

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE  
Ministère de L'Enseignement Supérieur et de la Recherche Scientifique  
**ECOLE NATIONALE POLYTECHNIQUE**  
Département d'Électronique



المدرسة الوطنية المتعددة التقنيات  
Ecole Nationale Polytechnique

Mémoire de Projet de Fin d'Études en vue de l'obtention du diplôme  
d'Ingénieur d'État en Électronique

Présenté par : **BOUDIA Abderaouf et BOUKHEDIMI Faten**

### Sujet

**Application des techniques de l'apprentissage  
automatique pour l'analyse de la marche chez les  
groupes de population à haut risque de chute.**

Soutenue publiquement, le 21/06/2023, devant le jury composé de :

M.	Sid-Ahmed BERRANI	Docteur	à l'ENP	Président
Mme.	LANI Fatiha	Docteur	à l'ENP	Examinatrice
Mme.	DIB Wassila	Attaché de recherche A	au CDTA	Promotrice
Mme	BOUADJNEK Nesrine	Docteur	à ENP	Co-promotrice

**ENP 2023**

**10 Rue des Frères OUDEK, El Harrach 16200**



RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE  
Ministère de L'Enseignement Supérieur et de la Recherche Scientifique  
**ECOLE NATIONALE POLYTECHNIQUE**  
Département d'Électronique



المدرسة الوطنية المتعددة التقنيات  
Ecole Nationale Polytechnique

Mémoire de Projet de Fin d'Études en vue de l'obtention du diplôme  
d'Ingénieur d'État en Électronique

Présenté par : **BOUDIA Abderaouf et BOUKHEDIMI Faten**

### Sujet

**Application des techniques de l'apprentissage  
automatique pour l'analyse de la marche chez les  
groupes de population à haut risque de chute.**

Soutenue publiquement, le 21/06/2023, devant le jury composé de :

M.	Sid-Ahmed BERRANI	Docteur	à l'ENP	Président
Mme.	LANI Fatiha	Docteur	à l'ENP	Examinatrice
Mme.	DIB Wassila	Attaché de recherche A	au CDTA	Promotrice
Mme	BOUADJNEK Nesrine	Docteur	à ENP	Co-promotrice

**ENP 2023**

**10 Rue des Frères OUDEK, El Harrach 16200**

Je dédie ce travail,

À toi, ma mère, Qui a fait de moi la personne que je suis aujourd'hui, je t'exprime ma gratitude la plus profonde pour tout ce que tu as fait. Je suis fier de te dédier ce projet, le fruit de ton amour inconditionnel et de ton soutien sans faille. Que cette dédicace témoigne de mon amour et de ma reconnaissance éternels envers toi. Tu es et seras toujours ma plus grande inspiration, et je suis honoré d'être ton enfant.

À mon père, Un modèle de persévérance et de détermination. Ton engagement absolu envers mon éducation et ton soutien constant m'ont poussé à donner le meilleur de moi-même. Je t'exprime ma reconnaissance sincère pour ton amour inépuisable et je dédie ce projet à ta fierté et à ta confiance en moi.

À mes frères et sœurs, Des compagnons de vie, votre présence m'a apporté du réconfort et de la motivation tout au long de ce parcours. Je vous suis reconnaissant pour les moments partagés, les encouragements et le soutien inconditionnel.

À mon binôme, je souhaite exprimer ma reconnaissance la plus sincère. Notre collaboration, notre complémentarité et notre dévouement mutuel ont été essentiels pour mener à bien ce projet.

À mes amis, Khawla, Katia, Zahra, Maroua, Romeila, Anes, Riad, votre soutien indéfectible, votre présence et vos mots d'encouragement ont été une source de motivation inestimable.

À ma chère promotion ELN MATCH, Ensemble, Nous avons surmonté les défis, les longues heures d'étude et les joies de l'apprentissage. Je vous exprime ma reconnaissance pour notre esprit d'équipe et notre entraide. Merci d'avoir rendu cette expérience académique enrichissante et plaisante.

**BOUKHEDIMI Faten**

Je dédie ce travail,

À ma mère, ton soutien inconditionnel et ta bienveillance ont été les moteurs de ma réussite. Je te suis infiniment reconnaissant pour ton amour et ton dévouement.

À mon père, qui s'est levé chaque matin pour me déposer à l'école, m'encourageant à poursuivre mes rêves et à ne jamais abandonner. Ta présence et ton soutien constants ont été essentiels pour me donner la force et la détermination nécessaires pour atteindre mes objectifs.

À mes chers frères et sœurs, vous êtes ma famille précieuse, source de bonheur et de soutien inébranlable.

À mes chers amis, studio Gang, Zaki, Kenzi, Okba, Khelifi, Rahim, Yacine, Jimmy et Yanis, je tiens à exprimer ma profonde reconnaissance pour notre amitié indéfectible, nos précieux moments partagés et notre complicité sans faille.

À mon comité du CAP 2022, mes amis, Aribus, Kellou, Celia, Ferrah, Hayat, Sofia, Admadi, Kelkouli, Imene, Rachel, votre présence est d'une valeur inestimable.

À tous les membres du CAP, j'ai adoré organiser pleins d'événements avec vous, toujours dans la bonne humeur.

À mes camarades, ELN Match, nous avons partagé tant de moments ensemble, des heures d'étude aux moments de détente. Votre soutien et votre camaraderie ont été inestimables tout au long de notre parcours.

À ma binôme, à notre collaboration, à notre complicité inégalée et au temps précieux que nous avons consacré ensemble pour travailler sur ce projet.

À Nawel, ma jumelle et mon soutien moral, je te remercie pour ta présence constante.

Le meilleur pour la fin, à Karine, je t'exprime ma profonde gratitude ton soutien indéfectible. Tu as été ma source de réconfort et de motivation.

**BOUDIA Abderaouf**

# Remerciement

Nous voici arrivés à la fin d'un chapitre majeur de notre parcours académique, une étape qui nous aura marqués par son intensité et ses défis. En rédigeant ces mots, nous ressentons une profonde émotion mêlée de fierté et de reconnaissance.

Nous souhaitons avant tout exprimer notre gratitude envers nos parents. Vous avez été notre source d'inspiration et notre motivation tout au long de nos études. Merci pour votre soutien infaillible en nous. Vos sacrifices ne seront jamais oubliés.

Nous tenons à exprimer nos plus profondes gratitudes et nos remerciements à notre encadrante de projet, Madame Wassila DIB, attachée de recherche chez CDTA, pour son soutien constant et ses conseils précieux, pour sa gentillesse et sa présence inébranlable, pour le savoir qu'elle partage toujours généreusement et pour sa passion contagieuse pour l'ingénierie et l'électronique, qui inspirent admiration et inspiration.

Nous tenons à exprimer une fois de plus nos sincères gratitudes envers notre encadrante, Mme. Nesrine BOUADJENEK, professeur à l'École Nationale Polytechnique, pour sa supervision et sa flexibilité, pour sa guidance précieuse et ses conseils avisés. Tout au long de notre projet de fin d'études, elle a partagé son assistance et ses connaissances avec une gentillesse constante. Nous tenons à remercier le président du jury, M. Sid-Ahmed Berrani, professeur à l'École Nationale Polytechnique, ainsi que l'examinatrice, Mme. Fatiha LANI, enseignante à l'École Nationale Polytechnique, d'avoir accepté d'être membres du comité de lecture et pour leur analyse constructive du présent travail. Nous leur dédions un remerciement particulier pour leur temps et leur enseignement au cours de ces trois dernières années

Nous exprimons également notre gratitude à tous ceux qui nous ont apporté leur soutien, qu'il s'agisse de nos amis ou de nos familles. Leur aide précieuse, directe ou indirecte, a contribué à la réalisation de ce travail de recherche.

## ملخص

السقوط بين كبار السن يُمثّل تحديًا كبيرًا في مجال الصحة العامة. في هذا البحث، نقدم نهجًا للتعلم الآلي يستند إلى تحليل المشي لحل هذه المشكلة. تم استغلال الإشارات المستخرجة من قاعدة بيانات "رصد الحركة على المدى الطويل"، حيث يتم جمع هذه الإشارات باستخدام وحدة قياس انعطافية واحدة موجودة في الجزء السفلي من الظهر. تم استخدام خوارزميات التجزئة لاستخراج السمات الزمكانية والإحصائية، وتم استخدام خوارزميات التصنيف المختلفة. تم تحقيق نتائج واعدة للتصنيف كمن بدقة 89.7% ونسبة ف1 90%، وللتصنيف سقم بدقة 87.6% ونسبة ف1 90%. تم تحقيق هذه النتائج مع ضمان إمكانية تنفيذ التطبيقات المستقبلية بسهولة.

---

كلمات مفتاحية : . كشف عن السقوط، حساس القصور ، التعلم الآلي، خوارزمية التصنيف

---

## Abstract

Falls among the elderly population represent a significant public health issue. In this study, we propose a machine learning approach that leverages gait analysis to address this problem. We utilized signals from the Physionet "Long Term Movement Monitoring" database, which were gathered using a single inertial measurement unit positioned at the lower back. Additionally, we employed segmentation algorithms to extract spatiotemporal and statistical features, and we utilized various classification algorithms. Encouraging results were obtained for the KNN classifier, yielding an accuracy of 89.7% and an F1-score of 90%, as well as for the SVM classifier, with an accuracy of 87.6% and an F1-score of 90%. These results were achieved while ensuring the feasibility of future implementations.

---

**Keywords :** Fall detection, inertial sensor, machine learning, classification algorithm.

---

## Résumé

La chute chez les personnes âgées est un enjeu majeur de la santé publique. Dans ce travail, nous proposons une approche d'apprentissage automatique basée sur l'analyse de la marche pour résoudre cette problématique. Les signaux issus de la base de données Physionet "Long Term Movement Monitoring" ont été exploités, ceux-ci sont collectés avec une seule unité de mesure inertielle située dans le bas du dos. De plus, des algorithmes de segmentation pour l'extraction des caractéristiques spatio-temporelles et statistiques, ainsi que différents algorithmes de classification ont été utilisés. Des résultats prometteurs ont été obtenus pour le classifieur KNN, avec une exactitude de 89.7% et un F1-score de 90%, et pour le SVM avec une exactitude de 87.6% et un F1-score de 90%. Ces résultats ont pu être obtenus tout en assurant la faisabilité de futures implémentations.

---

**Mots clés :** Détection de chute, capteur inertielle, apprentissage automatique, algorithme de classification.

---

# Table des matières

Table des figures

Liste des tableaux

<b>Introduction Générale</b>	<b>14</b>
<b>1 Analyse de la marche : Généralités et système portable</b>	<b>16</b>
1.1 Introduction . . . . .	16
1.2 Le modèle de la marche . . . . .	16
1.2.1 Définition de la marche . . . . .	16
1.2.2 Cycle de la marche . . . . .	16
1.2.3 Les caractéristiques de la marche . . . . .	17
1.3 Analyse de la marche . . . . .	21
1.3.1 Approche basée sur la vision . . . . .	22
1.3.2 Approche basée sur les capteurs . . . . .	23
1.4 Les systèmes de détection de chute portable basés sur l'analyse de la marche :	24
1.4.1 Approches de la détection des chutes dans les systèmes portable . .	25
1.4.2 Défis des systèmes de détection des chutes portables . . . . .	26
1.5 L'apprentissage automatique (Machine Learning ML) dans l'analyse de la marche aux fins de la détection des chutes . . . . .	27
1.5.1 Acquisition et pré-traitement de données : . . . . .	27
1.5.2 La segmentation de la marche . . . . .	29
1.5.3 Extraction des caractéristiques . . . . .	30
1.5.4 La classification . . . . .	30
1.6 Travaux connexes . . . . .	30
1.7 Conclusion . . . . .	32
<b>2 Méthodologie</b>	<b>33</b>
2.1 Introduction . . . . .	33



---

2.2	Les algorithmes de la segmentation . . . . .	33
2.2.1	Algorithme CWT . . . . .	34
2.2.2	Algorithme de Shin . . . . .	36
2.2.3	Algorithme de Gonzalez . . . . .	37
2.2.4	Algorithme de Ghersi . . . . .	38
2.3	Extraction des caractéristiques . . . . .	43
2.3.1	Caractéristiques spatio-temporelles . . . . .	43
2.3.2	Caractéristiques statistiques . . . . .	48
2.4	Apprentissage automatique . . . . .	49
2.4.1	K-plus proches voisins (K-Nearest Neighbor) . . . . .	50
2.4.2	Machine à vecteurs de support SVM : . . . . .	51
2.4.3	Arbre de décision (Decision trees) : . . . . .	52
2.4.4	Apprentissage ensembliste : . . . . .	54
2.4.5	Forêt aléatoire (Random Forest) : . . . . .	56
2.4.6	Réseaux de neurones : . . . . .	57
2.5	Deep learning . . . . .	59
2.5.1	Réseau de Neurones Convolutif (CNN) : . . . . .	60
2.6	Réduction de dimension : . . . . .	62
2.7	Conclusion . . . . .	66
<b>3</b>	<b>Analyse et resultats</b>	<b>67</b>
3.1	Introduction . . . . .	67
3.2	Base de donnée . . . . .	67
3.3	Application des algorithmes de segmentation pour la détection du Contact Initial et du Contact Final . . . . .	70
3.4	Analyse Comparative des Algorithmes de segmentation des Paramètres de Marche . . . . .	72
3.4.1	Évaluation de l'algorithme GHARSI . . . . .	72
3.4.2	Évaluation de l'algorithme SHIN . . . . .	72
3.4.3	Évaluation de l'algorithme Gonzalez . . . . .	73
3.4.4	Évaluation de l'algorithme CWT . . . . .	73
3.4.5	Choix de l'algorithme . . . . .	73
3.5	Extraction et Présentation des Caractéristiques Additionnelles pour les Mo- dèles d'Apprentissage Automatique . . . . .	74
3.6	Application des techniques de l'apprentissage automatique . . . . .	79
3.6.1	Répartition des données . . . . .	79
3.6.2	Réduction de la dimension . . . . .	81

3.6.3	Entraînement et paramétrage des modèles d'apprentissage automatique . . . . .	83
3.7	Présentation des résultats . . . . .	91
3.7.1	Métriques d'évaluation pour évaluer les performances des modèles .	91
3.7.2	Performances des modèles . . . . .	93
3.8	Discussion . . . . .	95
3.9	Conclusion . . . . .	97
	<b>Conclusion Générale</b>	<b>98</b>
	<b>Bibliographie</b>	<b>100</b>

# Table des figures

1.1	Cycle de la marche fondamentale. . . . .	17
1.2	Arbre des paramètres de l'analyse de la marche. . . . .	18
1.3	Schéma d'approches de l'analyse de la marche. . . . .	22
1.4	Chaîne d'analyse de la marche. . . . .	27
2.1	Schéma de la méthodologie complète. . . . .	34
2.2	Détection CF et CI à l'aide de l'algorithme de la transformée en ondelettes continues. . . . .	35
2.3	Algorithme de Shin. . . . .	37
2.4	Détection CF et CI à l'aide de l'algorithme de Ghersi. . . . .	39
2.5	Algorithmes de correction et d'optimisation . . . . .	43
2.6	a Caractéristique spatiaux de la marche. b Caractéristique temporels de la marche. . . . .	44
2.7	L'algorithme SVM : Hyperplan à marge maximale et vecteurs de support. . . . .	52
2.8	Principe de l'algorithme de l'arbre de décision. . . . .	54
2.9	Principe du Bagging : Combinaison d'estimateurs par échantillonnage avec remplacement. . . . .	55
2.10	Principe du Stacking : Combinaison de modèles par empilement. . . . .	55
2.11	Principe du Boosting : Renforcement de modèles par apprentissage itératif. . . . .	56
2.12	Principe des Forêts Aléatoires : Agrégation de Modèles d'Arbres de Décision. . . . .	57
2.13	Principe des Réseaux de Neurones : Modélisation d'un simple réseaux de neurones. . . . .	58
2.14	Graphe de la Fonction de Coût. . . . .	59
2.15	Réseau de Neurones Convolutif 2D (CNN) : Exemple d'xtraction de caractéristiques à partir d'images en utilisant des filtres convolutionnels. . . . .	61
2.16	Réseau de Neurones Convolutif 1D (CNN) : Extraction de caractéristiques à partir de séquences temporelles à l'aide de filtres convolutionnels. . . . .	62
2.17	Méthodes de réduction de la dimension . . . . .	64

2.18	Les étapes principales de l'ACP . . . . .	65
3.1	Exemple de la base de données des signaux d'accéléromètres et de gyroscopes des personnes non chuteurs. . . . .	69
3.2	Exemple de la base de données des signaux d'accéléromètres et de gyroscopes des personnes chuteurs. . . . .	69
3.3	Application de l'algorithme GHARSI : détection des points CI et CF sur l'axe x des signaux d'accéléromètre. . . . .	70
3.4	Application de l'algorithme Shin : détection des points CI et CF sur l'axe x des signaux d'accéléromètre. . . . .	70
3.5	Application de l'algorithme CWT : détection des points CI et CF sur l'axe x des signaux d'accéléromètre. . . . .	71
3.6	Application de l'algorithme Gonzalez : détection des points CI et CF sur l'axe x des signaux d'accéléromètre. . . . .	71
3.7	L'ensemble des caractéristique extraite. . . . .	75
3.8	Visualisation des données en fonction caractéristiques. . . . .	76
3.9	Distibutions des données en fonctions des caractéristiques. . . . .	77
3.10	Histogrammes des caractéristiques : Comparaison entre les chuteurs et les non-chuteurs. . . . .	78
3.11	Variance cumulée de l'ACP en fonction des caractéristiques extraites. . . . .	81
3.12	Validation croisée . . . . .	84
3.13	Matrice de confusion. . . . .	92
3.14	La courbe ROC. . . . .	93

# Liste des tableaux

1.1	Résumé des études sélectionnées. Les forces et les faiblesses sont basées sur la perception de l'auteur. . . . .	30
1.2	Performances récentes dans l'évaluation du risque de chute à l'aide de capteurs inertiels et d'apprentissage automatique : Études connexes . . . . .	32
2.1	Méthode d'estimation de longueur de pas. . . . .	47
3.1	Données démographiques, performance fonctionnelle et mesures dérivées de l'accélération en laboratoire . . . . .	69
3.2	Distribution moyenne des paramètres de la marche spatiale-temporelle chez les chuteurs et non chuteurs . . . . .	72
3.3	Répartition de l'ensemble des caractéristiques. . . . .	80
3.4	Tableau des Composantes Principales et Caractéristiques Essentielles. . . . .	82
3.5	Exemple de la grille d'hyperparamètres du Support Vector Machine (SVM)	85
3.6	Exemple de la grille d'hyperparamètres du l'arbre de décision. . . . .	86
3.7	Exemple de la grille d'hyperparamètres du Random forest (RF). . . . .	87
3.8	Exemple de la grille d'hyperparamètres du réseau de neurones (DNN). . . . .	88
3.9	Exemple de la grille d'hyperparamètres CNN1D. . . . .	88
3.10	Tableau des hyperparamètres optimaux pour les modèles d'apprentissage automatique . . . . .	89
3.11	Meilleurs hyperparamètres pour le modèle DNN. . . . .	90
3.12	Meilleurs hyperparamètres pour le modèle CNN1D. . . . .	90
3.13	Performances Comparatives des Modèles d'Apprentissage Automatique. . . . .	94
3.14	Précision, Rappel et Score F1 des Modèles d'Apprentissage Automatique pour la Classification de Chute et Non-Chute . . . . .	95

# Liste des abréviations

<b>OMS</b>	<i>Organisation Mondiale de la Santé.</i>
<b>MP</b>	<i>Maladie de Parkinson.</i>
<b>CI</b>	<i>Contact Initial.</i>
<b>CF</b>	<i>Contact Final.</i>
<b>EMG</b>	<i>Electromyogramme.</i>
<b>IMC</b>	<i>Indice de Masse Corporelle.</i>
<b>GRF</b>	<i>Ground Reaction Force (Forces de Réaction au Sol.</i>
<b>MUAP</b>	<i>Motor Unit Action Potential (potentiel d'action des unités motrices.)</i>
<b>IRM</b>	<i>Imagerie par Résonance Magnétique.</i>
<b>SSC</b>	<i>Systèmes Sensibles au Contexte.</i>
<b>SDC</b>	<i>Systèmes de Détection des Chutes.</i>
<b>IA</b>	<i>Intelligence Artificielle.</i>
<b>ML</b>	<i>Machine Learning ( Apprentissage Automatique).</i>
<b>RNN</b>	<i>Recurrent Neural Network.</i>
<b>LSTM</b>	<i>Long Short Term Memory.</i>
<b>IMU</b>	<i>Unité de Mesure Inertielle.</i>
<b>SVM</b>	<i>Support Vector Machine (Machine à Vecteurs de Support).</i>
<b>KNN</b>	<i>K-Nearest Neighbor (K-plus proches voisins).</i>
<b>DT</b>	<i>Decision trees (Arbres de décision).</i>

<b>RF</b>	<i>Random Forest (Forêt aléatoire).</i>
<b>CNN</b>	<i>Convolutional Neural Network (Réseau de Neurones Convolutif).</i>
<b>RNA</b>	<i>Réseaux Nneuronaux Artificiels.</i>
<b>ZC</b>	<i>Zero Crossing (passage par zéro).</i>
<b>CWT</b>	<i>Continuous Wavelet Transform ( Transformée en Ondelettes Continue).</i>
<b>DWT</b>	<i>Discret Wavelet Transform (Transformée en Ondelettes Discrète).</i>
<b>AP</b>	<i>Accélération Antéro Postérieure.</i>
<b>ACP</b>	<i>Analyse en Composantes Principales.</i>
<b>AF</b>	<i>Analyse Factorielle.</i>
<b>LDA</b>	<i>Linear Discriminant Analysis (Analyse Factorielle Discriminante).</i>
<b>T-SNE</b>	<i>t-Distributed Stochastic Neighbor Embedding (Intégration des voisins stochastiques distribués en t).</i>

# Introduction Générale

Les chutes sont responsables au deuxième rang en termes de décès accidentels ou de décès par traumatisme involontaire à l'échelle mondiale, ce qui constitue un problème de santé majeur. Les personnes âgées sont les plus sujettes à ces incidents de chutes, en raison de nombreux facteurs, tels que la diminution de la force musculaire, la diminution de l'équilibre, les troubles de la vision et les problèmes de mobilité.

