confusion pour les modèles DL :

Dans ce qui suit , nous avons exposé les matrices de confusions pour les algorithmes d'apprentissage profond utilisés.

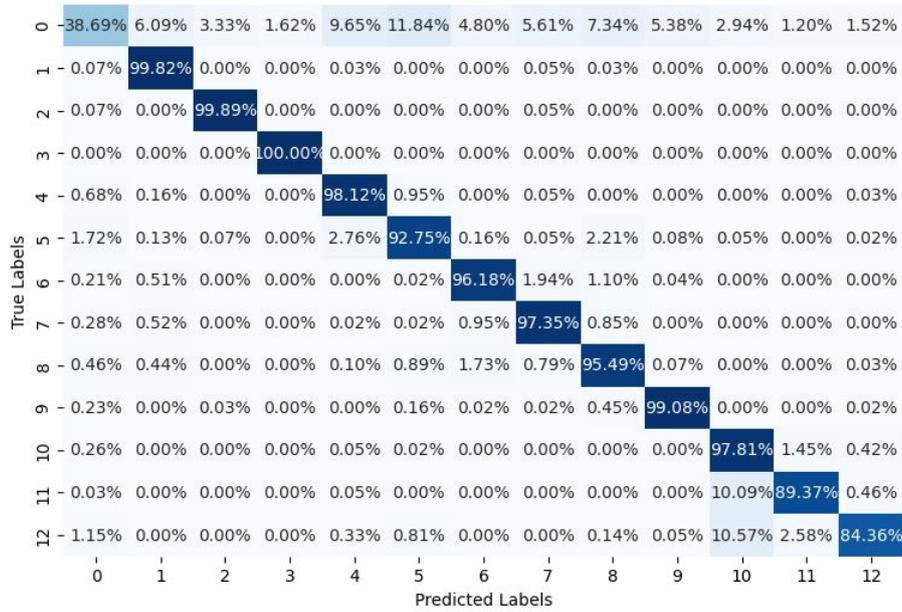


Figure 3.9: Matrice de confusion du CNN

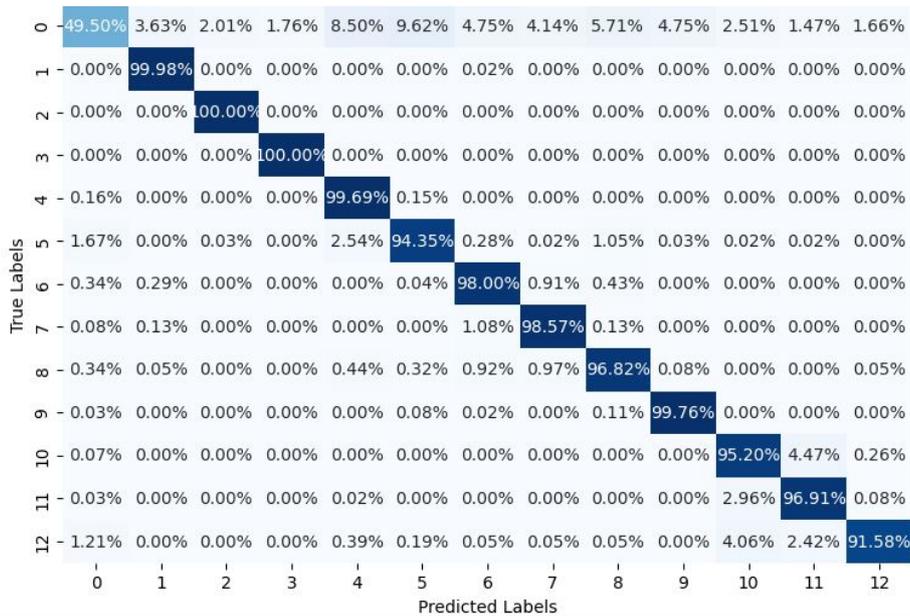


Figure 3.10: Matrice de confusion du LSTM

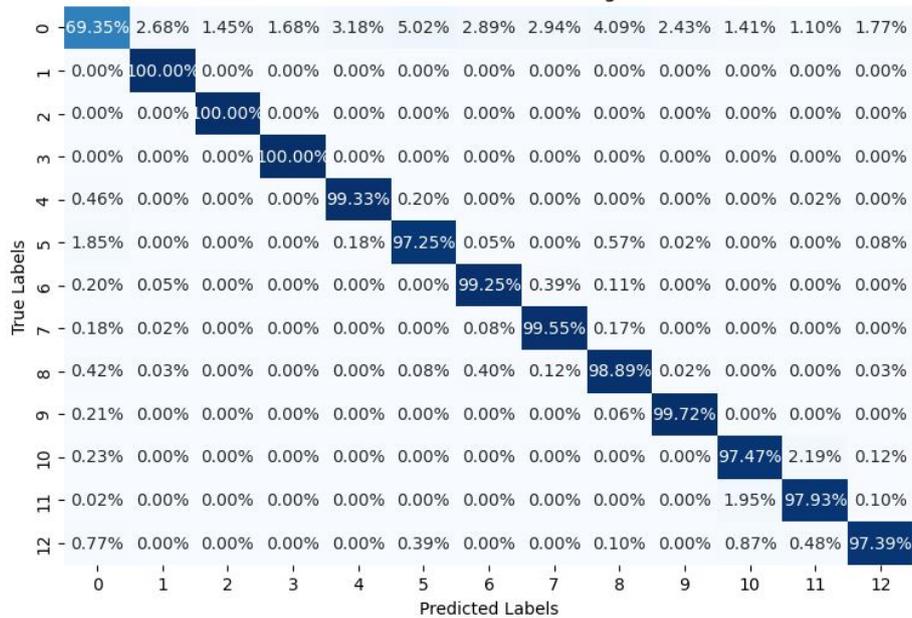


Figure 3.11: Matrice de confusion du BiLSTM

Les matrices de confusion fournissent une représentation visuelle des résultats de classification pour les modèles d'apprentissage profond. Pour chaque modèle :

Modèle CNN :

Un taux de classification de 100% pour la classe 3, ce qui indique une performance excellente pour elle.

Les taux de classification de les classes 1 et 2 sont légèrement bas, suggérant une certaine difficulté à distinguer ces classes.

Le taux de classification pour la classe 0 est le plus faible, avec seulement 38.69%.

Modèle LSTM :

Le taux de classification est 100% pour les classes 2 et 3, ces classes ont démontré une performance exceptionnelle.

Bien que les taux de classification pour les classes 1, 4 et 9 sont élevés, il y a une légère diminution par rapport aux classes 2 et 3, ce qui suggère une certaine complexité dans la distinction de ces classes.

Le taux de classification est de 49.50% pour la classe 0.

Modèle BiLSTM :

Le taux de classification est de 100% pour les classes 1, 2 et 3, démontrant une excellente performance pour ces classes. Pour la classe 0 le taux de classification est de 69.35%, ce qui est plus élevé que les autres modèles, mais suggère toujours une certaine difficulté à classer les exemples de cette classe.

En résumé, les modèles d'apprentissage profond semblent bien classer les activités 1 (se

lever), 2 (s'asseoir), 3 (allonger), 4 (marcher) et 9 (pédaler) avec des taux de classification élevés. Cependant, l'activité 0 semble être la plus difficile à classer pour tous les modèles.

3.5 Conclusion

Dans ce dernier chapitre, nous avons présenté les détails de la réalisation d'un système de reconnaissance d'activité humaine. Cette partie inclut les outils utilisés pour l'implémentation ainsi les techniques adoptés pour la phase prétraitement et la génération des modèles. Ensuite nous avons révélés les résultats des tests pour les méthodes d'apprentissage automatique et profond, à l'aide des métrique des évaluations.

Conclusion Générale

Notre projet vise le développement d'un modèle basé sur l'apprentissage automatique pour la reconnaissance d'activité humaine en utilisant une approche basée sur les capteurs, dans le but d'enrichir les travaux existants dans ce domaine.

Dans la première partie, nous avons réalisé une revue de l'état de l'art sur la reconnaissance d'activité humaine, en proposant une définition de l'activité humaine et en présentant les différents types d'activités cités dans la littérature. Nous avons également exposé les étapes clés d'un système dédié à la reconnaissance d'activité humaine, ainsi que les méthodes d'acquisition de données. Ensuite, nous avons exploré les approches d'apprentissage automatique et profond, en décrivant les algorithmes utilisés dans notre travail.

Dans la deuxième partie, nous avons présenté notre solution en élaborant une architecture qui comprend trois modules, et nous avons fourni une description détaillée de chacun de ces modules dans la section de conception détaillée. De plus, nous avons discuté des mesures utilisées pour évaluer les modèles.

Enfin, nous avons consacré la dernière partie de ce mémoire à l'implémentation de notre solution. Nous avons expliqué les techniques utilisées pour le prétraitement des données ainsi que la stratégie adoptée pour la génération des modèles d'apprentissage automatique et profond. Ensuite, nous avons présenté les résultats obtenus et effectué une comparaison entre les différents modèles.

Les perspectives de ce travail concernent les aspects suivants :

- Ces résultats suggèrent également des pistes d'amélioration futures, telles que l'optimisation des hyperparamètres, l'augmentation de la taille de l'ensemble de données d'entraînement.
- L'exploration d'autres architectures de modèles pour de meilleures performances dans des contextes spécifiques.
- Possibilité de suivre l'état de santé des patients grâce à cette solution.