Les statistiques de l'Organisation Mondiale de la Santé (OMS) révèlent qu'un nombre considérable de personnes âgées (65 ans et plus) subit des chutes chaque année [1]. Ces incidents engendrent des blessures graves telles que des fractures, des traumatismes crâniens et des atteintes musculo-squelettiques, nécessitant souvent une hospitalisation et une rééducation prolongée.

En effet, la prévention des chutes est d'une importance capitale pour garantir la sécurité et le bien-être des individus vieillissants. En identifiant les personnes qui présentent un risque élevé de chute, des mesures préventives adaptées peuvent être mises en place pour réduire ces risques et prévenir les blessures potentielles. Cela permet de préserver la qualité de vie des personnes âgées.

Les techniques d'analyse de la marche ont considérablement évolué au cours de la dernière décennie. La plupart des solutions permettent cependant d'estimer les phases de la marche sur la base d'une approche heuristique des angles articulaires ou des points temporels de certains événements de la marche. Or, ces mêmes approches ne fonctionnent souvent pas bien dans les groupes de population présentant un risque élevé de chute (personnes âgées, personnes atteintes de la maladie de Parkinson (MP) et pompiers en raison de la variabilité de leur cycle de marche par rapport aux personnes normales.

Ce travail vise à appliquer des techniques de l'apprentissage automatique dans l'analyse de la marche de groupes de population à haut risque de chute, plus spécifiquement les personnes âgées.

Pour atteindre cet objectif, nous devons d'abord nous familiariser avec la nature du signal qui nous intéresse, c'est-à-dire le signal relatif à la marche, son cycle et les caractéristiques qui nous permettront de distinguer un cycle de marche normal, d'un cycle



susceptible de conduire à une chute. Une fois maîtrisées, les techniques d'apprentissage automatique seront appliquées en respectant la chaîne complète d'un système de détection de chute, depuis la segmentation et l'extraction de caractéristiques jusqu'aux algorithmes de classification. Une fois la chaîne établie, les résultats seront évalués à l'aide de métriques dédiées.

Afin de mieux présenter notre travail, nous structurons notre mémoire en trois chapitres selon le plan suivant :

Dans le premier chapitre de notre étude, nous abordons la notion de marche en la définissant avec précision et en décrivant le déroulement cyclique de ce mouvement. De plus, nous nous penchons sur les différentes approches proposées dans la littérature pour analyser la marche.

Nous consacrons le deuxième chapitre à l'exploration de l'utilisation de l'apprentissage automatique. Nous accordons une attention particulière aux étapes essentielles de ce processus, à savoir le prétraitement des données, la segmentation de la marche, l'extraction des caractéristiques et la classification. Nous présentons plusieurs algorithmes d'apprentissage automatique de grande pertinence, tels que les méthodes des k-plus proches voisins, les machines à vecteurs de support, les arbres de décision. . .

Le troisième chapitre se concentre sur l'analyse approfondie des résultats obtenus. Nous débutons en décrivant en détail la base de données utilisée. Nous procédons ensuite à une analyse comparative approfondie de ces algorithmes et explorons l'intégration d'extraction de caractéristiques supplémentaires dans le but d'améliorer les modèles d'apprentissage automatique. Enfin, nous exposons les résultats obtenus, mettant en évidence les performances de classification obtenues grâce à nos méthodes.

Nous terminerons notre mémoire par une conclusion générale, dans laquelle nous présenterons un résumé du travail effectué au cours de ce projet.

# Chapitre 1

## Analyse de la marche : Généralités et système portable

### 1.1 Introduction

L'analyse de la marche est un domaine de recherche essentiel dans le secteur de la santé et de la mobilité humaine. Comprendre les mécanismes et les caractéristiques de la marche peut fournir des informations précieuses pour évaluer la fonctionnalité, détecter les anomalies et améliorer la qualité de vie des individus.

Dans ce chapitre, nous explorerons les concepts fondamentaux liés à la marche ainsi que les différents systèmes utilisés pour l'analyse et la détection des chutes. Nous aborderons l'utilisation de l'apprentissage automatique (Machine Learning) dans l'analyse de la marche aux fins de la classification des chutes.

### 1.2 Le modèle de la marche

#### 1.2.1 Définition de la marche

La marche est considérée comme une activité stéréotypée chez les personnes jeunes et âgées en bonne santé. Elle consiste en une séquence de mouvements involontaires déclenchés par des mouvements volontaires, répétés de façon cyclique, connus sous le nom de cycles de marche.

#### 1.2.2 Cycle de la marche

Un cycle de marche se compose principalement de deux phases : une phase d'appui et une phase d'oscillation [2]. En général, la phase d'appui commence par le contact du

talon et se termine par le décollement des orteils du même pied, elle représente la durée durant laquelle le pied reste en contact avec le sol, celle-ci représente environ 60 % du cycle normal de la marche. D'un autre côté, la durée durant laquelle le pied ne touche pas le sol est appelée phase d'oscillation et représente 40 % du cycle de marche, elle commence par le décollement de la pointe du pied effacé et se termine par le contact du talon de ce même pied. La phase d'appui et la phase d'oscillation peuvent être segmentées en huit parties qui permettent aux examinateurs de préciser les aspects anormaux de la marche.

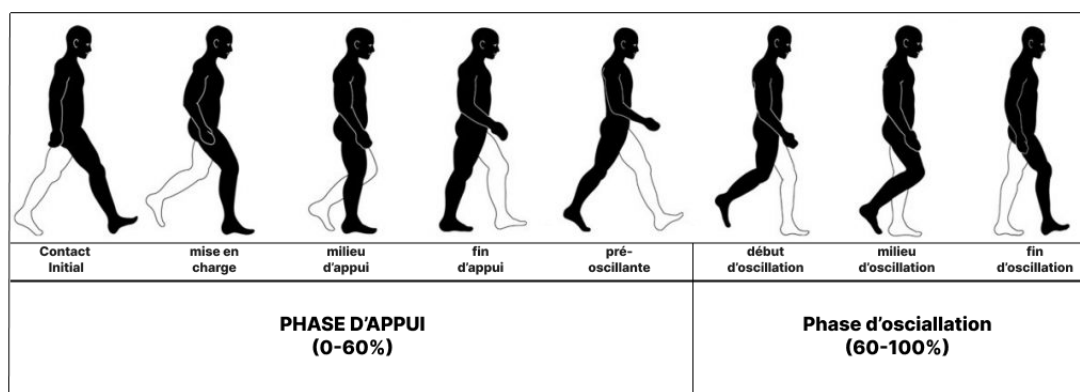


FIGURE 1.1 – Cycle de la marche fondamentale.

Le modèle de marche classique de Perry divise le cycle de marche en huit sous-phases (cinq phases d'appui et trois phases de balancement) [2]. En outre, la phase d'appui est divisée en cinq sous-phases : Le contact initial, la mise en charge, le milieu d'appui, la fin d'appui et la pré-oscillante, tandis que la phase d'oscillation est divisée en trois sous-phases : le début d'oscillation, le milieu d'oscillation et la fin d'oscillation. La figure 1.1 montre les principes fondamentaux des phases de la marche et l'intervalle prévu entre les phases et les sous-phases dans le cycle total de la marche [3].

### 1.2.3 Les caractéristiques de la marche

Selon la littérature, six types de paramètres peuvent être utilisés dans l'analyse de la marche (voir figure 1.2) à savoir : des paramètres anthropométriques, des paramètres spatio-temporels, des paramètres cinématiques, des paramètres cinétiques, des paramètres dynamiques basés sur l'électromyographie (EMG), des paramètres statistiques, ainsi que des paramètres basés sur des caractéristiques combinées [3].

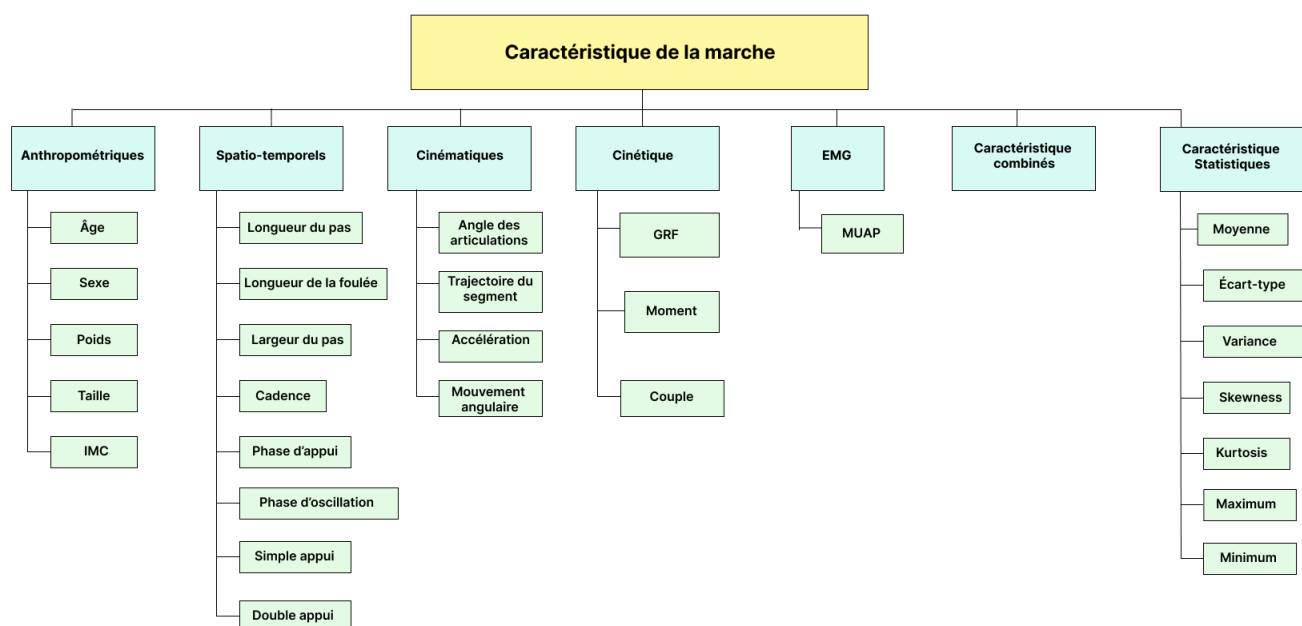


FIGURE 1.2 – Arbre des paramètres de l'analyse de la marche.

**Caractéristiques anthropométriques :** Ce sont des caractéristiques est basée sur des paramètres tels que les dimensions corporelles de l'être humain, notamment l'âge, le sexe, le poids, la taille et l'indice de masse corporelle (IMC) d'un sujet [4].

**Caractéristiques spatio-temporelles :** Ils sont également connus sous le nom de caractéristiques généraux de la marche. Ils sont responsables de la forme la plus simple d'évaluation objective de la marche [5,6].

**Caractéristiques cinématiques :** Les caractéristiques cinématiques sont les mesures du mouvement ou la description géométrique du mouvement des segments du corps. Ils comprennent les angles des articulations en tenant compte du mouvement du repère corporel sélectionné pour l'analyse. Outre l'angle des articulations (angle du tronc, angle de la hanche, angle du genou, angle de la cheville, etc.), il comprend également le mouvement angulaire, l'accélération et la trajectoire du segment. Habituellement, des marqueurs et des capteurs sont utilisés pour mesurer ces paramètres [3,7].

**Caractéristiques cinétiques :** Il s'agit du groupe de forces impliquées dans la production des forces de réaction au sol (GRF), du couple, des modèles de pression et des forces articulaires. Les plateformes de force sont utilisées pour mesurer les composantes

tridimensionnelles de la force [8].

**Caractéristiques basées sur l’EMG :** Des électrodes EMG de surface ou d’aiguille sont utilisées pour enregistrer les potentiels d’action des unités motrices (MUAP) [3, 9]. Ces données permettent l’étude de l’activité musculaire pendant la marche.

**Caractéristiques statistiques :** Il s’agit des mesures quantitatives qui permettent de décrire et d’analyser de manière objective les données relatives à la marche. Elles sont utilisées pour quantifier différents aspects du mouvement, tels que la vitesse, l’amplitude des mouvements articulaires, la variabilité et la stabilité [10–12].

**Caractéristiques combinées :** Pour une meilleure analyse et visualisation, les chercheurs tentent de combiner les paramètres susmentionnés, tels que les angles articulaires et les forces de réaction au sol, avec les mesures anthropométriques [3].

La sélection de caractéristiques de marche précises est très importante dans l’analyse de la marche, car les résultats de la recherche dépendent fortement du choix des caractéristiques de marche les plus appropriées. Dans notre cas, nous nous concentrons spécifiquement sur la sélection des caractéristiques spatio-temporelles puisque nous nous concentrons sur la segmentation des signaux d’accéléromètre. Ces caractéristiques capturent à la fois les aspects spatiaux et temporels du modèle de marche, ce qui nous permet d’analyser et d’identifier les différentes phases et transitions au cours du cycle de marche. Par la suite, nous exploiterons ces caractéristiques sélectionnées dans le contexte d’algorithmes d’apprentissage automatique pour la détection de chute.

### 1.2.3.1 Les caractéristiques spatio-temporelles :

Plusieurs caractéristiques spatio-temporelles peuvent être utilisées pour analyser la marche [5, 6].

- **Contact initial (CI) :** C’est le contact initial du talon avec le sol pendant la marche. Il marque le début de la phase d’appui, qui est la période du cycle de la démarche lorsque le pied est en contact avec le sol. Le talonnage se caractérise par une décélération rapide du pied lorsqu’il entre en contact avec le sol, suivie d’une période de stabilisation lorsque le poids du corps est transféré sur la jambe d’appui.
- **Contact final (CF) :** le CF marque la fin de la phase d’appui et le début de la phase d’oscillation, qui est la période du cycle de la démarche lorsque le pied est soulevé du sol et propulsé vers l’avant. Toe-off se caractérise par une flexion plantaire rapide du pied, ce qui signifie que la cheville est étendue vers le bas, ce qui aide à générer la propulsion et de préparer le pied pour le prochain coup de talon. Pendant le congé, le poids du corps est transféré sur la jambe contra-latérale, et

le pied est soulevé du sol à mesure que les articulations du genou et de la hanche sont déployées.

- **Largeur de la foulée** : la distance latérale entre les points de contact initiaux de chaque pied pendant la marche. Elle mesure l'écart entre les pieds au cours d'une foulée et peut fournir des informations sur l'équilibre et la stabilité de la personne en mouvement.
- **Longueur de pas** : la distance parcourue par un pied entre un point de CI et le point de CI suivant du même pied. Elle mesure la distance parcourue par chaque pas et peut varier d'une personne à l'autre en fonction de sa taille et de sa vitesse de marche.
- **Longueur de la foulée** : La longueur de la foulée correspond à la distance totale parcourue par les deux pieds depuis le point de CI d'un pied jusqu'au point de CI suivant du même pied au cours de la foulée suivante. Elle mesure la distance totale parcourue au cours d'une foulée complète, y compris la longueur des deux pas.
- **Temps de foulée** : le temps nécessaire pour effectuer un cycle de marche complet, c'est-à-dire deux mouvements successifs du même pied.
- **Temps de pas** : le temps nécessaire pour effectuer un pas, soit deux mouvements successifs de pieds opposés. C'est une autre mesure du rythme de la marche
- **Cadence** : le nombre de pas effectués par unité de temps, généralement mesuré en pas par minute.
- **Temps de la phase d'appui** : le temps durant lequel le pied est en contact avec le sol au cours d'un cycle de marche.
- **Temps de la phase d'oscillation** : le temps durant lequel le pied n'est pas en contact avec le sol au cours d'un cycle de marche
- **Temps du double appui** : le temps durant lequel les deux pieds sont en contact avec le sol. Cela se produit deux fois au cours de chaque cycle de marche : une fois lorsque le pied arrière décolle et une autre fois lorsque le pied avant atterrit. Une durée plus longue peut indiquer une démarche plus prudente ou instable.
- **Temps du simple appui** : le temps durant lequel un seul pied est en contact avec le sol pendant un cycle de la marche. Il est généralement plus long pour le pied "dominant" ou "sain" dans les cas de boiterie ou de blessure.
- **Symétrie** : degré de similitude des caractéristiques de la marche entre les côtés gauche et droit du corps, généralement mesuré à l'aide d'un indice de symétrie.

### 1.2.3.2 Les caractéristiques statistiques :

Les caractéristiques statiques fournissent des informations importantes sur les propriétés globales du signal. En utilisant les caractéristiques statiques des signaux de l'accéléromètre, il est possible d'extraire des informations quantitatives sur la marche, d'identifier des schémas de mouvement spécifiques et de détecter des anomalies et les changements dans le modèle de marche. [10–12]

- Puissance : La puissance est utilisée pour évaluer l'énergie dépensée pendant la marche. Elle permet de comparer l'intensité de la marche entre les différents individus. Elle est calculée en prenant la moyenne des carrés des valeurs du signal.
- Maximum et minimum : Ce sont les valeurs les plus élevées et les plus basses du signal, respectivement.
- Skewness : Il s'agit d'une mesure de l'asymétrie du signal par rapport à sa moyenne. Le skewness permet de détecter des problèmes de posture et de marche asymétrique.
- Écart type et variance : Ces mesures de dispersion des données évaluent la stabilité et la régularité du mouvement pendant la marche. Une plus grande variabilité ou un écart type élevé indique des fluctuations ou des irrégularités dans le mouvement.
- Moyenne : La moyenne de l'accélération donne une indication générale du niveau d'activité physique pendant la marche. Une moyenne plus élevée est associée à une marche plus vigoureuse.

## 1.3 Analyse de la marche

L'analyse de la marche est la recherche associée à la marche humaine et révèle les mécanismes du mouvement humain en quantifiant les facteurs régissant la fonctionnalité des extrémités inférieures. Plusieurs éléments peuvent être utilisés pour mesurer et analyser objectivement la marche qui se révèle comme un outil efficace pour détecter une maladie neurodégénérative naissante, pour suivre sa progression ou aussi pour détecter les chutes.

L'étude systématique de la marche a commencé avec la description du principe de la marche par Léonard de Vinci, Galilée et Newton. En 1682, un élève de Galilée, Borelli, a décrit comment utiliser le concept du centre de gravité du corps pour parvenir à une marche équilibrée. En 1878, Edward Muybridge et Leland Stanford ont été les premiers à étudier le mécanisme de la marche. Depuis lors, la recherche sur l'analyse de la marche n'a cessé de se développer au fur et à mesure que la compréhension de la biomécanique évoluait et que de nouveaux modèles et méthodes mathématiques apparaissaient. Cependant, ce n'est que depuis 1960 que l'analyse clinique de la marche a pris de l'ampleur, grâce à l'avènement de nouvelles techniques informatiques. En conséquence, de nombreuses approches

de l'analyse de la marche ont été proposées par des chercheurs et professionnels [3].

Les approches de l'analyse de la marche humaine peuvent être classées en quatre catégories : Les approches basées sur la vision ou le traitement de l'image à l'aide d'une caméra vidéo, les approches basées sur des capteurs, les approches basées sur d'autres technologies et les approches hybrides comme le montre la figure 1.3.

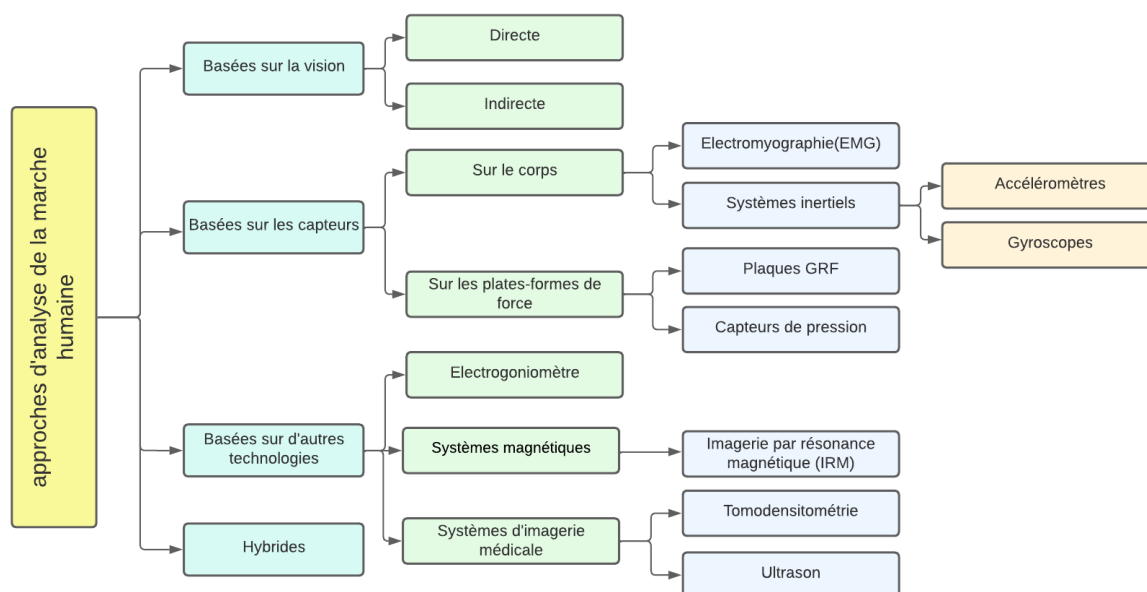


FIGURE 1.3 – Schéma d'approches de l'analyse de la marche.

### 1.3.1 Approche basée sur la vision

L'approche basée sur la vision s'appuie sur les informations visuelles capturées par des caméras pour analyser les mouvements humains, notamment pour analyser la marche et détecter les chutes. Cette approche a fait l'objet d'une attention particulière dans la recherche et les applications pratiques. Cette analyse peut être réalisée selon deux méthodes : avec (directe) ou sans marqueurs (indirecte) portés par le sujet [3].

#### Analyse basée sur les marqueurs (directe)

L'analyse de la marche à l'aide d'une approche basée sur les marqueurs consiste à placer des marqueurs réfléchissants sur des repères corporels spécifiques et à capturer leurs positions à l'aide de plusieurs caméras dans l'espace 3D. Cette méthode permet de mesurer avec précision les angles des articulations et les positions des segments, ce qui permet une analyse approfondie des paramètres de la marche. Cependant, elle nécessite un placement correct des marqueurs et un environnement contrôlé.