Bibliographie

- [1] Djamila Romaissa Beddiar, Brahim Nini, Mohammad Sabokrou, and Abdenour Hadid. Vision-based human activity recognition : a survey. *Multimedia Tools and Applications*, 79 :30509–30555, 2020.
- [2] Neha Gupta, Suneet K Gupta, Rajesh K Pathak, Vanita Jain, Parisa Rashidi, and Jasjit S Suri. Human activity recognition in artificial intelligence framework : A narrative review. *Artificial intelligence review*, 55(6) :4755–4808, 2022.
- [3] Farzana Kulsoom, Sanam Narejo, Zahid Mehmood, Hassan Nazeer Chaudhry, Ayesha Butt, and Ali Kashif Bashir. A review of machine learning-based human activity recognition for diverse applications. *Neural Computing and Applications*, pages 1–36, 2022.
- [4] Anindita Saha, Sajan Rajak, Jayita Saha, and Chandreyee Chowdhury. A survey of machine learning and meta-heuristics approaches for sensor-based human activity recognition systems. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–28, 2022.
- [5] Abhay Gupta, Kuldeep Gupta, Kshama Gupta, and Kapil Gupta. A survey on human activity recognition and classification. In *2020 international conference on communication and signal processing (ICCSP)*, pages 0915–0919. IEEE, 2020.
- [6] Nhan Nguyen-Duc-Thanh, Daniel Stonier, SungYoung Lee, and Dong-Han Kim. A new approach for human-robot interaction using human body language. In *Convergence and Hybrid Information Technology : 5th International Conference, ICHIT 2011, Daejeon, Korea, September 22-24, 2011. Proceedings 5*, pages 762–769. Springer, 2011.
- [7] Hedi Tabia, Michèle Gouiffes, Lionel Lacassagne, and Bures sur Yvette. Reconnaissance des activités humaines à partir des vecteurs de mouvement quantifiés. 2012.
- [8] Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa. Human action recognition by representing 3d skeletons as points in a lie group. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 588–595, 2014.

- [9] Uzair Asad Akansha, Mishra Shailendra, and Narayan Singh. Analytical review on video-based human activity recognition. In *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, pages 3839–3844. IEEE, 2016.
- [10] Merouche Zakaria Khallel Houssam Eddine. Reconnaissance de l’activité humaine en utilisant les wearable sensors. Master’s thesis.2021 Université 8Mai 1945–Guelma.
- [11] L Minh Dang, Kyungbok Min, Hanxiang Wang, Md Jalil Piran, Cheol Hee Lee, and Hyeonjoon Moon. Sensor-based and vision-based human activity recognition : A comprehensive survey. *Pattern Recognition*, 108 :107561, 2020.
- [12] Issam El Naqa and Martin J Murphy. *What is machine learning?* Springer, 2015.
- [13] Enrique Garcia-Ceja. *Behavior Analysis with Machine Learning Using R. Livre.* 2021. <http://behavior.enriquegc.com>.
- [14] Youcef REMIL. Interprétation et compression des modèles boîtes noires en machine learning. Master’s thesis 2020.
- [15] MOSTEFA-CHEBRA Bilel. Analyse des sentiments sur les données des réseaux sociaux : Approche machine learning, Master’s thesis 2020. Ecole nationale superieure d’informatique.
- [16] Leif E Peterson. K-nearest neighbor. *Scholarpedia*, 4(2) :1883, 2009.
- [17] Sotiris B Kotsiantis. Decision trees : a recent overview. *Artificial Intelligence Review*, 39 :261–283, 2013.
- [18] Jake S. Rhodes, Adele Cutler, and Kevin R. Moon. Geometry- and accuracy-preserving randomforest proximities. 2023.
- [19] Dietmar P. F. Möller. *Machine Learning and Deep Learning.* April 2023.
- [20] Jianqing Fan, Cong Ma, and Yiqiao Zhong. A selective overview of deep learning. *Statistical science : a review journal of the Institute of Mathematical Statistics*, 36(2) :264, 2021.
- [21] Y. Djeriri. Les réseaux de neurones artificiels, 2017.
- [22] Philip J Drew and John RT Monson. Artificial neural networks. *Surgery*, 127(1) :3–11, 2000.
- [23] Qingwen Han, Ke Liu, Lingqiu Zeng, Guangyan He, Lei Ye, and Fengxi Li. A bus arrival time prediction method based on position calibration and lstm. *IEEE Access*, 8 :42372–42383, 2020.
- [24] Shaoming Qiu, Yahui Wang, Yana Lv, Fen Chen, and Jiancheng Zhao. Optimizing bilstm network attack prediction based on improved gray wolf algorithm. *Applied Sciences*, 13(12) :6871, 2023.