#### Analyse sans marqueur (indirecte)

L'analyse sans marqueur élimine le besoin de marqueurs en utilisant des caméras à détec-



tion de profondeur ou des caméras RVB avec des capteurs de profondeur. Des algorithmes analysent les données de profondeur pour estimer les positions des articulations et les paramètres de la marche. Cette approche est non invasive, permet des mouvements plus naturels et fournit une analyse en temps réel. Cependant, elle peut être légèrement moins précise et peut être affectée par l'occlusion et les facteurs environnementaux.

### 1.3.2 Approche basée sur les capteurs

L'analyse de la marche pour la détection de chute peut être réalisée à l'aide de capteurs placés soit sur le corps du sujet, soit sur le sol. Ces deux approches offrent des informations complémentaires pour comprendre et évaluer la démarche d'une personne.

**Approche basée sur des capteurs sur le corps** Cette approche implique le placement de capteurs directement sur le corps du sujet, tels que des capteurs inertiels (accéléromètres, gyroscopes, magnétomètres) ou des capteurs d'électromyographies (EMG). Les capteurs inertiels sont généralement placés sur des segments du corps pour mesurer les accélérations linéaires et angulaires, les orientations et les informations sur le champ magnétique, tandis que les capteurs EMG enregistrent l'activité électrique générée par les muscles pendant la marche. Ces capteurs fournissent des informations précieuses sur la cinématique, la coordination musculaire et les forces impliquées dans la démarche. Cette approche permet une mesure directe sur le corps et donc elle peut être utilisée dans des environnements réels et pendant des activités quotidiennes. Cependant, elle nécessite un positionnement précis des capteurs sur le corps, ce qui peut être contraignant et avoir un impact sur les mouvements naturels du sujet. En outre, les capteurs utilisés peuvent nécessiter un étalonnage et une configuration appropriés pour obtenir des mesures précises et fiables. Aussi, les données obtenues peuvent nécessiter un traitement complexe pour être interprétées.

#### **Approche basée sur des capteurs sur le sol**

Cette approche implique l'utilisation de capteurs placés sur le sol, tels que des plateformes de force ou des capteurs de pression. Les plateformes de force mesurent les forces exercées par le sujet sur le sol lors de la démarche, fournissant des informations sur la dynamique de la marche, l'équilibre et la répartition des forces. Les capteurs de pression mesurent la répartition de la pression sous les pieds, ce qui peut aider à détecter des anomalies dans la démarche.

L'analyse basée sur les capteurs sur le sol offre des mesures objectives et quantitatives des forces exercées sur le sol, permettant d'évaluer la dynamique de la marche et l'équilibre, mais elle nécessite un équipement spécialisé et peut être limitée à des environnements contrôlés.

### 1.3.2.1 Approche basée sur d'autres technologies

De nombreuses autres technologies peuvent être et sont utilisées pour analyser la démarche humaine, telles que les électro-goniomètres, les systèmes magnétiques et l'imagerie médicale.

La goniométrie est un aspect de l'analyse du mouvement qui vise à quantifier l'amplitude du mouvement d'une ou plusieurs articulations. Il existe toute une série de méthodes et d'instruments permettant d'accumuler des données goniométriques, dont l'électrogoniomètre, qui offre une grande précision, la possibilité d'une mesure dynamique et qui est indépendante du centre de l'articulation, d'où son utilisation croissante dans l'analyse de la marche [13].

Contrairement à l'approche basée sur la vision, l'approche d'analyse de la marche basée sur le système magnétique, ne dépend pas de la lumière, et ne nécessite pas de ligne de visée pour les marqueurs, car elle utilise un champ magnétique pour suivre les marqueurs ferromagnétiques. Cependant, ces systèmes sont facilement perturbés par des matériaux magnétiques [14].

Les mouvements du segment et les données anatomiques du segment du sujet peuvent être obtenus par imagerie par résonance magnétique (IRM), tomographie assistée par ordinateur (CT) et échographie. Ces données sont ensuite utilisées pour personnaliser un modèle informatique du patient auquel des données cinématiques et cinétiques peuvent être appliquées [2]. Ces systèmes sont également sensibles aux interférences

### 1.3.2.2 Approche hybride

Outre ces approches, il existe également une autre approche hybride qui utilise une combinaison de deux ou plusieurs des approches susmentionnées. [2, 15, 16]

## 1.4 Les systèmes de détection de chute portable basés sur l'analyse de la marche :

Bien que des recherches considérables aient été menées dans le domaine de l'analyse de la marche en utilisant les différentes approches mentionnées ci-dessus, il reste encore beaucoup de travail à faire si nous voulons un jour disposer d'un système d'analyse de la marche fiable et peu coûteux. Notre projet de fin d'étude s'inscrit dans cette perspective. Plus spécifiquement, nous nous intéressons à l'analyse de la marche basée sur des capteurs portés sur le corps en vue de détecter les chutes.

Les systèmes de détection des chutes peuvent être classés en deux groupes génériques :

les systèmes sensibles au contexte (SSC) et les systèmes portables. Les systèmes sensibles au contexte sont basés sur des capteurs situés dans l'environnement de l'utilisateur à surveiller. Ces solutions intègrent généralement des systèmes basés sur la vision et l'environnement, tels que des caméras, des microphones, des capteurs de vibrations, etc, qui sont déployés dans une "zone de surveillance" spécifique (par exemple, une pièce domestique) où l'utilisateur sera surveillé.

En raison de cette zone de surveillance restreinte, de la vulnérabilité des décisions de détection SSC aux événements externes (tels que les chutes d'objets, les bruits inattendus ou les changements de niveau d'éclairage) et des coûts élevés d'installation et de maintenance associées aux détecteurs de type SSC, les systèmes portables de détection des chutes ont suscité un intérêt considérable au cours de ces dernières années. Ces systèmes utilisent des accéléromètres (et d'autres capteurs de mobilité) qui sont fixés aux vêtements ou portés par les patients en tant que vêtements personnels. Les systèmes portables évaluent directement la mobilité de l'utilisateur sans altérer sa vie privée ni prendre en compte les particularités de l'environnement. Contrairement aux architectures SSC, les systèmes portables peuvent être facilement équipés d'interfaces de communication (par exemple une connexion 3G/4G ), ce qui permet de surveiller l'utilisateur presque n'importe où et à faible coût. Les systèmes portables basés sur les smartphones ont aussi été largement étudiés dans la littérature puisqu'ils intègrent une unité de mesure inertielle (IMU) (comprenant un accéléromètre, un gyroscope et, dans certains cas, un magnétomètre) tout en prenant en charge les communications sans fil multi-interfaces (Wi-Fi, 3G/4G, Bluetooth) [17, 18].

### 1.4.1 Approches de la détection des chutes dans les systèmes portable

Deux approches sont utilisées pour les systèmes portables de détection des chutes : l'approche basée sur les seuils et l'approche basée sur l'apprentissage automatique.

**Systèmes basés sur les seuils** : Dans les systèmes à seuil, une valeur maximale est prédéfinie pour les paramètres relevés par des capteurs tels qu'un accéléromètre ou un gyroscope. Une valeur mesurée au-delà de ce seuil est un indicateur d'un événement anormal. Ces systèmes sont simples à mettre en œuvre et moins gourmands en ressources informatiques. Toutefois, ils présentent certains inconvénients, un mauvais seuil peut entraîner une baisse de la précision de la détection des chutes. En outre, les seuils eux-mêmes peuvent être différents entre les sujets en raison des différences dans leurs habitudes d'activités de la vie quotidienne [19].

**Systèmes basés sur l'apprentissage automatique** : Bien que les systèmes basés

sur les seuils soient populaires en raison de leur faible coût de calcul, ils pourraient être sujets à davantage de faux positifs et de faux négatifs étant donné qu'ils ne sont pas toujours faciles à mettre en œuvre. Par conséquent, Les systèmes basés sur l'apprentissage automatique pour la détection de chute ont fait l'objet de nombreuses recherches. Ces systèmes utilisent des algorithmes et des modèles d'apprentissage automatique pour analyser les données des capteurs et détecter les chutes. Ils collectent des données, extraient des caractéristiques significatives, entraînent des modèles sur des ensembles de données annotées, et utilisent ces modèles pour détecter les chutes en temps réel. Ils offrent une capacité d'apprentissage et d'adaptation, ainsi que la possibilité d'intégrer différentes sources de données pour améliorer la précision de la détection. Ces systèmes diffèrent les uns des autres par de multiples facteurs, les capteurs utilisés, l'emplacement des capteurs, les algorithmes appliqués, l'ensemble de données, les paramètres de performance contrôlés, etc. Des recherches approfondies ont été menées sur l'efficacité de divers algorithmes d'apprentissage automatique pour la détection des chutes. Par exemple, de Quadros et al. [20] comparent les mécanismes basés sur les seuils et les mécanismes basés sur l'apprentissage automatique pour la détection des données générées par l'accéléromètre, le gyroscope et le magnétomètre. L'article conclut que le mécanisme basé sur l'apprentissage automatique donne de bien meilleurs résultats que les solutions basées sur le seuil.

### 1.4.2 Défis des systèmes de détection des chutes portables

Les systèmes de détection des chutes portables sont confrontés à plusieurs défis.

#### **La garantie des performances en conditions réelles :**

La garantie des performances en conditions réelles constitue un défi de taille. Le système doit être robuste et précis dans divers environnements, du terrain et des activités de l'utilisateur. Les conditions réelles introduisent des complexités qui peuvent affecter les performances du système, et il est essentiel de relever ces défis pour assurer la fiabilité de la détection des chutes.

#### **L'acceptation des systèmes portables de détection des chutes**

Les dispositifs portables doivent être confortables, non intrusifs et faciles à utiliser. L'acceptation par l'utilisateur joue un rôle essentiel dans l'adoption et l'utilisation à long terme de ces systèmes. La conception d'interfaces conviviales, la prise en compte des préoccupations en matière de respect de la vie privée et l'intégration du retour d'information des utilisateurs sont des facteurs clés pour améliorer l'acceptation et la satisfaction des utilisateurs.

#### **La consommation d'énergie**

La consommation d'énergie est une préoccupation importante pour les systèmes portables

de détection des chutes. Les appareils doivent avoir une batterie d'une longue durée de vie et une gestion efficace de l'énergie pour assurer une surveillance continue sans recharges fréquentes. Il faut pour cela optimiser les composants matériels et logiciels afin de minimiser la consommation d'énergie.

## 1.5 L'apprentissage automatique (Machine Learning ML) dans l'analyse de la marche aux fins de la détection des chutes

Dans cette section, nous décrivons les techniques d'apprentissage automatique utilisées dans l'analyse de la marche aux fins de la détection des chutes. La figure 1.4 résume les principales étapes d'une chaîne d'analyse de marche. Les différents blocs sont détaillés dans ce qui suit.



FIGURE 1.4 – Chaîne d'analyse de la marche.

### 1.5.1 Acquisition et pré-traitement de données :

Le premier bloc d'une chaîne d'analyse de la marche concerne l'acquisition de données brutes provenant de capteurs. Ces données nécessitent généralement un prétraitement afin d'éliminer les différents artefacts affectant le signal utile. Cela dit, il est essentiel de disposer d'un ensemble de données adéquates pour l'analyse de la marche en vue de détecter les chutes, dans le but de développer des algorithmes, d'évaluer des systèmes et de comparer différentes approches. Cela permet aux chercheurs d'améliorer la précision et la fiabilité des systèmes de détection des chutes et de faire progresser cet important domaine de recherche. La littérature fait état de nombreux systèmes portables de détection des chutes, mais très peu d'ensembles de données sont accessibles au public. Dans ce qui suit, nous présentons une vue générale des ensembles de données de détection de chute basés sur la marche :

**L'ensemble de données du DLR (Centre aérospatial allemand) :** C'est une collecte de données provenant d'une unité de mesure inertielle (IMU) portée à la ceinture

par 16 personnes (6 femmes et 5 hommes) âgées de 23 à 50 ans.

Sept activités sont prises en compte (marcher, courir, se tenir debout, s’asseoir, s’allonger, tomber et sauter). Les types de chute n’ont pas été distingués. [21]

**L’ensemble de données UMAFALL** : C’est un ensemble de données comprenant trois types de chutes et huit ADL obtenues à partir d’un smartphone porté dans la poche de la cuisse droite et quatre capteurs portables portés à la cheville, à la taille, au poignet droit et à la poitrine. Les sujets ont effectué au moins trois essais de chaque activité dans un environnement domestique [22].

**L’ensemble de données MobiFall** : Elle a été développée par le laboratoire d’informatique biomédicale et d’e-santé de l’Institut d’Enseignement Technologique de Crète. Ils ont capturé des données générées par des capteurs inertiels d’un smartphone (accéléromètre 3D et gyroscope) placé dans la poche d’un pantalon. Les 24 sujets, dix-sept hommes et sept femmes, âgés de 22 à 47 ans, ont effectué entre 3 à 6 essais pour chaque activité. Les auteurs ont pris en compte quatre types de chutes et neuf activités différentes de la vie quotidienne [23].

**L’ensemble de données tFall** : Elle a été développée par EduQTech (Education, Quality and Technology) à l’Université de Saragosse, elle a recueilli les données de dix participants, trois femmes et sept hommes, dont l’âge varie entre 20 et 42 ans. Ils ont obtenu les données par des capteurs inertiels de deux smartphones portés par les sujets dans la vie quotidienne [21].

Cependant, il est important de noter qu’aucun de ces ensembles de données n’inclut spécifiquement les personnes âgées, ce qui est l’un des principaux objectifs de notre étude. Comprenant les défis et les caractéristiques uniques associés aux chutes dans la population âgée, nous avons mené une recherche approfondie pour identifier les ensembles de données qui répondaient à cette exigence. Voici les bases de données que nous avons trouvées qui se concentrent spécifiquement sur les personnes âgées et incluent des informations pertinentes sur les chutes :

**L’ensemble de données SisFall** : C’est un ensemble de données de chutes et d’ADL obtenus à l’aide d’un dispositif embarqué doté de deux accéléromètres et d’un gyroscope. L’appareil était positionné au niveau de la taille. L’ensemble de données a été généré avec la collaboration de 38 participants, dont 15 personnes âgées et 23 jeunes adultes âgés de 19 à 75 ans ( $40.16 \pm 21.33$ ) [24].

**L’ensemble de Données LTMM** : C’est un ensemble de données dérivées de l’accélération et du gyroscope sur trois jours dont 71 personnes âgées entre 65-87 ans ont été étudiés. Les participants ont effectué une marche sur le tapis roulant pendant une minute en portant une ceinture avec un capteur léger (DynaPort,  $87 \times 45 \times 14$  mm, 74 g). Le capteur a été placé sur le bas du dos et se compose d’un accéléromètre et un gyroscope

triaxiaux. Les sujets ont été classés en fallers et non-fallers sur la base de leur déclaration de chutes antérieures. Après avoir effectué des tests en laboratoire, les participants ont porté la ceinture pendant trois jours consécutifs [25].

**L'ensemble de données Gstride :** C'est un ensemble de données métriques et de mouvements pertinents sur les personnes âgées pour l'évaluation de leur état de santé. Les données correspondent à 163 patients, 45 hommes et 118 femmes, âgés de 70 à 98 ans. GSTRIDE est spécialement axé, sans s'y limiter, sur l'étude des chutes et non chutes des personnes âgées. L'objectif principal de cet ensemble de données est de permettre l'étude de ces différentes populations. En incluant les résultats des tests d'évaluation de la santé et des questionnaires, ainsi que les données inertielles et spatio-temporelles, les chercheurs peuvent analyser différentes techniques d'identification des chuteurs [26].

### 1.5.2 La segmentation de la marche

La segmentation des enregistrements de la marche à partir de dispositifs portables et non portables est une étape fondamentale pour l'étude du risque de chute. Le CI et le CF sont deux événements indispensables pour cette segmentation. C'est des points de repère pour segmenter les données des capteurs, en cycles de marche ou en pas individuels. En identifiant ces événements dans le signal de l'accéléromètre, on peut extraire des caractéristiques spécifiques à la marche. Ces caractéristiques peuvent être utilisées pour la reconnaissance de modèles de marche, la détection d'anomalies de la marche ou l'évaluation du risque de chute.

Plusieurs méthodes peuvent être utilisées pour la segmentation automatique, Les plus couramment utilisées dans la littérature scientifique sont les suivantes [27] :

1. **Méthode de Seuil :** L'algorithme de cette méthode consiste à fixer un seuil basé sur l'amplitude du signal d'accélération. Lorsque le signal dépasse ce seuil, il indique un contact avec le sol, soit CI où CF.
2. **Méthode de détection des pics (peak detection) :** cette méthode repose sur l'identification des pics dans le signal d'accélération. Les pics correspondent généralement aux moments de CI et de CF.
3. **Méthode de passage à zéro (Zero crossing) :** l'algorithme de cette méthode détecte les passages à zéro du signal d'accélération. Lorsque le signal traverse la valeur zéro, cela peut indiquer un changement de phase entre CI et CF.

### 1.5.3 Extraction des caractéristiques

L'extraction de caractéristiques est un processus par lequel un ensemble de données initiales est réduit à des caractéristiques correspondant à des informations pertinentes. Dans le domaine de l'analyse de la marche, un grand nombre de caractéristiques a été proposé [2.8](#)

### 1.5.4 La classification

Dans l'apprentissage automatique, la classification fait référence à un problème de modélisation prédictive où une étiquette de classe est prédite pour un exemple de données d'entrée. Plusieurs algorithmes de classification ont été proposés dans la littérature. Dans le cadre de ce travail, nous avons utilisé plusieurs algorithmes de classification, notamment SVM 'svm' , KNN, DT, RF, DNN, et CNN1d [\[28\]](#).

## 1.6 Travaux connexes

Dans cette partie, nous passons en revue les détails des études les plus pertinentes et les plus récentes, qui utilisent des capteurs inertiels et des méthodes d'apprentissage automatique pour évaluer le risque de chute. Ils sont répertoriés dans le tableau ci-dessous. Les objectifs et les méthodes de chaque étude, le type de capteur utilisé et le nombre de participants sont inclus dans ce tableau. Les avantages et les inconvénients des différentes études sont également mis en évidence [\[29\]](#).

**TABLEAU 1.1** – Résumé des études sélectionnées. Les forces et les faiblesses sont basées sur la perception de l'auteur.

Méthodologie	Capteurs	Sujets	Forces	Faiblesses
Extraction des caractéristiques de la démarche (à partir des données brutes), sélection des caractéristiques (sélection d'emballage avant), classification en risque élevé/faible (RBFN, SVM, kNN, NB) <a href="#">[30]</a>	10 accéléromètres sur divers emplacements du corps	15 à risque élevé, 5 à faible risque	Haute précision de classification (95 %). Utilisation de méthodologie d'apprentissage automatique.	Nombre limité de sujets à faible risque. Très contraignant (les capteurs sont câblés et collés sur la peau).



Extraction des caractéristiques et des paramètres de la démarche (Méthodes d'autres études), classification en chuteurs/non-chuteurs (DT, LR) [31]	1 accéléromètre sur le tronc	19 chuteurs, 27 non-chuteurs	Étude prospective de 12 mois.	Faible sensibilité (58 %), bien qu'elle augmente à 74 % dans le cas où les scores d'activité physique (collectés pendant 12 mois) sont inclus.
Extraction des caractéristiques temporelles et spectrales de la démarche lors des tests fonctionnels (à partir des données brutes), régression pour inférer le nombre de chutes (LR) [32]	1 accéléromètre sur le tronc	22 chuteurs, 46 non-chuteurs	Grand nombre de caractéristiques (126) combinant l'accélération et les caractéristiques des tests fonctionnels. Classes multiples (# de chutes).	Aucun paramètre spatial considéré. Non ciblé vers les troubles neurologiques.
Extraction des caractéristiques de la démarche (à partir des données brutes, par enjambée), classification en chuteurs/non-chuteurs (DT) [33]	1 IMU en emplacement non spécifié	20 chuteurs, 20 non-chuteurs	Étude prospective de 2,4,8 mois. Considère chaque cycle de démarche comme une instance séparée (ensemble de données plus grand).	Faible sensibilité dans la classification à long terme (33,6 %). Les instances par cycle ne peuvent pas représenter la variabilité intra-sujet.

Le tableau 1.2 présenté ci-dessous résume les performances atteintes dans les travaux connexes qui ont utilisé des capteurs inertiels et des méthodes d'apprentissage automatique pour évaluer le risque de chute dans ces dernières années.

Année	Algorithme	Capteur	Emplacement de capteur	Métrique
2015 [34]	SVM	Capteur de Force	Sous le pied	Exactitude : 90% Sensibilité : 90%
2015 [35]	CNN	Acceleromtre	Poche,taille	Exactitude : 93.8%
2016 [36]	SVM, MLP, CNN	IMU	Pas mentionné	Exactitude : 97.01%
2016 [37]	LSTM	MIMU, Acceleromtre	Dos, Hanche	Sensibilité : 99%
2016 [38]	SVM	IMU	Taille	Exactitude : 97.0%
2017 [39]	ANN	IMU	Poche	RMSE :0.02266 MAE :0.0174
2017 [40]	CNN	Acceleromtre	Poche	Exactitude : 96.6%
2017 [41]	DNN	Acceleromtre	Pas mentionné	Exactitude : 96.9-99.5%
2017 [42]	SVM, RF	Acceleromtre	Hanche	Exactitude : 99%
2018 [43]	CNN	IMU	Bas du dos	Exactitude : 90.6%
2018 [44]	SVM	Capteur de Force	Sous le pied	Exactitude : 97.01%
2018 [45]	SVM	Acceleromtre	Taille	Exactitude : 97.34% Sensibilité : 98.52%
2019 [46]	RF	Acceleromtre, Gyroscope	Poche	Exactitude : 98%
2019 [47]	CNN	Acceleromtre, Gyroscope	Poche	Exactitude : 91.8%
2019 [48]	SVM	Acceleromtre	Pas mentionné	Exactitude : 92%
2020 [49]	ANN	IMU	Cheville	RMSE : 0.004
2022 [50]	SVM	Accelerometre	Bas du dos	Exactitude : 59% F1-score : 91%

**TABLEAU 1.2** – Performances récentes dans l'évaluation du risque de chute à l'aide de capteurs inertiels et d'apprentissage automatique : Études connexes

## 1.7 Conclusion

Ce chapitre servira de base solide pour la compréhension des principes et des techniques de base de l'analyse de la marche. Il jettera les bases nécessaires pour la suite de notre mémoire, où nous explorerons plus en détail les aspects spécifiques de l'analyse de la marche et de la détection des chutes.

# Chapitre 2

## Méthodologie

### 2.1 Introduction

Ce chapitre se concentrera sur la méthodologie utilisée dans notre étude, en mettant l'accent sur deux aspects majeurs. Nous explorerons les différents algorithmes de segmentation qui sont essentiels dans le processus d'analyse de la marche. Ces algorithmes nous permettent de diviser un ensemble complexe de données de marche en segments plus gérables et significatifs pour l'analyse, capturant ainsi les événements clés de la marche. Nous aborderons par la suite l'application de divers algorithmes d'apprentissage automatique pour classer les incidents de chute et de non-chute.

La figure 2.1 illustre la méthodologie utilisée dans notre étude, mettant en évidence les différentes étapes.

### 2.2 Les algorithmes de la segmentation

L'analyse de la marche humaine nécessite des méthodes précises pour extraire et interpréter les informations contenues dans les signaux de mouvement. La segmentation de la marche est l'une des étapes clés de cette analyse.

Nous présentons ici les algorithmes de segmentation utilisés pour analyser de manière continue la marche humaine.

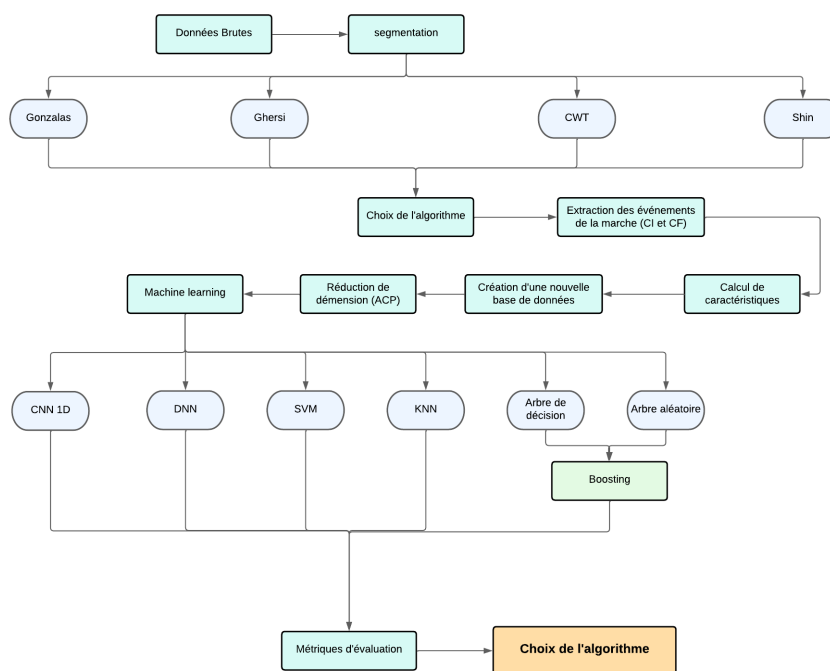


FIGURE 2.1 – Schéma de la méthodologie complète.

### 2.2.1 Algorithme CWT

CWT (Continuous Wavelet Transform) est une technique mathématique pour l'analyse et le traitement des signaux, elle est généralement utilisée pour étudier les coupures de fréquence, discontinuité dans le temps et aussi pour détecter les différentes caractéristiques telles que les bords et les pics.

Cette technique effectue des opérations comme la différentiation pour détecter les changements d'un signal au cours du temps. Cette différentiation s'effectue en décomposant le signal en un ensemble d'ondelettes [51]. Une ondelette est une forme d'onde limitée en durée avec une moyenne nulle qui a les propriétés d'un filtre passe haut, ce qui permet d'amplifier les composantes hautes fréquences et d'éliminer les composantes basses fréquences. Elle est définie par la formule suivante :

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right) \quad a, b \in \mathbb{R} \quad (2.1)$$

a et b sont appelés les paramètres de dilatation et de translation respectivement. a est estimé par la formule suivante :

$$a = \frac{F_c}{F_a \cdot \Delta} \quad (2.2)$$

$f_c$ ,  $f_a$  sont les fréquences centrales et dominantes de l'ondelette respectivement.

Le signal est convolué avec des ondelettes de différentes échelles et les coefficients résultants sont utilisés pour construire une représentation de 2 dimensions dans l'espace temps fréquence (scalogram). Le signal résultant est ensuite utilisé pour extraire les caractéristiques et analyser les changements de signal au cours du temps. le CWT d'un signal est donné par la formule suivante :

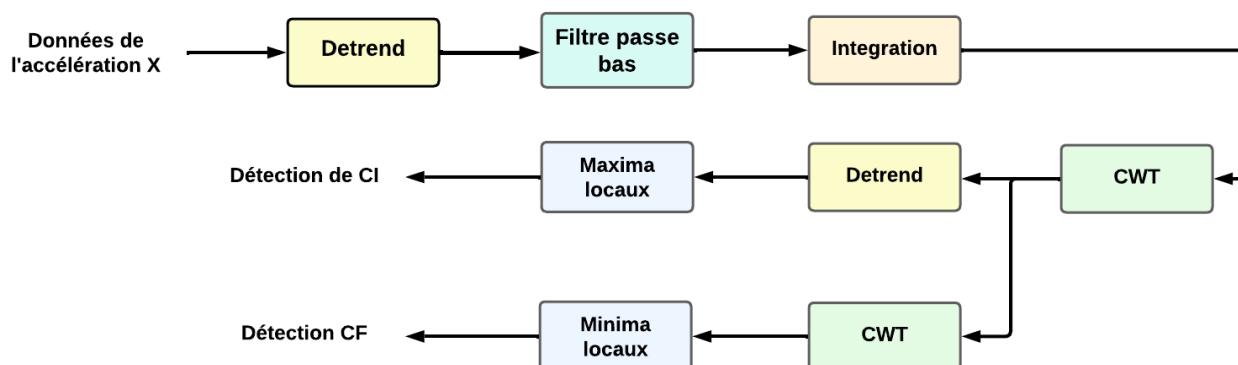
$$CWT_{(a,b)} = \langle f, \psi_{a,b} \rangle = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} f(t - \tau) \psi \left( \frac{t - b}{a} \right) d\tau \quad (2.3)$$

### La segmentation de signal utilisant CWT :

Il est prouvé dans les études précédentes que CWT [52] donnent des très bons résultats en ce qui concerne la détection des deux événements de la marche CI et FC.

CI et CF sont extraits à partir de l'accélération AP (accélération antéro-postérieure).

La figure ci-dessous 2.2 illustre les étapes de l'algorithme suivi pour extraire le CI et FC.



**FIGURE 2.2** – Détection CF et CI à l'aide de l'algorithme de la transformée en ondelettes continues.

- *Detrend* : La suppression de la tendance du signal est une étape de pré-traitement couramment utilisée, en particulier pour l'analyse des séries temporelles. Elle a pour but de supprimer le comportement à long terme du signal, qui ne présente pas d'intérêt dans notre application. La suppression des tendances nous permet de nous concentrer sur les variations à court terme du signal, ce qui est plus pertinent pour la détection de CI et FC. Il existe plusieurs méthodes pour faire le detrending de signal utilisant différentes techniques statiques, linéaire, filtre de Baxter-king, polynomial detrending, moving average detrending, and wavelet-based detrending. Dans notre cas, on a utilisé la suppression de la tendance linéaire :

- À l'aide de la régression linéaire, on ajuste une ligne droite aux données qui

décrivent le mieux la tendance ou le modèle.

- On calcule la différence entre les données et la ligne ajustée et on la soustrait des données d'origine pour obtenir les données sans tendance.
- *LPF* : Le signal est ensuite passé par un filtre passe bas pour éliminer le bruit qui est un filtre de butterworth de second ordre à une fréquence de coupure de 10 hertz.
- *Intégration et différentiation* : Le signal pré-traité est ensuite intégré et différencié en utilisant CWT.

En appliquons le CWT sur le signal, nous obtenons une représentation du signal dans le domaine temps-fréquence, où chaque point dans la matrice 2D résultante représente l'énergie du signal à un moment et une fréquence particulière. On extrait les CI qui représentent les minima locaux.

En appliquant de nouveau le CWT sur cette matrice résultante, nous pouvons analyser plus en détail le contenu temps-fréquence du signal, et l'on identifie les CF qui représentent les maxima locaux.

### 2.2.2 Algorithme de Shin

Shin et Park [53] ont proposé un algorithme adaptatif de détection de pas (Step détection) dans l'analyse de la marche. Il est basé sur les données d'accélération collectées à partir de capteurs portables. Figure 2.3 présente la procédure de l'algorithme sous forme schématique.

**Magnitude de l'accélération** : donnée par l'équation 2.25.

**Somme de fenêtre glissante** : ou "Sliding window summation" Afin de réduire le bruit dans les données, une sommation par fenêtre coulissante est utilisée.

$$SWS(k) = \sum_{t=k-N+1}^k a_{\text{NORM}}(t) \quad (2.4)$$

$N$  est généralement appelé la "taille de la fenêtre". La taille de la fenêtre fait référence au nombre de points de données qui sont inclus dans chaque sommation ou calcul individuel lorsque la fenêtre "glisse" sur l'ensemble de données.

$$N = T \cdot f$$

$T$  est la durée de la fenêtre en secondes.

$f$  est la fréquence à laquelle les points de données sont échantillonnés.

**Technique d'accélération différentielle :** La technique du différentiel d'accélération est utilisée pour éliminer l'effet de la gravité sur les données d'accélération.

$$a(k) = SWS(k + N) - SWS(k) \quad (2.5)$$

**Zero crossing :** Il s'agit du point où la valeur de la norme de l'accélération passe de positive à négative ou vice versa. Le passage du point zéro de négatif à positif de ce signal est considéré comme le moment de Contact initial.

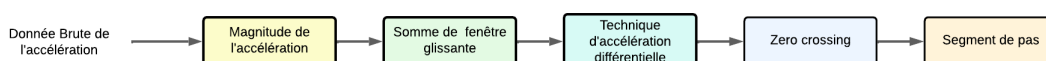


FIGURE 2.3 – Algorithme de Shin.

### 2.2.3 Algorithme de Gonzalez

L'algorithme de Gonzalez [54] est une méthode couramment utilisée pour détecter les événements de CI et de CF dans l'analyse de la marche. Il fonctionne en détectant les points dans une série de données où le pied entre en contact avec le sol (CI) et où il quitte le sol (CF).

L'algorithme comprend deux étapes principales :

- La phase de pré-traitement.
- La phase de détection.

— *La phase de pré-traitement :*

À l'étape du pré-traitement, les données brutes de l'accélération antéro-postérieure sont filtrées à l'aide d'un filtre passe-bas de 11e ordre pour éliminer tout bruit à haute fréquence.

— *La phase de détection :*

Au stade de la détection, l'algorithme identifie d'abord les maxima locaux dans le signal qui dépassent un seuil prédéfini. Ces maximums locaux sont ensuite soumis à un ensemble de critères pour déterminer lesquels correspondent aux événements CI et FC. Ces critères comprennent la vérification d'une séparation minimale entre les événements CI et FC, d'une durée minimale et maximale pour les phases CI et FC, et d'une différence maximale autorisée de hauteur entre les événements CI et CF détectés.

L'algorithme est basé sur la localisation de passages par zéro du positif au négatif de la sortie du filtre FIR appliqué à l'entrée de l'accélération antéro-postérieure (Zero Crossing). Chaque fois qu'un ZC est trouvé, nous estimons numériquement la surface délimitée par des valeurs positives de notre signal. Si cette surface est en dessous d'un seuil donné, le ZC est rejeté, car il n'est pas considéré comme associé à un événement CI. Sinon, l'accélération antéro-postérieure brute est analysée comme suit :

- Localiser le pCI associé à l'événement IC, donnant ainsi son temps estimé d'occurrence. À cette fin, nous définissons une fenêtre de recherche pour les maxima.
  - La fenêtre de recherche peut présenter plusieurs pics locaux. Pour identifier le pCI lié à l'événement IC, nous appliquons les règles heuristiques suivantes :
    - L'accélération verticale doit être supérieure à la gravité.
    - Le CI doit se produire avant que l'accélération verticale atteigne 99 % de son niveau local.
    - Si plusieurs pics remplissent ces conditions, le plus proche de ZC est sélectionné.
  - Chaque détection de CI lance une recherche du CF dans l'accélération verticale. À partir de l'échantillon correspondant au IC
  - Les échantillons entrants sont traités, à la recherche du premier minimum local qui vient après le premier maximum local détecté
- Une fois les événements CI et CF ont été détectés, ils sont utilisés pour calculer divers paramètres de la marche tels que la longueur de pas, le temps de pas, etc.

L'algorithme de Gonzalez est une méthode largement utilisée et efficace pour détecter les événements CI et CF dans l'analyse de la marche, et est souvent utilisé en combinaison avec d'autres techniques pour une analyse plus complète.

### 2.2.4 Algorithme de Gherzi

Un nouveau système d'extraction et d'analyse automatique des cycles de marche à partir des signaux d'accélération de la partie inférieure du tronc, combinant les connaissances des stratégies précédentes avec une fonction de déformation temporelle Dynamique (DWT) [55], est présenté. Les performances ont été testées sur des signaux de marche provenant de bases de données publiques.

Cette méthode se concentre uniquement sur l'accélération antéro-postérieure plutôt que sur la magnitude de l'accélération composite. Elle cherche une simple représentation



pour cette accélération  $a_{AP}$  qui ne nécessiterait pas de pré-entraînement, Gherzi et Al [56] ont proposé le modèle de  $a_{AP}$  sous forme d'une onde triangulaire modifiée. L'étape initiale de cette méthode est la détection des événements de la marche, CI et FC. L'étape suivante sera une étape de correction, le modèle peut sauter/répliquer des points de ces deux événements (CI<sub>x</sub> et FC<sub>x</sub>), et donc suivant le cycle de la marche, on aura des pas surdivisés (pas supplémentaires) ou des pas manqués.

La dernière étape sera une étape d'optimisation où pour ce travail une implémentation de la méthode de González et al [57] a été choisie, étant donné que cette méthode a été testée sur un large éventail de populations.

L'optimisation de CI<sub>x</sub> aboutit à la segmentation finale des pas et des cycles de marche, comme le propose cette méthode.

La figure 2.4 ci-dessous présente l'algorithme de traitement de signal.

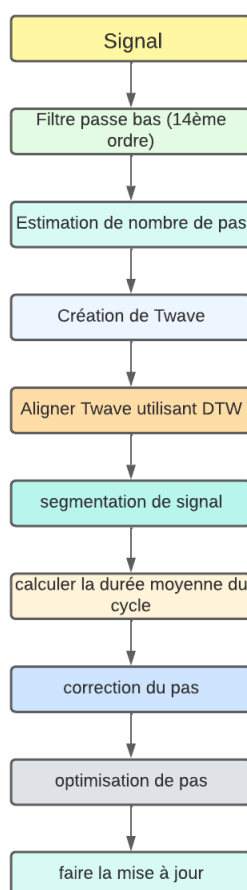


FIGURE 2.4 – Détection CF et CI à l'aide de l'algorithme de Gherzi.

- *Filtrage* : Au début du processus, un segment a-x est sélectionné par l'utilisateur et filtré à l'aide d'un filtre FIR passe-bas de 14e ordre ( $f_{\text{coupure}} = 4$  Hz). Avant

chaque ajustement, une onde triangulaire (Twave) est synthétisée. Trois paramètres sont déterminés :

1. *Fréquence moyenne de pas* : Elle est obtenue à partir de la valeur absolue de la transformée de Fourier rapide (FFT) de l'accélération antéro-postérieure (AP), en recherchant la fréquence de la valeur maximale qui est inférieure à 3 Hz (définie pour les conditions de la marche).

$$f_{moyenne} = \text{fmax}\{|FFT(aAP)|\} \quad (2.6)$$

Le nombre attendu de pas ( $n\_pas$ ) peut être estimé à partir de  $f\_moyenne$  et de la durée du segment ( $r\_taille$ ).

$$n\_pas = \frac{r\_taille}{f\_moyenne} \quad (2.7)$$

2. *L'amplitude de l'onde* : est la plage de l'accélération antéro-postérieure (AP).
3. *La valeur moyenne de l'onde* : est la moyenne de l'accélération antéro-postérieure (AP).

— *Création d'une onde triangulaire* : L'équation de l'onde triangulaire synthétisée Twave est la suivante, où  $Ts$  est le temps d'échantillonnage :

$$T_{\text{wave}} = \begin{cases} \min(a_{AP}) t = Ts, 5Ts, 9Ts, \dots, (4(n_{\text{pas}}) - 3) Ts \\ \text{moyenne}(a_{AP}) t = 2Ts, 4Ts, 6Ts, \dots, (4n_{\text{pas}}) Ts \\ \max(a_{AP}) t = 3Ts, 7Ts, 11Ts, \dots, (4(n_{\text{pas}}) - 1) Ts \end{cases} \quad (2.8)$$

— *Application de l'algorithme de Déformation Temporelle Dynamique* : Afin d'obtenir un ajustement adéquat pour l'accélération antéro-postérieure (AP) avec une onde triangulaire, la technique DTW a été sélectionnée [55].

DTW est un algorithme d'évaluation de la similarité entre deux signaux, qui est indépendant du nombre d'échantillons dans chacun d'eux. L'algorithme cherche la version déformée de l'onde qui s'ajuste le mieux à la référence.

Il s'agit de transformer l'accélération antéro-postérieure (AP) en une forme d'onde triangulaire, où les pentes négatives correspondent à des décélérations de l'AP. Les maxima locaux de l'onde triangulaire sont des estimations de IC, et les minima locaux représentent des approximations de FC.

*Principe* : La méthode DTW utilise la matrice des distances pour comparer deux séries temporelles, notées  $t$  et  $r$ , de longueurs respectives  $m$  et  $n$ . Cette matrice, que nous notons  $D$ , est calculée selon la procédure suivante :

1. La matrice  $D$ , de taille  $m$  par  $n$ , est utilisée pour stocker les distances DTW entre les sous-séquences  $(t_{_i}, \dots, t_{_m})$  et  $(r_{_j}, \dots, r_{_n})$ . Chaque élément d'indice  $(i, j)$  de la matrice, noté  $D(i, j)$ , représente cette distance DTW.
2. On calcule chacune des valeurs de la matrice d'une manière récursive utilisant la formule suivante :

$$D(i, j) = |t(i) - r(j)| + \min\{D(i + 1, j), D(i + 1, j + 1), D(i, j + 1)\} \quad (2.9)$$

3. Une fois la matrice obtenue, la distance peut être calculée en effectuant une simple somme. On commence par la première case en haut à droite de la matrice  $D$  et l'on se déplace vers la case voisine ayant la valeur la plus faible parmi la case de gauche, la case du bas et la diagonale gauche. Les valeurs correspondantes aux cases du chemin sélectionné sont additionnées.

**Le pseudo code :**

- Entrées :  $x_{1:N}$  et  $y_{1:M}$

- La matrice du coût :  $\mathbf{D} \in \mathbb{R}^{(N+1) \times (M+1)}$

- Initialisation :

$$\text{for } i = 1 \text{ to } N : D_{i,0} = \infty$$

$$\text{for } j = 1 \text{ to } M : D_{0,j} = \infty$$

- Calculer la matrice du coût : for  $i = 1$  to  $N$  : for  $j = 1$  to  $M$  :

$$D_{i,j} = d(x_i, y_j) + \min \begin{cases} D_{i-1,j-1} & \text{(Correspondance)} \\ D_{i-1,j} & \text{(insertion)} \\ D_{i,j-1} & \text{(suppression)} \end{cases}$$

- Obtenir l'alignement : Remonter de  $D_{N,M}$  vers  $D_{0,0}$  .

En trouvant le chemin de coût minimal à travers la matrice de coût, il nous indique quel est le moyen le moins cher selon notre métrique de distance pour aligner ces deux signaux.

DWT nous indique aussi quel est le coût global de l'alignement de ces deux signaux, qui est la toute dernière valeur de notre matrice du coût.

La définition de (insertion, suppression et correspondance), c'est ce que nous devons faire au signal triangulaire pour le changer en A-ap. Voici une explication de chaque notion :

— *Insertion* : Au cours de cette étape, l'algorithme insère ou ajoute des éléments supplémentaires au signal triangulaire pour qu'il corresponde à mieux au signal

de l'accélération.

— *Suppression* : L'étape consiste à supprimer des points du signal triangulaire pour mieux s'aligner sur l'accélération antéro-postérieure (AP).

— *Mise en correspondance* : La mise en correspondance signifie que les deux signaux sont bien alignés et pas besoin de faire une insertion ou une dilution.

En effectuant ces étapes, l'algorithme DWT trouve l'alignement optimal entre les deux séquences.

— *Correction des pas* : Avec l'estimation initiale de  $CI_x$ , les pas ( $CI_x : CI_{x+1}$ ) et les cycles ( $CI_x : CI_{x+2}$ ) sont segmentés. La moyenne des pas et des cycles de marche pour aAP est calculée. Les estimations de  $CI_x$  sont ensuite ajustées à l'aide des stratégies de correction suivantes :

1. Un pas est considéré comme subdivisé (pas supplémentaire) lorsque sa période (distance entre  $CI_x$  et  $CI_{x+1}$ ) est inférieure à 50% du temps de pas moyen de l'ensemble de l'enregistrement pour le même sujet, dans ce cas, le  $CI_{x+1}$  est éliminé.

2. On parle de pas manqués lorsque le segment entre  $CI_x$  et  $CI_{x+1}$  comprend deux pas. Chaque pas est comparé à la moyenne des pas et des cycles. Si la somme de la différence au carré entre le signal et le cycle complet moyen ( $C_{diff}$ ) est inférieur à la différence avec l'étape moyenne ( $S_{diff}$ ), le segment est subdivisé en deux étapes. Une nouvelle vague triangulaire est créée ( $n_{pas} = 2$ ) et DTW est réalisé sur cette section. Les résultats sont rejetés si les étapes dérivées sont considérées comme trop divisées.

— *Optimisation du CI* : Après la détection des pas, les signaux peuvent être re-filtrés. Les méthodes de détection des points caractéristiques peuvent être appliquées, par exemple, en adaptant le seuil de passage à zéro (ZC) aux valeurs centrales de aAP. Un filtre Butterworth de 4e ordre avec une fréquence de coupure de 20 Hz a été appliqué aux signaux, et les premiers maxima locaux dans AAP avant ZC ont été sélectionnés comme des estimations de CI finaux.

La figure 2.2.4 ci-dessous explique les algorithmes de correction et d'optimisation des pas [56] .