- [25] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. Understanding of a convolutional neural network. In *2017 international conference on engineering and technology (ICET)*, pages 1–6. Ieee, 2017.
- [26] Sumit Saha. A comprehensive guide to convolutional neural networks — the eli5 way, 2020.
- [27] N MOULKRALOUA and F ZENAGUI. Reconnaissance de l’activité humaine à partir des données smartphone, Master’s thesis 2021.
- [28] definition of flatten layers. <https://www.pcmag.com/encyclopedia/term/flatten-layers>.
- [29] Jakaria Rabbi, Md Fuad, Tahmid Hasan, Md Awal, et al. Human activity analysis and recognition from smartphones using machine learning techniques. *arXiv preprint arXiv :2103.16490*, 2021.
- [30] Junhuai Li, Ling Tian, Linglun Chen, Huaijun Wang, Ting Cao, and Lei Yu. Optimal feature selection for activity recognition based on ant colony algorithm. In *2019 14th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pages 2356–2362. IEEE, 2019.
- [31] Ku Nurhanim, Irraivan Elamvazuthi, LI Izhar, and T Ganesan. Classification of human activity based on smartphone inertial sensor using support vector machine. In *2017 ieee 3rd international symposium in robotics and manufacturing automation (roma)*, pages 1–5. IEEE, 2017.
- [32] Fajri Nurwanto, Igi Ardiyanto, and Sunu Wibirama. Light sport exercise detection based on smartwatch and smartphone using k-nearest neighbor and dynamic time warping algorithm. In *2016 8th International Conference on Information Technology and Electrical Engineering (ICITEE)*, pages 1–5. IEEE, 2016.
- [33] Yiming Tian, Jie Zhang, Lingling Chen, Yanli Geng, and Xitai Wang. Single wearable accelerometer-based human activity recognition via kernel discriminant analysis and qpso-kelm classifier. *Ieee Access*, 7 :109216–109227, 2019.
- [34] Shaohua Wan, Lianyong Qi, Xiaolong Xu, Chao Tong, and Zonghua Gu. Deep learning models for real-time human activity recognition with smartphones. *Mobile Networks and Applications*, 25 :743–755, 2020.
- [35] ZELLEG imane DRID, Akram. Mise en place d’un tableau de bord de performance pour la direction de la stratégie de mobilis.
- [36] Nirmal Sankalana. mhealth dataset - data set csv. <https://www.kaggle.com/datasets/nirmalsankalana/mhealth-dataset-data-set-csv>, 2023.

- [37] Abdulhamit Subasi, Dalia H Dammas, Rahaf D Alghamdi, Raghad A Makawi, Eman A Albiety, Tayeb Brahimi, and Akila Sarirete. Sensor based human activity recognition using adaboost ensemble classifier. *Procedia computer science*, 140 :104–111, 2018.
- [38] Tianlan Mo, Linjing Wang, Yuliang Wu, Junrong Huang, Weikun Liu, Ruimeng Yang, and Xin Zhen. Classifier ensemble with evolutionary optimisation enforced random projections. *Expert Systems with Applications*, 222 :119845, 2023.
- [39] Soraya Cheriguene. *Approches Ensemblistes de Classification et Sélection des Prédicteurs pour la Reconnaissance de Formes*. PhD thesis, 2018.
- [40] Daniel Garcia-Gonzalez, Daniel Rivero, Enrique Fernandez-Blanco, and Miguel R Luaces. New machine learning approaches for real-life human activity recognition using smartphone sensor-based data. *Knowledge-Based Systems*, page 110260, 2023.
- [41] Aiguo Wang, Guilin Chen, Jing Yang, Shenghui Zhao, and Chih-Yung Chang. A comparative study on human activity recognition using inertial sensors in a smartphone. *IEEE Sensors Journal*, 16(11) :4566–4578, 2016.
- [42] Md Tahmid Hasan Fuad Jakaria Rabbi and Md Abdul Awal. Human activity analysis and recognition from smartphones using machine learning techniques. *arXiv preprint arXiv :2103.16490*, 2021.
- [43] Gobinda G Chowdhury. *Introduction to modern information retrieval*. Facet publishing, 2010.
- [44] Tianci Zhang, Jinglong Chen, Fudong Li, Kaiyu Zhang, Haixin Lv, Shuilong He, and Enyong Xu. Intelligent fault diagnosis of machines with small & imbalanced data : A state-of-the-art review and possible extensions. *ISA transactions*, 119 :152–171, 2022.