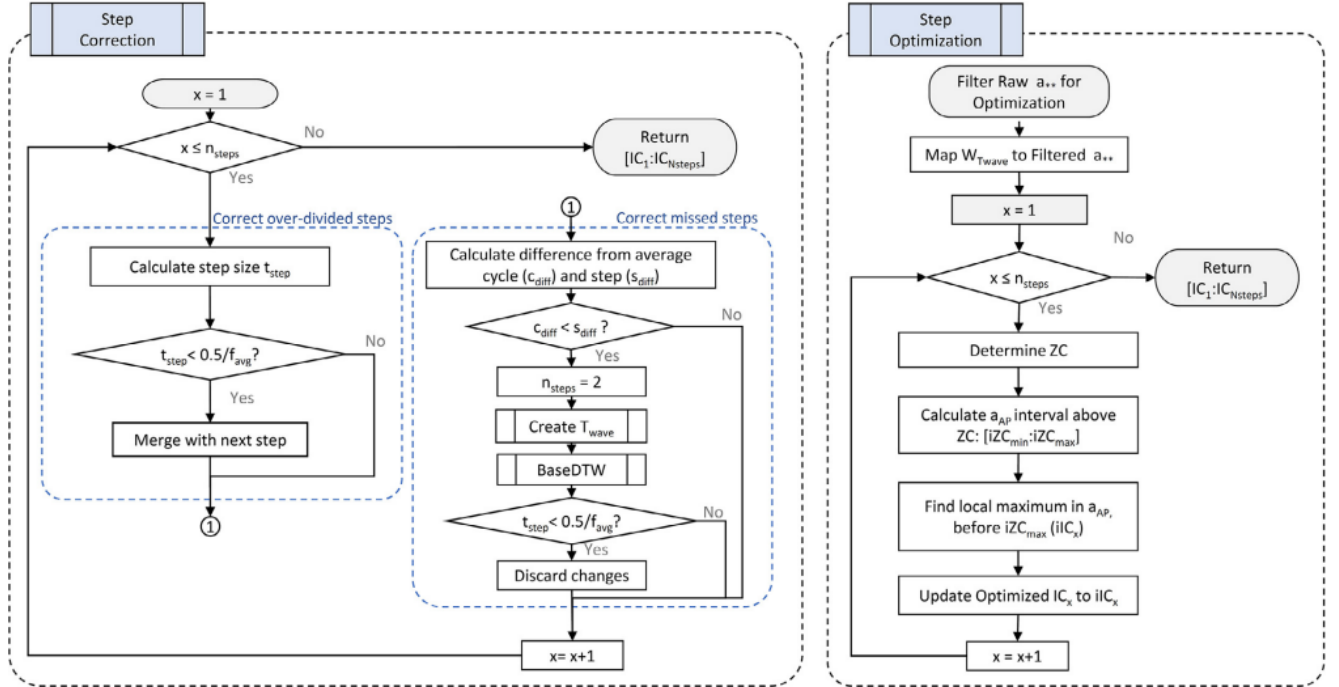


FIGURE 2.5 – Algorithmes de correction et d'optimisation .

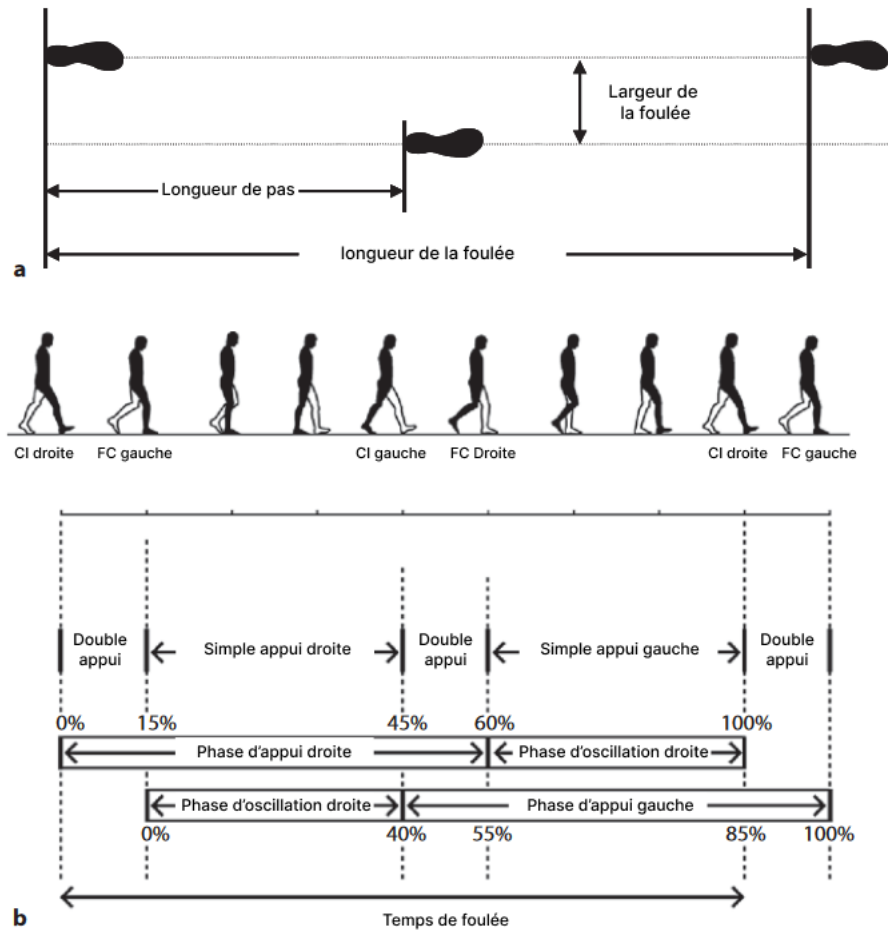
## 2.3 Extraction des caractéristiques

L'extraction des caractéristiques est une étape cruciale dans l'analyse de la marche humaine. Elle permet de quantifier et de représenter les informations pertinentes contenues dans les signaux de mouvement. Dans cette section, nous présenterons les différentes catégories de caractéristiques utilisées dans notre étude.

### 2.3.1 Caractéristiques spatio-temporelles

Les caractéristiques spatio-temporelles captent les informations relatives à la position et au temps lors de la marche. Elles permettent de décrire les mouvements et les changements de posture pendant chaque cycle de la marche.

Pour calculer les caractéristiques spatio-temporelles, il est nécessaire d'identifier au moins 1 cycle d'un pied (droit et gauche), ce qui correspond à l'identification de 4 événements continus de CI et de 4 événements de CF comme le montre la figure 2.6.



**FIGURE 2.6** – a Caractéristique spatiaux de la marche. b Caractéristique temporels de la marche.

Si le premier événement détecté correspond à une CI droite, utiliser l'équation 2.10. Dans le cas contraire, utiliser l'équation 2.11 . D correspond à un événement de droite, G correspond à un événement gauche et l'indice  $i$  un itérateur du nombre maximal de cycles identifiés.

$$cycle_i = \{CI_{D_i}, CF_{G_i}, CI_{G_i}, CF_{D_i}, CI_{D_{i+1}}, CF_{G_{i+1}}, CI_{G_{i+1}}, CF_{D_{i+1}}\} \quad (2.10)$$

$$cycle_i = \{CI_{G_i}, CF_{D_i}, CI_{D_i}, CF_{G_i}, CI_{G_{i+1}}, CF_{D_{i+1}}, CI_{D_{i+1}}, CF_{G_{i+1}}\} \quad (2.11)$$

Une fois qu'au moins 1 cycle de marche a été obtenu, les équations des caractéristiques spatio-temporelles peuvent être tirées de la figure 2.6. Ces dernières sont comme suit :

**Cadence** : calculée par l'équation 2.12, est le nombre de pas effectués par minute.

$$\begin{aligned} Cadence_D &= \frac{60}{(CI_{D_{i+1}} - CI_{G_i})} \\ Cadence_G &= \frac{60}{(CI_{G_{i+1}} - CI_{D_i})} \end{aligned} \quad (2.12)$$

**Temps de la foulée** : calculé à partir de l'équation 2.13, du contact initial d'un pied au contact initial suivant du même pied.

$$\begin{aligned} tempsfoule_D &= (CI_{D_{i+1}} - CI_{D_i}) \\ temps - foulée_G &= (CI_{G_{i+1}} - CI_{G_i}) \end{aligned} \quad (2.13)$$

**Temps de pas** : donné par l'équation 2.14, est le temps entre le contact initial d'un pied au contact initial du pied opposé.

$$\begin{aligned} tempspas_D &= (CI_{D_{i+1}} - CI_{G_i}) \\ tempspas_G &= (CI_{G_{i+1}} - CI_{D_i}) \end{aligned} \quad (2.14)$$

**Temps du simple appui** : calculé à partir de l'équation 2.15,

$$\begin{aligned} tempssimpleappui_D &= (CI_{G_i} - CF_{G_i}) \\ tempssimpleappui_G &= (CI_{D_{i+1}} - CF_{D_i}) \end{aligned} \quad (2.15)$$

**Temps du double appui** : Le temps du double appui, donné par l'équation 2.16, est le temps durant lequel les deux pieds sont en contact avec le sol. Cela se produit deux fois au cours de chaque cycle de marche : une fois lorsque le pied arrière décolle et une autre fois lorsque le pied avant atterrit. Une durée plus longue peut indiquer une marche plus prudente ou instable.

$$\begin{aligned} tempsdoubleappui_D &= (CF_{G_i} - CI_{D_i}) \\ tempsdoubleappui_G &= (CF_{D_i} - CI_{G_i}) \end{aligned} \quad (2.16)$$

**Temps de la phase d'appui** : Le temps de la phase d'appui, calculé par l'équation, est la durée durant laquelle un pied est en contact avec le sol, exprimée en pourcentage du temps total de la foulée. Une proportion plus élevée peut indiquer une marche plus prudente, tandis qu'une proportion plus faible peut indiquer une marche plus rapide ou

une course.

$$\begin{aligned} \text{tempsappui}_D &= \frac{100(CF_{D_i} - CI_{D_i})}{CI_{D_{i+1}} - CI_{D_i}} \\ \text{tempsappui}_G &= \frac{100(CF_{G_{i+1}} - CI_{G_i})}{CI_{G_{i+1}} - CI_{G_i}} \end{aligned} \quad (2.17)$$

**Temps de la phase d'oscillation :** Calculé par l'équation 2.18, exprimé en pourcentage du temps total de la foulée. C'est le complément du temps de la phase d'appui. Un pourcentage plus élevé indique une marche plus rapide ou une course, tandis qu'un pourcentage plus faible indique une marche plus lente ou une marche prudente.

$$\begin{aligned} \text{tempsoscillation}_D &= \frac{100(CI_{D_{i+1}} - CF_{D_i})}{CI_{D_{i+1}} - CI_{D_i}} \\ \text{tempsoscillation}_G &= \frac{100(CI_{G_i} - CF_{G_i})}{CI_{G_{i+1}} - CI_{G_i}} \end{aligned} \quad (2.18)$$

**Symétrie :** La symétrie, donnée par l'équation 2.19, est une mesure de la différence entre les temps d'appui du pied droit et du pied gauche, normalisée par leur moyenne. Une valeur proche de zéro indique une marche symétrique, tandis qu'une valeur élevée (positive ou négative) indique une asymétrie significative.

$$\text{symtrie} = \frac{\text{tempsappui}_{D_i} - \text{tempsappui}_{D_i}}{\frac{\text{tempsappui}_{D_i} + \text{tempsappui}_{G_i}}{2}} \quad (2.19)$$

### 2.3.1.1 Estimation de la longueur de pas :

Il existe différentes méthodes pour estimer la longueur d'un pas, l'une d'entre elles étant la méthode proposée par Wiebren Zijlstra [58]. Cette méthode repose sur une prédiction confirmée par des données expérimentales. En utilisant cette approche, il est possible de prédire la longueur d'un pas en connaissant la longueur de la jambe ( $l$ ) et en calculant les variations de hauteur ( $h$ ) à partir des données d'accélération en utilisant la formule suivante :

$$L = 2\sqrt{(2lh - h^2)} \quad (2.20)$$

Cependant, notre base de données actuelle ne contient malheureusement pas d'informations sur la longueur de jambe ( $l$ ). Dans cette situation, il est nécessaire de recourir à d'autres approches pour estimer la longueur de pas.

Il existe différentes méthodes statistiques qui peuvent être utilisées à cet effet. Le tableau récapitule quelques-unes de ces méthodes couramment utilisées dans la littérature. En choisissant l'une de ces méthodes statistiques, nous pourrions estimer la longueur de



pas de manière fiable, même en absence d'informations sur la longueur de jambe. Le tableau 2.1 résume les techniques d'estimation les plus utilisées :

Modèle	paramètres	Équation
Mikov et al. [59]	la fréquence de pas (F), valeurs d'accélération verticale maximale et minimale d'un pas, la constante accordable (K)	$\frac{K}{F} \cdot \sqrt{a_{\max} - a_{\min}}$
Bylemans et al. [60]	fréquence des pas (F), accélération verticale de pas, les valeurs minimales et maximales de l'accélération verticale de pas, constante (K)	$0.1 \cdot \sqrt[27]{a_{\text{moyenne}}} \cdot \sqrt{K \sqrt{\frac{F}{a_{\max} - a_{\min}}}}$
Shin and Park [53]	les constantes (K1, K2 et K3), fréquence des pas (F), accélération verticale de pas	$K_1 \cdot F + K_2 \cdot a_v + K_3$
Sharp and Yu [61]	les constantes (K1, K2, K3, K4), fréquence des pas (F), Les valeurs maximales et minimales de l'accélération verticale de pas, hauteur (h) de sujet	$K_1 \cdot h^{K_2} \cdot (a_{\max} - a_{\min})^{K_3} \cdot F^{K_4}$
NH Ho [58]	fréquence des pas (F), accélération verticale de pas, les valeurs minimales et maximales de l'accélération verticale de pas, constante K	$L_{\text{pas}} = K \times \sqrt[4]{A_{\max} - A_{\min}}$

TABLEAU 2.1 – Méthode d'estimation de longueur de pas.

Pour notre étude, nous avons décidé d'adopter la méthode de NH Ho [58] comme approche pour estimer la longueur de pas.

Le facteur K est un paramètre qui diffère d'une personne à une autre. La valeur du facteur K est influencée par des caractéristiques individuelles telles que la taille, la morphologie, l'âge, le sexe, la condition physique, etc.

Afin de tenir compte de cette variabilité, la méthode de Ho introduit une nouvelle approche pour estimer le facteur K.

La formule utilisée pour estimer le facteur K dans la méthode de NH Ho [58] est la suivante :

$$K = 0.68 - 0.37 \times \bar{v}_{\text{pas}} + 0.15 \times \bar{v}_{\text{pas}}^2 \quad (2.21)$$

tel que :

$$\bar{v}_{\text{pas}} = \sqrt{\bar{v}_{\text{pas}X}^2 + \bar{v}_{\text{pas}Y}^2 + \bar{v}_{\text{pas}Z}^2} \quad (2.22)$$

et :

$$\begin{aligned} \bar{v}_{\text{pas}X} &= \text{moyenne} \left( \int a_{r\text{-pas}X}(t) dt \right) \\ \bar{v}_{\text{pas}Y} &= \text{moyenne} \left( \int a_{r\text{-pas}Y}(t) dt \right) \\ \bar{v}_{\text{pas}Z} &= \text{moyenne} \left( \int a_{r\text{-pas}Z}(t) dt \right) \end{aligned} \quad (2.23)$$

La formule suivante est utilisée pour estimer la longueur de pas :

$$L_{\text{pas}} = K \times \sqrt[4]{A_{\max} - A_{\min}} \quad (2.24)$$

### 2.3.2 Caractéristiques statistiques

L'analyse des caractéristiques statistiques joue un rôle crucial dans l'évaluation précise de la marche. Ces paramètres permettent de caractériser plus précisément les différents aspects de la marche d'une personne. L'utilisation conjointe de ces mesures peut aider à identifier des tendances ou des modèles dans la marche, à détecter des irrégularités ou des variations.

Il est important de souligner que la magnitude de l'accélération,  $a[i]$ , telle que définie par l'équation 2.25, sert de base pour les calculs dans les équations ultérieures.

$$a[i] = \sqrt{a_x[i]^2 + a_y[i]^2 + a_z[i]^2} \quad (2.25)$$

**Moyenne :** La moyenne,  $\mu$ , donnée par l'équation 2.26, est utilisée pour obtenir une valeur centrale de l'accélération pour chaque cycle de marche. Cela nous permet d'évaluer le niveau d'accélération générale à chaque pas, ce qui pourrait être utile pour détecter des irrégularités ou des variations dans la marche.

$$\mu = \frac{1}{CI_{D_{i+1}} - CI_{D_i}} \sum_{i=CI_{D_i}}^{CI_{D_{i+1}}-1} a_i \quad (2.26)$$

**Puissance :** La puissance, donnée par l'équation 2.27, est une mesure de l'énergie globale du signal d'accélération pour chaque cycle de marche. Elle peut être utilisée pour identifier l'intensité du mouvement. Par exemple, une personne qui marche avec plus de force (peut-être en raison d'une boiterie) pourrait présenter une puissance d'accélération plus élevée.

$$Puissance = \sum_{i=CI_{D_i}}^{CI_{D_{i+1}}} a_i^2 \quad (2.27)$$

**Variance :** La variance, calculée à partir de l'équation 2.28, est utilisée pour évaluer la dispersion du signal d'accélération autour de sa moyenne pour chaque cycle de marche. Une variance élevée peut indiquer un pas irrégulier, par exemple, un pas plus long suivi d'un pas plus court.

$$Variance = \frac{1}{CI_{D_{i+1}} - CI_{D_i}} \sum_{i=CI_{D_i}}^{CI_{D_{i+1}}-1} (a_i - \mu)^2 \quad (2.28)$$

**Écart type :** L'écart type, calculé à partir de l'équation 2.29, est une mesure de la dispersion des valeurs d'accélération autour de la moyenne pour chaque cycle de marche.

L'écart type peut être utile pour comprendre à quel point les valeurs d'accélération varient par rapport à la moyenne.

$$\sigma = \sqrt{\frac{1}{CI_{D_{i+1}} - CI_{D_i}} \sum_{i=CI_{D_i}}^{CI_{D_{i+1}}-1} (a_i - \mu)^2} \quad (2.29)$$

**Skewness** : L'asymétrie, ou skewness, est calculée selon l'équation 2.30. Elle peut révéler des asymétries dans la distribution de l'accélération pour chaque cycle de marche.

$$Skewness = \frac{\frac{1}{CI_{D_{i+1}} - CI_{D_i}} \sum_{i=CI_{D_i}}^{CI_{D_{i+1}}-1} (a_i - \mu)^3}{\left( \frac{1}{CI_{D_{i+1}} - CI_{D_i}} \sum_{i=CI_{D_i}}^{CI_{D_{i+1}}-1} (a_i - \mu)^2 \right)^{\frac{3}{2}}} \quad (2.30)$$

**Max** : La valeur maximale, calculée par l'équation 2.31, indique l'accélération maximale atteinte lors de chaque cycle de marche. Elle peut aider à identifier les pics d'accélération, ce qui peut être lié à des caractéristiques spécifiques de la marche, comme une marche claudicante.

$$Max = \max(a_{CI_{D_i}}, \dots, a_{CI_{D_{i+1}}}) \quad (2.31)$$

**Min** : La valeur minimale, calculée par l'équation 2.32, représente l'accélération minimale atteinte lors de chaque cycle de marche. Cela peut aider à identifier les moments de faible accélération, comme lorsque le pied est au repos.

$$Min = \min(a_{CI_{D_i}}, \dots, a_{CI_{D_{i+1}}}) \quad (2.32)$$

En combinant les informations obtenues à partir des caractéristiques spatio-temporelles et statistiques, nous sommes en mesure d'extraire une image complète et précise du comportement de marche, ce qui est l'objectif principal de notre étude.

## 2.4 Apprentissage automatique

L'apprentissage automatique, également connu sous le nom de machine learning, est un domaine de l'intelligence artificielle qui permet aux ordinateurs d'apprendre à partir de données sans être explicitement programmés. Il s'agit d'une approche puissante pour résoudre des problèmes complexes de classification, de régression et de clustering, en exploitant les structures et les motifs cachés dans les données.

Dans cette section, nous allons explorer plusieurs algorithmes populaires d'apprentissage automatique utilisés dans notre étude de classification de chutes. Nous examinerons

en détail les algorithmes SVM, KNN, DT, les algorithmes ensemblistes tels que le Random Forest et le boosting, ainsi que les réseaux de neurones. En outre, nous introduirons le concept novateur des réseaux de neurones convolutifs 1D (CNN 1D), qui sont spécifiquement conçus pour traiter des données séquentielles telles que les séries temporelles.

### 2.4.1 K-plus proches voisins (K-Nearest Neighbor)

Les K-voisins les plus proches sont l'un des algorithmes de classification les plus élémentaires, mais aussi les plus essentiels, de l'apprentissage automatique. Il s'agit d'un modèle d'apprentissage supervisé couramment utilisé pour réaliser les tâches de classification. Il est largement utilisé dans les scénarios de la vie réelle, car il est non paramétrique, ce qui signifie qu'il ne fait aucune hypothèse sous-jacente sur la distribution des données.

Pour faire une prédiction, l'algorithme K-NN n'a pas besoin de construire un modèle prédictif à partir d'un ensemble d'apprentissage, c'est pourquoi K-NN est parfois catégorisé comme "apprentissage paresseux". Au lieu de cela, K-NN s'appuie directement sur l'ensemble de données lui-même pour générer un résultat prédictif. En d'autres termes, il n'y a pas de phase d'entraînement explicite ou très minimale dans le K-NN. Cela signifie que la phase de formation est assez rapide. La méthode KNN suppose que les données sont situées dans un espace de caractéristiques. Cela signifie que les points de données existent dans un espace métrique. Les données peuvent être des scalaires ou même des vecteurs multidimensionnels.

Cette méthode utilise principalement deux paramètres : une fonction de similarité pour comparer les individus dans l'espace des caractéristiques, et la valeur de k qui détermine le nombre de voisins qui influencent la classification.

Différentes mesures de distance peuvent être utilisées dans l'algorithme KNN.

**La distance euclidienne :** la distance qui calcule la racine carrée de la somme des différences carrées entre les coordonnées de deux points :

$$D_e(x, y) = \sqrt{\sum_{j=1}^n (x_j - y_j)^2}$$

**Distance Manhattan :** La distance de Manhattan : calcule la somme des valeurs absolues des différences entre les coordonnées de deux points :

$$D_m(x, y) = \sum_{i=1}^k |x_i - y_i|$$

**Distance Hamming :** la distance entre deux points données est la différence maximale

entre leurs coordonnées sur une dimension.

$$D_h(x, y) = \sum_{i=1}^k |x_i - y_i|$$

En résumé, l'algorithme KNN calcule les distances entre l'instance de test et tous les points de données, sélectionne les  $k$  voisins les plus proches et attribue l'étiquette majoritaire parmi ces voisins comme étiquette prédite pour l'instance de test.

### 2.4.2 Machine à vecteurs de support SVM :

La machine à vecteurs de support (SVM) est l'un des classificateurs d'apprentissage supervisé les plus exploités dans le domaine de l'analyse de la marche. Ils sont couramment utilisés pour la classification binaire, où l'objectif est de séparer les points de données en deux classes. L'algorithme apprend une frontière de décision (hyperplan) qui maximise la marge entre les classes, afin d'obtenir la meilleure séparation possible [62].

Les paramètres utilisés dans les SVM peuvent affecter de manière significative leurs performances et leur capacité de généralisation. Voici quelques paramètres importants à prendre en compte :

- **Marge maximale** : Le SVM cherche à trouver l'hyperplan qui a la marge maximale, c'est-à-dire la distance entre l'hyperplan et les points de données les plus proches de chaque classe. Cette marge permet d'améliorer la capacité de généralisation du modèle et de réduire le risque de sur-ajustement.
- **Vecteurs de support** : Les vecteurs de support sont les points de données les plus proches de la limite de décision. Ces points jouent un rôle crucial dans la définition de l'hyperplan et la détermination de la frontière de décision.
- **Fonctions de noyau** : La fonction noyau est un outil mathématique qui permet d'effectuer une classification bidimensionnelle d'un ensemble de données. Cette fonction projette les données d'un espace de faible dimension vers un espace de dimension supérieure. Le bon choix de cette fonction permet de séparer les données dans l'espace dimensionnel supérieur résultant. Les fonctions de noyau les plus courantes sont les fonctions linéaires, polynomiales, à base radiale (RBF) et sigmoïdes. Le choix du noyau dépend des données et du problème à résoudre.
- **Paramètre de régularisation** : Les SVM comprennent un paramètre de régularisation, souvent désigné par  $C$ . Il contrôle le compromis entre l'obtention d'une marge plus importante et l'autorisation d'erreurs de classification. Une valeur  $C$  plus petite autorise davantage de classifications erronées, mais peut conduire à une marge plus importante, tandis qu'une valeur  $C$  plus grande vise une marge plus

petite, mais tolère moins de classifications erronées.

- **Influence de Gamma** : Le paramètre gamma contrôle l'influence d'un seul exemple d'entraînement. Une petite valeur de gamma signifie que la limite de décision est davantage influencée par les points de données éloignés, ce qui donne une limite de décision plus lisse. Inversement, une valeur gamma élevée donne plus de poids aux points de données proches, ce qui se traduit par une frontière de décision plus complexe. Lorsqu'on utilise un noyau non linéaire tel que la fonction de base radiale (RBF), la limite de décision est déterminée par une combinaison de vecteurs de support et de leurs poids correspondants. Le paramètre gamma influence la forme de la limite de décision dans l'espace des caractéristiques transformé. La figure 2.7 illustre le schéma explicatif du SVM (Support Vector Machine).

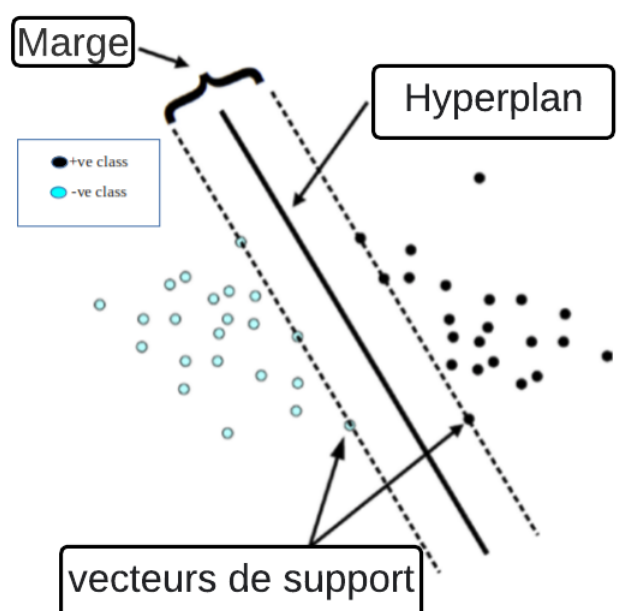


FIGURE 2.7 – L'algorithme SVM : Hyperplan à marge maximale et vecteurs de support.

### 2.4.3 Arbre de décision (Decision trees) :

Un arbre de décision est un modèle non paramétrique d'apprentissage automatique largement utilisé pour la classification et la régression. Il est caractérisé par sa structure en forme d'arbre, où chaque nœud représente une règle de décision basée sur les caractéristiques des données d'entrée. Cette structure de modèle comprend des nœuds et des branches qui sont désignés selon la terminologie spécifique à l'arbre de décision [63].

**La structure de modèle :**

- **Nœud racine** : Le nœud racine est le point de départ de l'arbre décisionnel. Il représente l'ensemble des données, qui sont ensuite divisées en deux ou plusieurs ensembles homogènes.
- **Nœud feuille** : Les nœuds feuilles constituent le nœud de sortie final et l'arbre ne peut plus être divisé après l'obtention d'un nœud feuille.
- **Fractionnement** : Le fractionnement consiste à diviser le nœud de décision/nœud racine en sous-nœuds en fonction des conditions données.
- **Branche/sous-arbre** : Un arbre formé par la division de l'arbre.
- **Élagage** : L'élagage consiste à supprimer les branches indésirables de l'arbre.
- **Nœud parent/enfant** : Le nœud racine de l'arbre est appelé nœud parent, et les autres nœuds sont appelés nœuds enfants.

L'algorithme de l'arbre de décision commence par évaluer l'importance des différentes caractéristiques en fonction de leur capacité à diviser les données et à faire des prédictions précises. Il sélectionne la meilleure caractéristique comme nœud racine de l'arbre et divise les données en sous-ensembles en fonction des valeurs de cette caractéristique. Ce processus est répété de manière récursive pour chaque sous-ensemble, créant ainsi d'autres divisions et branches dans l'arbre jusqu'à ce qu'un critère d'arrêt soit atteint, comme atteindre une profondeur maximale ou un nombre minimum d'échantillons dans un nœud feuille [64].

Chaque nœud interne représente une caractéristique, chaque branche représente une règle de décision et chaque nœud feuille représente une étiquette de classe ou une valeur prédite. Dans le cas de classification, l'arbre de décision prédit l'étiquette de classe d'une nouvelle instance en traversant l'arbre en fonction de ses valeurs de caractéristiques. Dans les tâches de régression, les nœuds feuilles contiennent des valeurs numériques prédites.

Les arbres de décision ont plusieurs avantages, notamment leur interprétabilité, car les règles de décision apprises peuvent être facilement visualisées et comprises. Ils peuvent gérer à la fois des caractéristiques catégorielles et numériques, et ils sont robustes aux valeurs aberrantes et aux données manquantes. Cependant, les arbres de décision peuvent être sujets à un surapprentissage s'ils ne sont pas correctement régularisés ou si l'arbre devient trop complexe.

Des techniques telles que l'élagage, la limitation de la profondeur de l'arbre ou l'utilisation de méthodes d'ensemble telles que les forêts aléatoires ou le boosting peuvent aider à atténuer le surapprentissage et à améliorer les performances de généralisation.

Les arbres de décision sont des algorithmes polyvalents et puissants qui peuvent être utilisés pour diverses tâches d'apprentissage automatique, offrant à la fois une interprétabilité et des capacités prédictives.

Examinons la figure ci-dessous 2.8 qui illustre l'algorithme d'un arbre de décision.

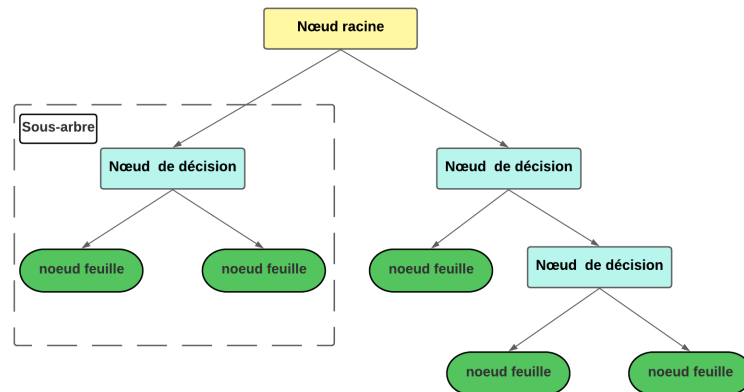


FIGURE 2.8 – Principe de l’algorithme de l’arbre de décision.

#### 2.4.4 Apprentissage ensembliste :

L’apprentissage ensembliste [65] est une approche de l’apprentissage automatique qui vise à améliorer les performances prédictives en combinant les prédictions de plusieurs modèles. Trois méthodes dominent le domaine de l’apprentissage ensembliste, chacune est un domaine d’étude qui a donné naissance à de nombreuses méthodes spécialisées. Les trois principales classes de méthodes d’apprentissage ensembliste sont le **bagging**, le **stacking** et le **boosting** [66] .

1. Le **bagging** consiste à ajuster de nombreux arbres de décision sur différents échantillons du même ensemble de données et à faire la moyenne des prédictions.

Le schéma 2.9 illustre le principe de l’approche bagging.



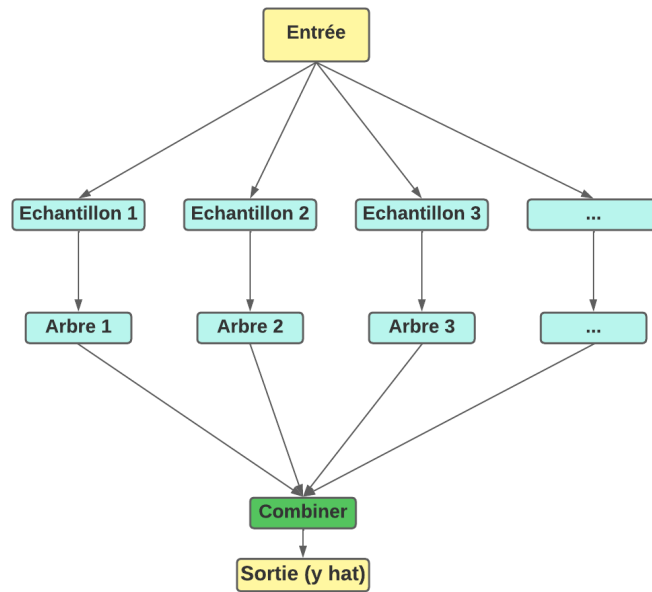


FIGURE 2.9 – Principe du Bagging : Combinaison d’estimateurs par échantillonnage avec remplacement.

2. **Le stacking** consiste à ajuster plusieurs types de modèles différents sur les mêmes données et à utiliser un autre modèle pour apprendre comment combiner au mieux les prédictions. La figure 2.10 représente un schéma explicatif de l’approche stacking.

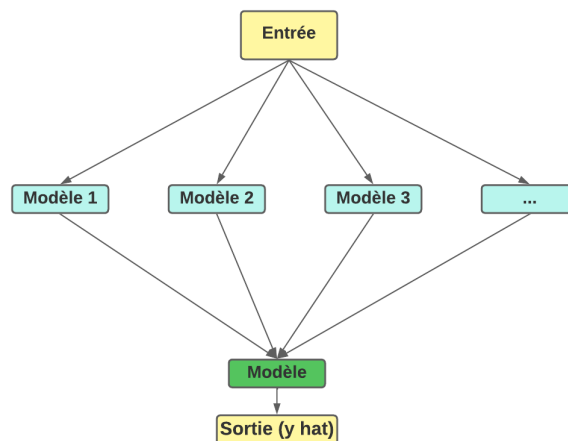


FIGURE 2.10 – Principe du Stacking : Combinaison de modèles par empilement.

3. **Le boosting** consiste à ajouter séquentiellement des membres de l’ensemble qui corrigent les prédictions faites par les modèles précédents et à produire une moyenne pondérée des prédictions.

La figure 2.11 ci-dessous explique le principe de boosting.

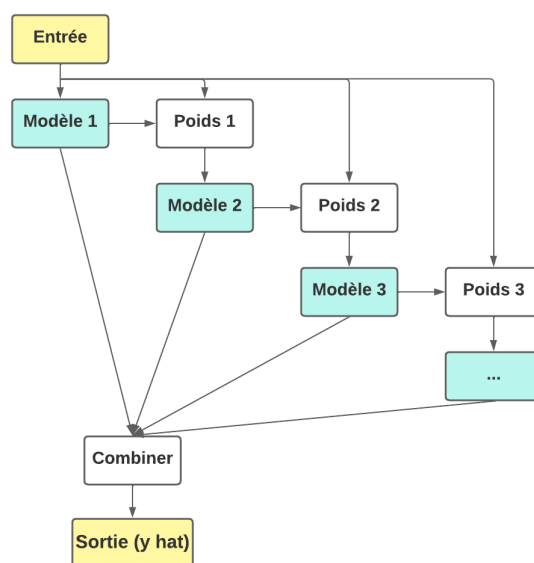


FIGURE 2.11 – Principe du Boosting : Renforcement de modèles par apprentissage itératif.

### 2.4.5 Forêt aléatoire (Random Forest) :

La forêt aléatoire (Random Forest) est un algorithme d'apprentissage qui combine plusieurs arbres de décision pour effectuer des prédictions. Il appartient à la famille des méthodes d'ensemble, où plusieurs modèles sont combinés pour améliorer les performances prédictives [67].

L'algorithme de la forêt aléatoire construit un ensemble d'arbres de décision indépendants les uns des autres. Chaque arbre est formé sur un sous-ensemble aléatoire des données d'entraînement, et chaque arbre effectue des prédictions individuelles. Lors de la prédiction, la forêt aléatoire agrège les prédictions de tous les arbres pour obtenir une prédiction finale. Cela permet d'obtenir une réponse plus robuste et plus précise, en réduisant les effets du surapprentissage.

Il est alors nécessaire de définir des critères d'arrêt pour arrêter la croissance d'un arbre avant qu'il n'atteigne un nombre trop élevé de niveaux pour éviter le surapprentissage,

Parmi ces paramètres, les plus courants :

1. Le nombre minimum d'échantillons dans un nœud terminal pour permettre la division.
2. Le nombre minimum d'échantillons dans un nœud feuille lorsque le nœud terminal est divisé.

- La profondeur maximale de l'arbre : c'est le nombre maximal de niveaux qu'un arbre peut atteindre.

L'un des avantages clés de la forêt aléatoire est sa capacité à gérer des ensembles de données de grande taille et complexes avec un nombre élevé de caractéristiques. La forêt aléatoire est également moins sensible aux valeurs aberrantes et au bruit dans les données, ce qui en fait un choix populaire dans de nombreux domaines.

La forêt aléatoire est un algorithme puissant d'apprentissage automatique qui combine plusieurs arbres de décision pour obtenir des prédictions précises et robustes. Elle est largement utilisée dans de nombreux domaines, notamment la classification, la régression et la détection d'anomalies.

La figure suivante 2.12 offre une représentation schématique claire de l'architecture des forêts aléatoires.

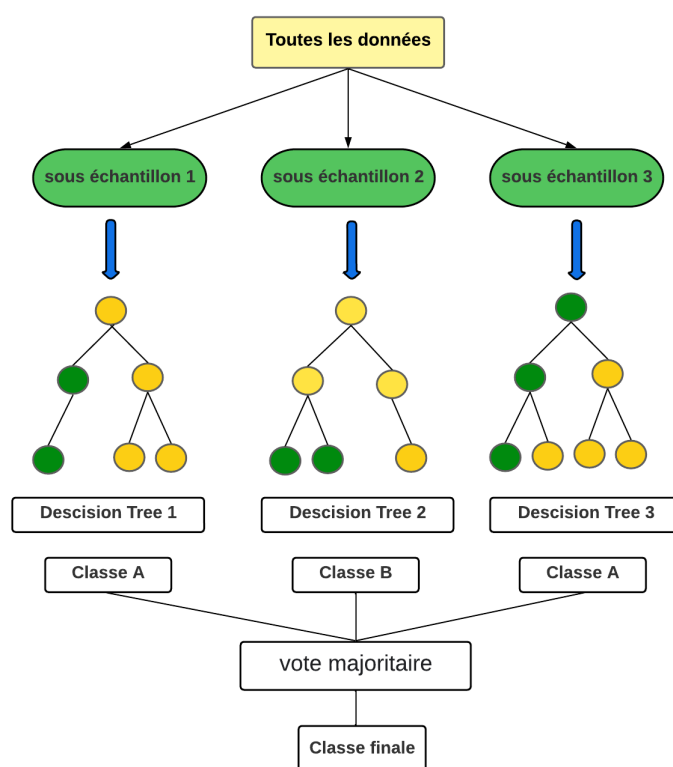


FIGURE 2.12 – Principe des Forêts Aléatoires : Agrégation de Modèles d'Arbres de Décision.

## 2.4.6 Réseaux de neurones :

Les réseaux de neurones, également connus sous le nom de réseaux neuronaux artificiels (RNA), font partie de l'apprentissage automatique et sont le cœur des algorithmes de deep

learning. Leur nom et leur structure sont inspirés du cerveau humain, imitant la manière dont les neurones biologiques se signalent mutuellement.

Les réseaux neuronaux artificiels (RNA) [68] sont composés de couches de nœuds, comprenant une couche d'entrée, une ou plusieurs couches cachées et une couche de sortie. Chaque nœud, ou neurone artificiel, est connecté à un autre et possède un poids associé. Ces poids aident à déterminer l'importance de chaque neurone, les neurones avec des poids plus importants contribuant de manière plus significative à la sortie par rapport aux autres nœuds. L'entraînement d'un réseau de neurones implique de sélectionner les poids les plus optimaux parmi l'ensemble des neurones qui composent le réseau.

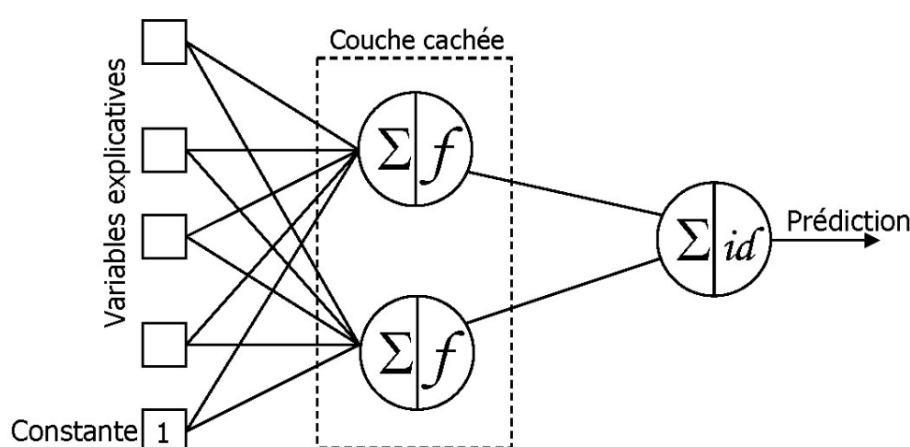


FIGURE 2.13 – Principe des Réseaux de Neurones : Modélisation d'un simple réseaux de neurones.

L'objectif est de modifier les poids  $w$  de telle façon que pour chaque entrée, le réseau prédit le plus précisément possible la valeur  $y$  associée. Cela consiste à minimiser la somme des carrés des erreurs  $E(w)$  appelée fonction de coût, qui peut être exprimé mathématiquement comme suit :

$$\min_w E(w) = \sum_{(x,y)} (f_w(x) - y)^2 \quad \begin{array}{l} f_w(x) - \text{la valeur prédite} \\ y - \text{la valeur réelle} \end{array} \quad (2.33)$$

À mesure que le modèle ajuste ses poids et son biais, il utilise la fonction de coût pour atteindre le point de convergence, ou le minimum local. Le processus par lequel l'algorithme ajuste ses poids est appelé descente de gradient, permettant au modèle de déterminer la direction à prendre pour réduire les erreurs (ou minimiser la fonction de coût).

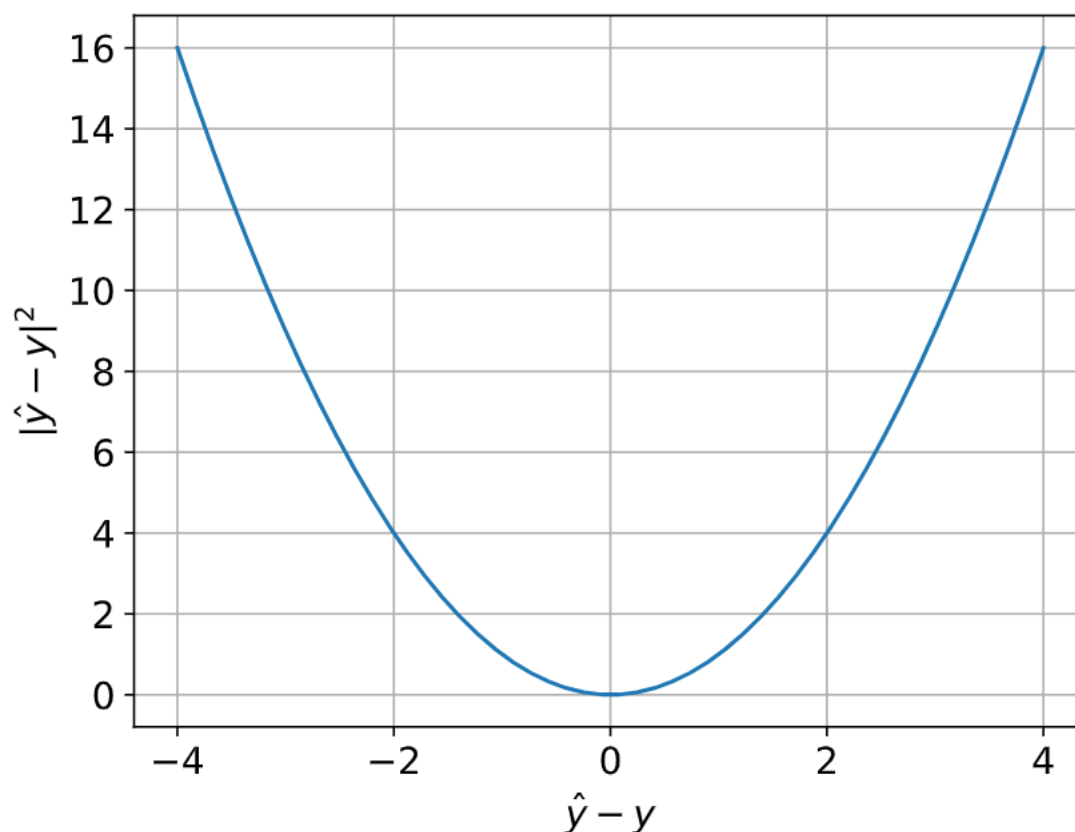


FIGURE 2.14 – Graphe de la Fonction de Coût.

### L'apprentissage profond (Deep Learning) :

Le terme "profond" dans l'appellation apprentissage profond fait référence à la profondeur des couches d'un réseau de neurones. Un réseau de neurones composé de plus de trois couches, incluant les couches d'entrée et de sortie, peut être considéré comme un algorithme d'apprentissage profond. Un réseau de neurones qui n'a que deux ou trois couches est simplement un réseau de neurones basique.

## 2.5 Deep learning

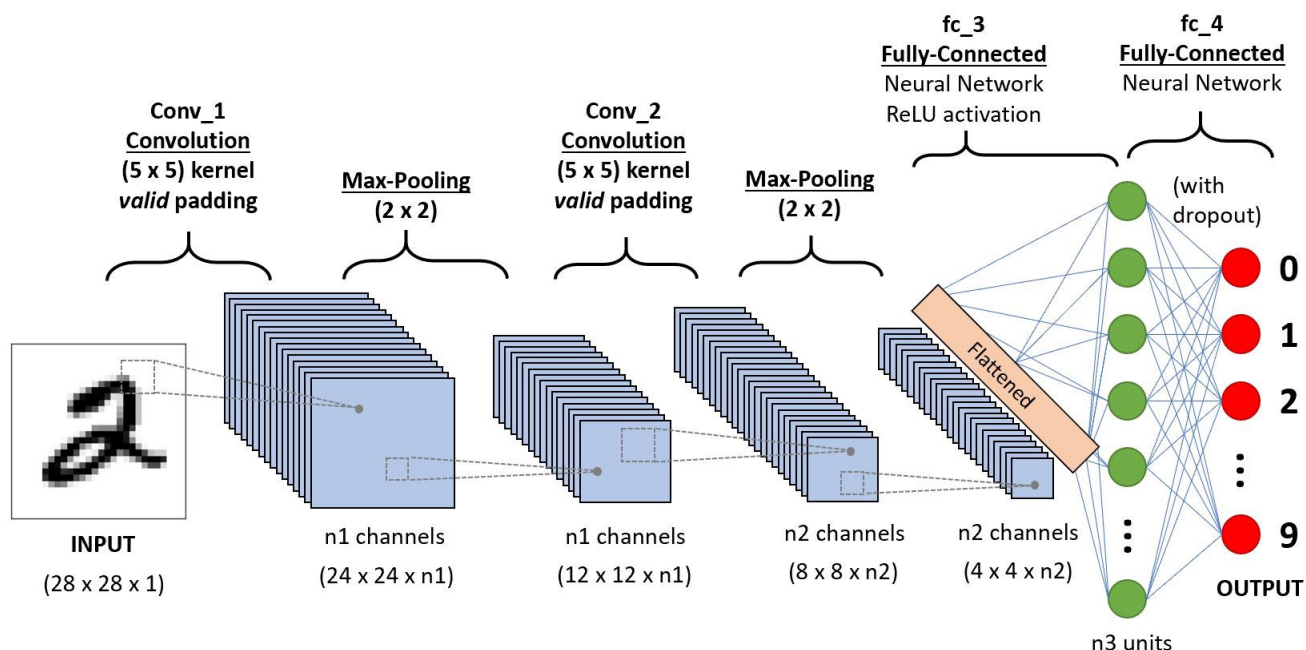
Le Deep Learning est un nouveau domaine de recherche du ML, qui a été introduit dans le but de rapprocher l'apprentissage automatique de son objectif principal : l'intelligence artificielle. Il concerne les algorithmes inspirés par la structure et le fonctionnement du cerveau. Ils peuvent apprendre plusieurs niveaux de représentation dans le but de modéliser des relations complexes entre les données.

### 2.5.1 Réseau de Neurones Convolutif (CNN) :

CNN est l'acronyme de "Convolutional Neural Network", qui se traduit en français par "Réseau de Neurones Convolutif". Un CNN est un type spécifique de réseau de neurones artificiels, conçu pour traiter principalement des données structurées en grille, telles que des images.

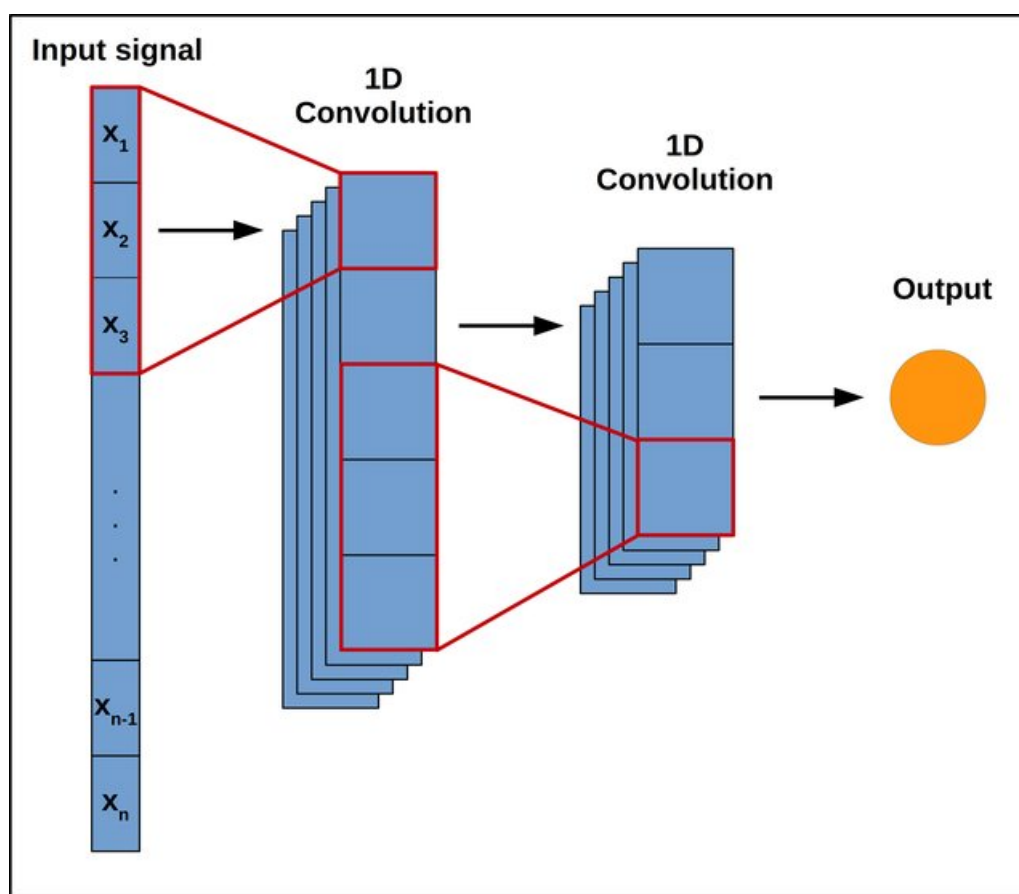
Un CNN est composé de plusieurs couches, dont les couches de convolution, de pooling et de classification. Les couches de convolution sont responsables de l'extraction des caractéristiques de l'image en appliquant des filtres convolutifs pour détecter des motifs visuels tels que des bords, des textures ou des formes. Les couches de pooling réduisent la dimensionnalité des caractéristiques extraites, ce qui permet de réduire les calculs tout en conservant les informations importantes. Enfin, les couches de classification utilisent les caractéristiques extraites pour effectuer la classification de l'image dans différentes catégories prédéfinies.

CNN 1D, ou "Convolutional Neural Network 1D", est une variante spécifique du CNN conçue pour traiter des données unidimensionnelles, telles que des séquences temporelles ou des signaux. Contrairement au CNN traditionnel qui traite des données en grille (2D), le CNN 1D applique des opérations de convolution sur une dimension. Il est souvent utilisé pour des tâches telles que la classification de séquences temporelles, la reconnaissance de gestes, l'analyse de signaux audio, etc. Le schéma du CNN2D suivant [2.15](#) illustre de manière visuelle les différentes étapes clés de ce réseau de neurones convolutif spécialement conçu pour l'analyse et le traitement d'images.



**FIGURE 2.15** – Réseau de Neurones Convolutif 2D (CNN) : Exemple d’xtraction de caractéristiques à partir d’images en utilisant des filtres convolutionnels.

L’architecture d’un CNN 1D est similaire à celle d’un CNN traditionnel, mais adaptée pour une dimension. Il comprend généralement des couches de convolution 1D, des couches de pooling 1D et des couches de classification. Les opérations de convolution 1D permettent d’extraire les motifs et les structures temporelles importantes des données, tandis que les couches de pooling 1D réduisent la dimensionnalité de manière similaire à un CNN traditionnel. La figure ci-dessous 2.16 présente de manière graphique les différentes étapes fondamentales de ce réseau de neurones convolutif unidimensionnel



**FIGURE 2.16** – Réseau de Neurones Convolutif 1D (CNN) : Extraction de caractéristiques à partir de séquences temporelles à l’aide de filtres convolutifs.

Le CNN 1D est une extension du CNN traditionnel adaptée au traitement de données unidimensionnelles. Il permet d’extraire des caractéristiques pertinentes et d’effectuer des tâches d’analyse et de classification sur des séquences temporelles ou des signaux.

## 2.6 Réduction de dimension :

En apprentissage automatique, le nombre d’attributs, de caractéristiques ou de variables d’entrée d’un ensemble de données est appelé dimensionnalité. La réduction de la dimensionnalité est une étape de pré-traitement qui se réfère au processus de réduction du nombre d’attributs dans un ensemble de données tout en conservant autant que possible la variation dans l’ensemble de données original [69]. Lorsque nous réduisons la dimensionnalité, nous perdons un certain pourcentage d’informations des données d’origine, généralement de 1 % à 15 %, en fonction du nombre de composantes ou de caractéristiques que nous conservons [70].

**Avantages :**



- Un nombre réduit de dimensions signifie moins de temps d'apprentissage et moins de ressources informatiques.
- Dans un ensemble de données à haute dimension, la plupart des points de données sont susceptibles d'être très éloignés les uns des autres. Par conséquent, les algorithmes ne peuvent pas s'entraîner de manière efficace et efficiente, Ce type de problème est appelé "malédiction de la dimensionnalité"
- Évite le problème de l'overfitting : Lorsque les données contiennent de nombreuses caractéristiques, les modèles deviennent plus complexes et ont tendance à s'adapter de manière excessive aux données d'apprentissage.
- La réduction de la dimensionnalité s'occupe de la multicollinéarité, celle-ci se produit lorsqu'une variable indépendante est fortement corrélée avec une ou plusieurs autres variables indépendantes.
- La réduction de la dimension supprime le bruit dans les données en ne conservant que les caractéristiques les plus importantes et en supprimant les caractéristiques redondantes.

Il existe principalement deux types de méthodes de réduction de la dimensionnalité. Le premier type de méthode ne conserve que les caractéristiques les plus importantes de l'ensemble de données et supprime les caractéristiques redondantes. Aucune transformation n'est appliquée à l'ensemble des caractéristiques. L'autre méthode consiste à trouver une combinaison de nouvelles caractéristiques, le nouvel ensemble de caractéristiques contient différentes valeurs des valeurs originales. Cette méthode peut être divisée en deux catégories : les méthodes linéaires et les méthodes non linéaires. Les méthodes non linéaires sont bien connues sous le nom de Manifold learning. L'analyse en composantes principales (ACP), l'analyse factorielle (AF) sont des exemples de méthodes linéaires. L'ACP à noyau, l'intégration des voisins stochastiques distribués en t (t-SNE) sont des exemples de méthodes non linéaires.

Pour faciliter la compréhension des différentes méthodes de réduction de dimension, le schéma ci-dessous [2.17](#) qui présente de manière synthétique les approches les plus couramment utilisées.

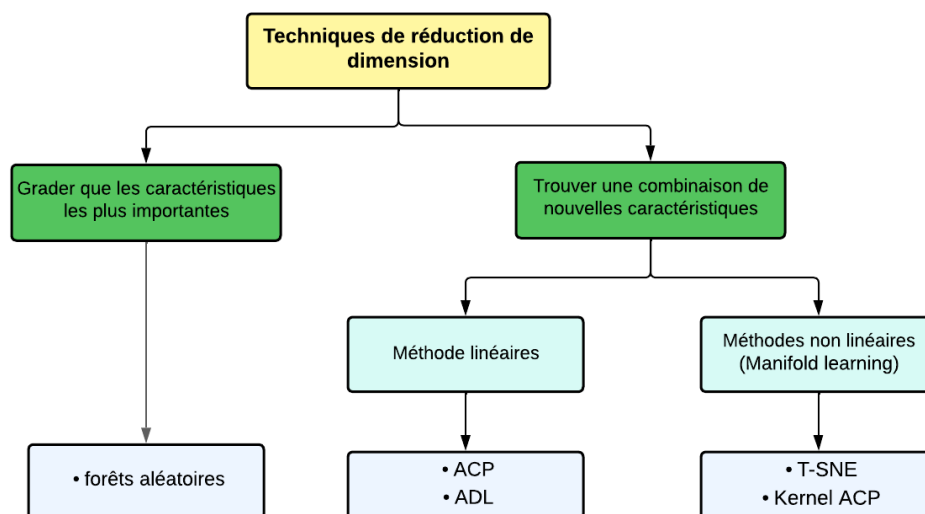


FIGURE 2.17 – Méthodes de réduction de la dimension

— **Analyse en composantes principales (ACP)** : L'ACP est une technique linéaire qui transforme les variables originales en un nouvel ensemble de variables non corrélées appelées composantes principales. Ces composantes principales sont ordonnées par variance décroissante, ce qui permet de conserver l'essentiel de l'information dans un espace de dimension réduit [71].

1. **Standardisation des données** : La première étape consiste à centrer les variables en soustrayant la moyenne et à les mettre à l'échelle en divisant par l'écart type. Cela permet de s'assurer que les variables sont comparables et évite que certaines variables avec une variance plus élevée dominant l'analyse.
2. **Calcul de la matrice de covariance** : Ensuite, on calcule la matrice de covariance, qui mesure les relations linéaires entre les variables. La covariance entre deux variables mesure comment elles varient ensemble.
3. **Calcul des valeurs propres et des vecteurs propres** : À partir de la matrice de covariance, on calcule les valeurs propres et les vecteurs propres. Les valeurs propres représentent l'importance de chaque composante principale et mesurent la quantité de variance expliquée par chaque composante. Les vecteurs propres indiquent la direction de chaque composante principale.
4. **Sélection des composantes principales** : Les composantes principales sont ordonnées en fonction de leurs valeurs propres, de la plus grande à la plus petite. On peut choisir de garder un certain nombre de composantes principales qui captent une quantité significative de variance dans les données. Par exemple,

on peut choisir de conserver les composantes principales qui expliquent 80% de la variance totale.

5. **Projection des données** : Enfin, les données sont projetées sur les composantes principales sélectionnées pour obtenir les nouvelles variables transformées. Chaque observation est représentée par ses coordonnées dans l'espace des composantes principales.

La figure ci-dessous 2.18 présente de manière séquentielle les différentes étapes de cette méthode.

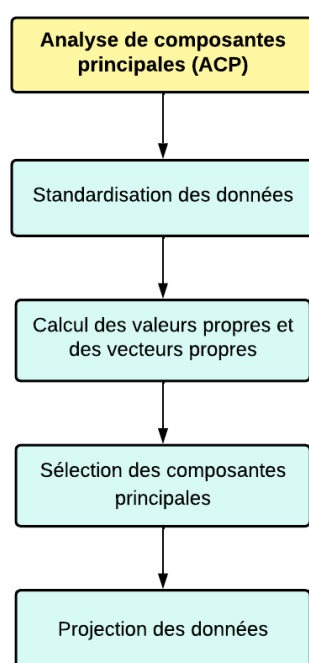


FIGURE 2.18 – Les étapes principales de l'ACP

- **Analyse factorielle discriminante (LDA)** : L'analyse factorielle discriminante est similaire à la PCA, mais elle est utilisée dans le contexte de la classification supervisée. Elle cherche à trouver une projection linéaire qui maximise la séparation entre les classes tout en maintenant la compacité des exemples appartenant à la même classe.
- **T-SNE (t-Distributed Stochastic Neighbor Embedding)** : Le t-SNE est une technique non linéaire de réduction de dimension qui tente de préserver les relations de similarité entre les exemples dans l'espace d'origine. Il est souvent utilisé pour la visualisation de données de haute dimension, car il est capable de capturer des structures complexes dans les données.

- **Les méthodes basées sur les arbres**, telles que les arbres de décision et les forêts aléatoires, sont également utilisées pour la réduction de dimension en sélectionnant les caractéristiques les plus importantes au lieu de transformer les données.

## 2.7 Conclusion

Au cours de ce chapitre, nous avons étudié plusieurs méthodologies pour segmenter et analyser les données de marche. Nous avons d'abord examiné plusieurs algorithmes de segmentation, dont l'algorithme CWT, l'algorithme de Shin, l'algorithme de Gonzales, et l'algorithme de Ghersi. Chacun de ces algorithmes offre une approche unique pour diviser les données de marche en segments exploitables, capturant ainsi les événements clés de la marche.

Ensuite, nous avons étudié l'utilisation de l'apprentissage automatique pour notre analyse, en nous concentrant sur deux aspects clés : la réduction de dimension pour extraire les caractéristiques les plus pertinentes, et l'utilisation de différents algorithmes de classification. Nous avons exploré plusieurs algorithmes, dont le SVM, le KNN, l'arbre de décision (DT), l'algorithme ensembliste de forêts aléatoires (RF), ainsi que les réseaux de neurones profonds (DNN) et les réseaux de neurones à convolution 1D (CNN1D).

# Chapitre 3

## Analyse et resultats

### 3.1 Introduction

Dans ce troisième chapitre de notre travail, nous nous concentrerons sur l'application des méthodologies et des techniques que nous avons présentées dans les chapitres précédents. Nous allons tout d'abord introduire la base de données utilisée pour l'analyse, puis nous examinerons les différentes techniques de segmentation des signaux, en mettant l'accent sur leur efficacité. Ensuite, nous passerons en revue l'ensemble des caractéristiques extraites. Le chapitre se poursuivra en détaillant le paramétrage et l'entraînement des algorithmes d'apprentissage automatique. Nous discuterons des différents algorithmes utilisés, des métriques d'évaluation et des paramètres choisis pour obtenir des modèles performants. Enfin, nous présenterons les résultats obtenus à partir de notre analyse et de nos algorithmes d'apprentissage automatique. Nous examinerons les performances des algorithmes en termes de prédiction des chutes, en mettant en évidence les résultats les plus significatifs et les conclusions que nous pouvons en tirer.

### 3.2 Base de donnée

Dans cette section, nous présentons la base de données utilisée pour notre étude, intitulée "Long Term Movement Monitoring" (LTMM) de Physionet, provenant de l'article [25]. La base de données LTMM a été conçue pour évaluer le risque de chute chez les adultes âgés dans leurs maisons et leurs communautés en surveillant en continu leur activité quotidienne.

La base de données se compose d'informations recueillies auprès de 71 adultes âgés vivant en communauté, dont l'âge moyen est de  $78.36 \pm 4.71$  ans, avec une fourchette de 65 à 87 ans. Les participants ont été classifiés en deux groupes, chuteurs et non-chuteurs, en

fonction de leur auto-déclaration de chutes au cours de l'année précédente. Les individus ayant fait preuve d'au moins 2 chutes ont été identifiées comme chuteurs, tandis que les autres ont été classés comme non-chuteurs.

Ces sujets ont effectué leurs activités quotidiennes habituelles, sans aucune instruction spécifique ni modification de leur routine typique, ce qui a permis d'observer des schémas de marche divers et variés. Néanmoins, malgré l'absence de structure prédéfinie dans les routines de marche, les paramètres dérivés de l'accéléromètre se sont avérés être liés au statut de chute, à la fois rétrospectivement et prospectivement. Cela suggère que les caractéristiques extraites peuvent être utilisées comme des mesures objectives du risque de chute.

En termes de matériel, la base de données contient des signaux d'accéléromètre et de gyroscope, capturés à l'aide d'une ceinture équipée d'un capteur léger (DynaPort,  $87 \times 45 \times 14$  mm, 74 g). Ce capteur, placé sur le bas du dos des sujets, est équipé d'un accéléromètre et d'un gyroscope triaxiaux, permettant ainsi une mesure précise et diversifiée de l'activité de marche.

Bien que la base de données LTMM contienne des données recueillies à la fois dans des environnements de la vie réelle et en laboratoire, nous avons décidé, pour les besoins de notre étude, de n'utiliser que les données générées en laboratoire. Cette décision repose sur le fait que l'évaluation de la marche en laboratoire offre un environnement contrôlé, permettant une standardisation et une reproductibilité des mesures, et une précision accrue des capteurs. De plus, dans ce contexte, les participants étaient libres de marcher à leur vitesse confortable et auto-sélectionnée, offrant ainsi une représentation plus fidèle de leur marche naturelle.

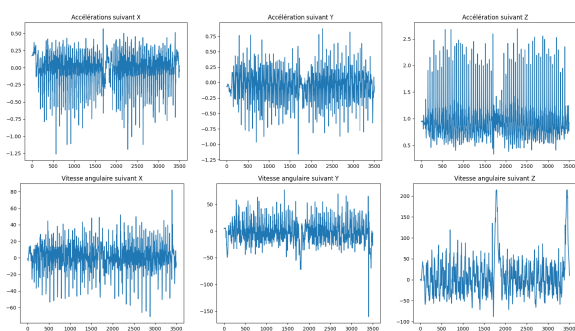
Pour donner un aperçu des données disponibles dans la base de données LTMM, le tableau 1 présente quelques mesures démographiques et de performance fonctionnelle :

La suite de notre étude sera donc focalisée sur l'exploitation et l'analyse de ces données de laboratoire afin de mieux comprendre les mécanismes et facteurs associés au risque de chute chez les personnes âgées.

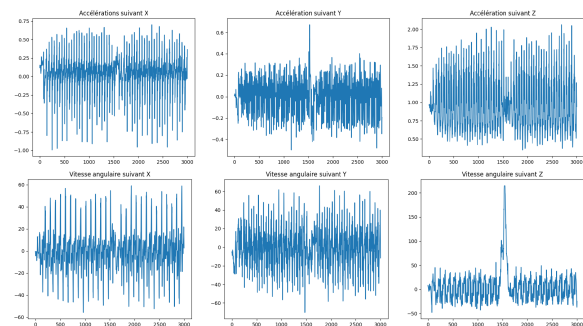
La figure 3.2 et 3.1 présentent un exemple de signaux de l'accéléromètre et de gyroscope des sujets chuteurs et des non chuteurs.

**TABLEAU 3.1** – Données démographiques, performance fonctionnelle et mesures dérivées de l'accélération en laboratoire

	Non-chuteurs	Chuteurs
Nombre de sujets (n)	39	32
Âge (années)	$78.77 \pm 4.39$	$77.86 \pm 5.09$
Genre (% de femmes)	64.10	65.62
Taille (m)	$1.64 \pm 0.06$	$1.61 \pm 0.09$
Poids (kg)	$72.02 \pm 13.36$	$71.94 \pm 12.29$
Indice de masse corporelle (kg/m <sup>2</sup> )	$26.63 \pm 3.99$	$27.86 \pm 3.66$
Années d'éducation	$14.39 \pm 3.18$	$15.20 \pm 3.48$
Examen de l'état mental mini	$28.47 \pm 1.22$	$28.10 \pm 1.58$
Nombre de chutes au cours des 6 derniers mois	$0.15 \pm 0.36$	$2.56 \pm 2.23$
Nombre de faux pas au cours de l'année écoulée	$0.82 \pm 1.75$	$18.5 \pm 52.43$
Vitesse de marche (m/s)	$1.19 \pm 0.24$	$0.97 \pm 0.30$

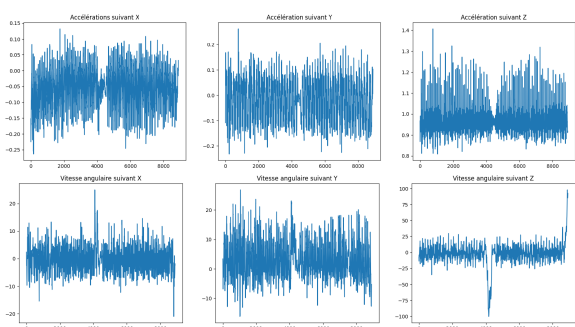


(a) signaux de participant 'Non chuteur 033'

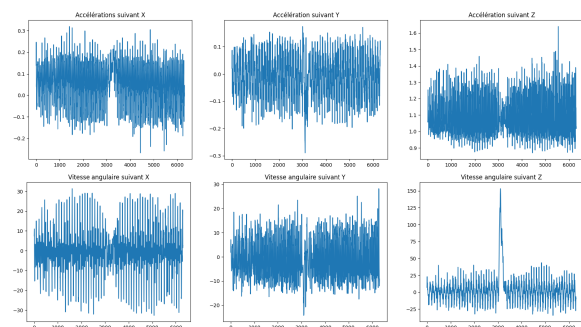


(b) signaux de participant 'Non chuteur 013'

**FIGURE 3.1** – Exemple de la base de données des signaux d'accéléromètres et de gyroscopes des personnes non chuteurs.



(a) signaux de participant 'Chuteur 017'



(b) signaux de participant 'Chuteur 025'

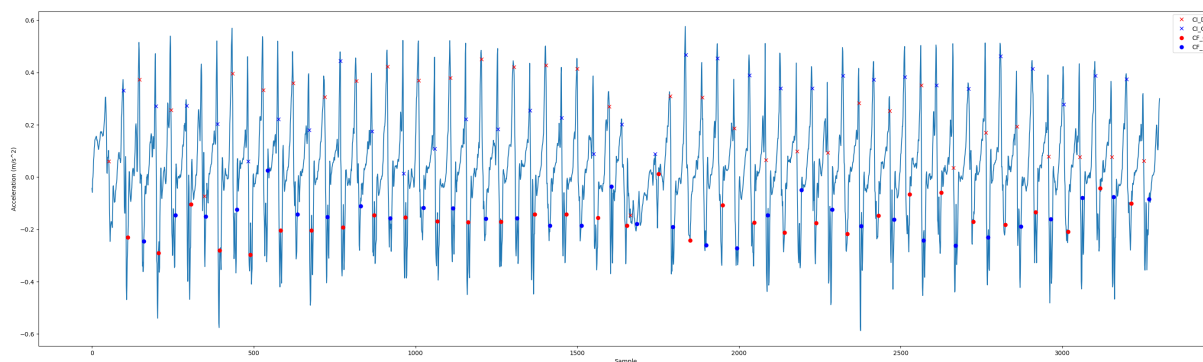
**FIGURE 3.2** – Exemple de la base de données des signaux d'accéléromètres et de gyroscopes des personnes chuteurs.

### 3.3 Application des algorithmes de segmentation pour la détection du Contact Initial et du Contact Final

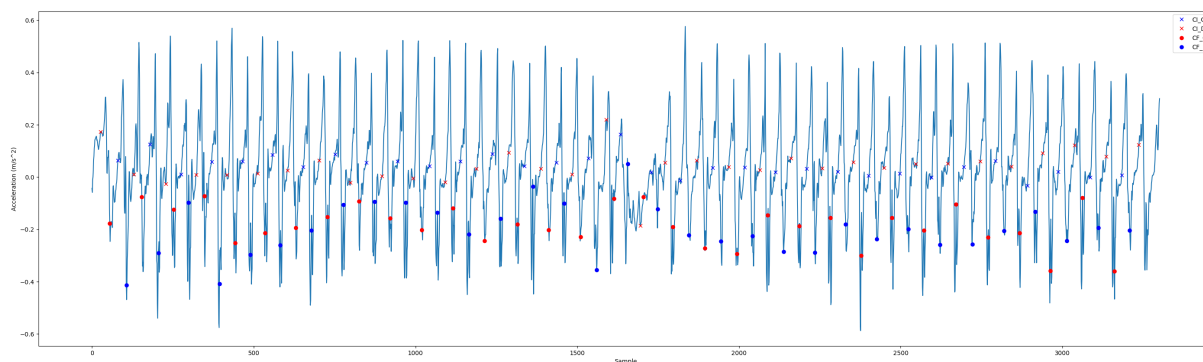
Dans cette section, nous mettons en application les quatre algorithmes de segmentation des signaux que nous avons décrits en détail dans le chapitre 2. Ces algorithmes ont été appliqués à la base de données LTMM dans le but de détecter deux événements clés lors de la marche : le Contact Initial (CI) et le Contact Final (CF).

Une détection précise du CI et du CF est une étape essentielle pour une analyse précise de la marche et une évaluation ultérieure du risque de chute. Cela sera également déterminant pour notre modèle de classification ultérieur visant à distinguer les sujets chuteurs des non-chuteurs.

Afin de mieux visualiser l'application de ces algorithmes, nous avons représenté sur plusieurs figures l'application de chaque algorithme sur l'axe x des signaux d'accéléromètre. Ces figures illustrent la détection des points CI et CF pour chaque algorithme.

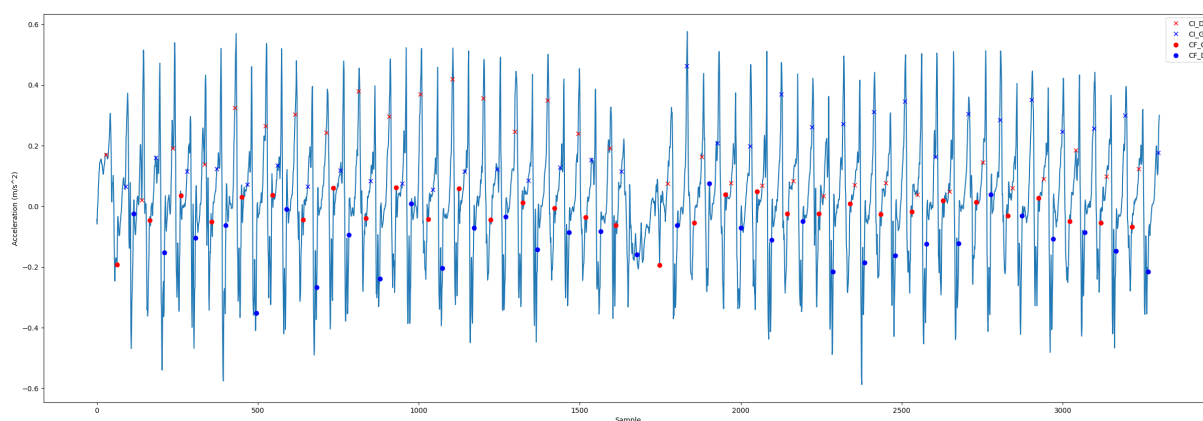


**FIGURE 3.3** – Application de l'algorithme GHARSI : détection des points CI et CF sur l'axe x des signaux d'accéléromètre.

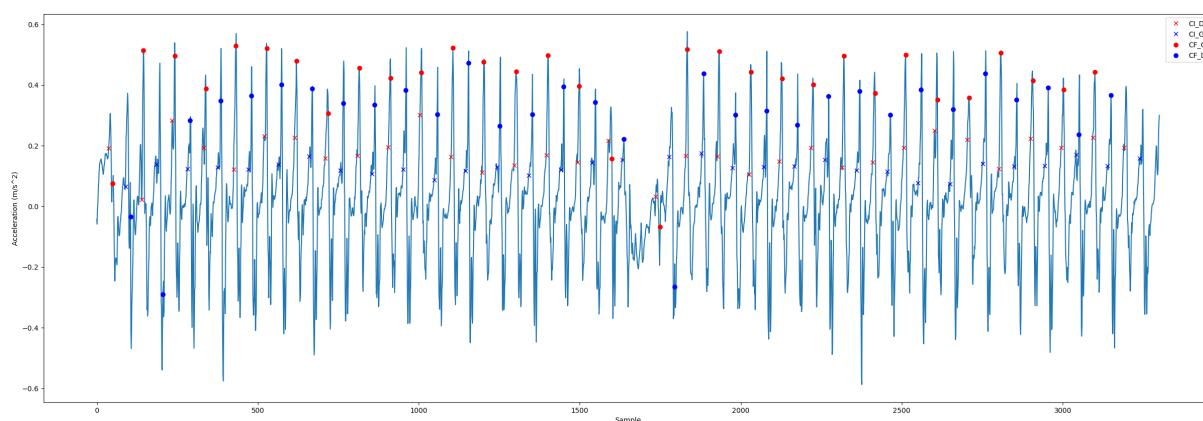


**FIGURE 3.4** – Application de l'algorithme Shin : détection des points CI et CF sur l'axe x des signaux d'accéléromètre.





**FIGURE 3.5** – Application de l’algorithme CWT : détection des points CI et CF sur l’axe x des signaux d’accéléromètre.



**FIGURE 3.6** – Application de l’algorithme Gonzalez : détection des points CI et CF sur l’axe x des signaux d’accéléromètre.

En analysant les figures ci-dessus, nous pouvons observer comment chaque algorithme détecte les points de CI et de CF. La capacité de ces algorithmes à détecter précisément ces événements est cruciale car elle impacte directement la qualité des caractéristiques extraites pour notre modèle de classification des sujets chuteurs et non-chuteurs.

Dans la section suivante, nous présenterons les résultats obtenus à partir de ces algorithmes. Ces résultats, basés sur l’extraction de caractéristiques, nous permettront de déterminer la précision avec laquelle chaque algorithme a été capable de détecter le CI et le CF à partir des données de la base de données LTMM. Cette évaluation de la précision de la détection est une étape cruciale dans notre analyse ultérieure des sujets chuteurs et non-chuteurs.

## 3.4 Analyse Comparative des Algorithmes de segmentation des Paramètres de Marche

Dans cette section, nous présentons les résultats obtenus à partir de quatre algorithmes : GHARSI, Gonzalez, CWT, et SHIN. Les valeurs moyennes pour chaque paramètre de marche ont été calculées pour les chuteurs et non-chuteurs, y compris la cadence, le temps de pas, le temps de la foulée, le temps de double appui, le temps de simple appui, la phase d'appui, la phase d'oscillation, et la symétrie de la marche.

Notre référence pour l'évaluation de ces algorithmes est l'étude présentée dans [72]. Cette étude fournit les valeurs de référence pour chaque paramètre de marche, ce qui nous permet d'évaluer la performance de nos algorithmes. Les résultats de l'étude de référence seront présentés dans le tableau suivant 3.2 pour une comparaison aisée.

	Cadence [step/min]	Temps de pas [s]	Temps de la foulée [s]	Temps de double appui [s]	Temps de simple appui [s]	Phase d'appui [% / 100]	Phase d'oscillation [% / 100]	Symétrie
Chuteurs	106.37 ± 10.88	0.57 ± 0.06	1.15 ± 0.13	0.14 ± 0.04	0.43 ± 0.05	0.62 ± 0.03	0.38 ± 0.03	0.92 ± 0.03
Non-Chuteurs	117.03 ± 10.14	0.52 ± 0.04	1.04 ± 0.08	0.11 ± 0.02	0.40 ± 0.04	0.61 ± 0.02	0.39 ± 0.02	0.91 ± 0.03

**TABLEAU 3.2** – Distribution moyenne des paramètres de la marche spatiale-temporelle chez les chuteurs et non chuteurs

Avec ces valeurs de référence à l'esprit, nous allons maintenant présenter et discuter les performances de chaque algorithme.

### 3.4.1 Évaluation de l'algorithme GHARSI

Nous présentons les résultats de l'algorithme GHARSI dans le tableau .

	Cadence [step/min]	Temps de pas [s]	Temps de la foulée [s]	Temps de double appui [s]	Temps de simple appui [s]	Phase d'appui [% / 100]	Phase d'oscillation [% / 100]	Symétrie
Chuteur	105.18 ± 12.4	0.58 ± 0.08	1.16 ± 0.16	0.104 ± 0.04	0.46 ± 0.08	0.599 ± 0.04	0.4 ± 0.03	0.95 ± 0.03
Non chuteur	114.24 ± 11.7	0.53 ± 0.05	1.05 ± 0.07	0.1 ± 0.03	0.42 ± 0.05	0.595 ± 0.03	0.405 ± 0.03	0.97 ± 0.03

Les résultats obtenus à partir de l'algorithme de Gharsi sont assez proches de ceux de l'étude de référence, notamment pour les non-chuteurs. Cependant, on note une légère différence dans la cadence, le temps de pas et le temps de la foulée pour les chuteurs. Cela pourrait indiquer que l'algorithme de Gharsi a tendance à sous-estimer la cadence et à surestimer le temps de pas et de foulée chez les personnes qui tombent.

### 3.4.2 Évaluation de l'algorithme SHIN

Nous présentons les résultats de l'algorithme SHIN dans le tableau .

	Cadence [step/min]	Temps de pas [s]	Temps de la foulée [s]	Temps de double appui [s]	Temps de s aimpleppui [s]	Phase d'appui [% / 100]	Phase d'oscillation [% / 100]	Symétrie
Chuteur	128.83 ±30.17	0.52 ±0.0	1.04 ±0.09	0.25 ±0.04	0.26 ±0.03	0.74 ±0.02	0.26 ±0.03	/
Non chuteur	122.91 ±11.1	0.51 ±0.03	1.03 ±0.07	0.26 ±0.02	0.25 ±0.03	0.75 ±0.01	0.24 ±0.01	/

Par rapport à l'étude de référence, l'algorithme de Shin présente des résultats significativement différents. En particulier, la cadence est nettement plus élevée pour les Chuteurs. De plus, le temps de double appui est considérablement plus élevé dans les deux groupes par rapport à la référence. Cela pourrait indiquer que l'algorithme de Shin mesure de manière disproportionnée le temps de double appui pendant la marche.

### 3.4.3 Évaluation de l'algorithme Gonzalez

Nous présentons les résultats de l'algorithme SHIN dans le tableau .

	Cadence [step/min]	Temps de pas [s]	Temps de la foulée [s]	Temps de double appui [s]	Temps de s aimpleppui [s]	Phase d'appui [% / 100]	Phase d'oscillation [% / 100]	Symétrie
Chuteurs	103.80 ±11.41	0.59 ±0.097	1.202 ±0.21	0.079 ±0.031	0.52 ±0.087	0.56 ±0.026	0.43 ±0.026	0.9571 ±0.024
Non Chuteurs	114 ±9.14	0.545 ±0.073	1.129 ±0.33	0.085 ±0.037	0.4599 ±0.084	0.58 ±0.038	0.42 ±0.038	0.9579 ±0.085

L'algorithme de Gonzalez présente un bon équilibre entre précision et rapidité. Bien qu'il ne soit pas aussi précis que GHARSI, il reste relativement proche de l'étude de référence, avec quelques petites différences. Par exemple, le temps de pas est légèrement surestimé pour les chuteurs. Le temps de simple appui est également légèrement surestimé par rapport à la référence.

### 3.4.4 Évaluation de l'algorithme CWT

Nous présentons les résultats de l'algorithme CWT dans le tableau .

	Cadence [step/min]	Temps de pas [s]	Temps de la foulée [s]	Temps de double appui [s]	Temps de s aimpleppui [s]	Phase d'appui [% / 100]	Phase d'oscillation [% / 100]	Symétrie
Chuteur	104.3 ±11.05	0.59 ±0.08	1.19 ±0.15	0.29 ±0.51	0.88 ±0.54	0.27 ±0.4	0.72 ±0.4	0.96 ±0.05
Non chuteur	112.61 ±9.44	0.548 ±0.052	1.08 ±0.1	0.14 ±0.66	0.69 ±0.69	0.37 ±0.54	0.62 ±0.54	0.96 ±0.033

L'algorithme CWT, quant à lui, présente une grande variabilité par rapport à l'étude de référence. Cette variation est particulièrement notable pour le temps de simple appui et le temps de double appui. Il est possible que l'algorithme CWT ne soit pas parfaitement adapté à notre jeu de données.

### 3.4.5 Choix de l'algorithme

Dans l'ensemble, il semble que chaque algorithme ait ses forces et faiblesses en termes de mesure des paramètres de marche. Compte tenu de nos observations, nous choisissons

l'algorithme Gonzalez pour notre système. Bien qu'il ne soit pas le plus précis, il offre un bon compromis entre précision et vitesse, ce qui est essentiel pour notre objectif de développer un système fonctionnant avec des capacités de calcul limitées.

La prochaine section de notre étude se penchera sur la présentation de l'ensemble des autres caractéristiques en utilisant la segmentation basée sur l'algorithme Gonzalez, en préparation pour l'entraînement et le paramétrage des modèles d'apprentissage automatique.

### 3.5 Extraction et Présentation des Caractéristiques Additionnelles pour les Modèles d'Apprentissage Automatique

Dans cette section, nous détaillons la préparation de notre base de données pour les modèles d'apprentissage automatique. Cela comprend la présentation de l'ensemble des caractéristiques extraites en utilisant la segmentation basée sur l'algorithme Gonzalez. Cette base de données enrichie servira de base pour l'entraînement et le paramétrage de nos modèles d'apprentissage automatique. Dans cette section, nous nous concentrerons sur la préparation de notre ensemble de données pour l'entraînement et le paramétrage des modèles d'apprentissage automatique. Pour cela, nous allons utiliser les caractéristiques spatiotemporelles et statistiques que nous avons précédemment définies et discutées au Chapitre 2.

Les caractéristiques utilisées dans notre base de données comprennent la cadence, le temps de foulée, le temps de pas, le temps de simple appui, le temps de double appui, la phase d'oscillation, la phase d'appui, la puissance, la variance, l'écart type, la symétrie, la moyenne, l'asymétrie, le maximum, le minimum, et la longueur de pas, sont toutes incluses dans notre base de données. Toutes ces caractéristiques sont résumées dans la figure 3.7.

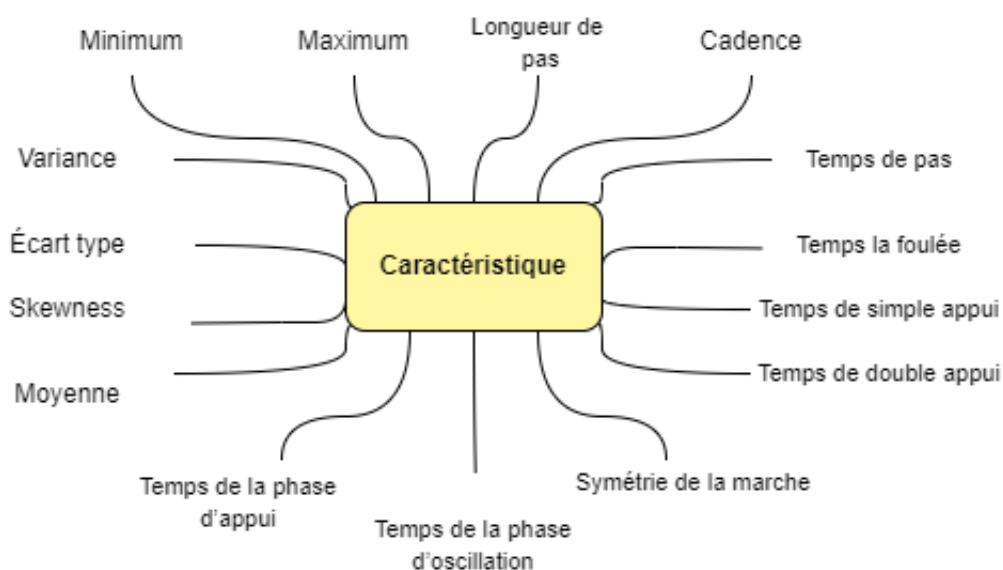


FIGURE 3.7 – L’ensemble des caractéristique extraite.

Afin de distinguer entre les individus qui ont chuté (chuteurs) et ceux qui ne l’ont pas fait (non-chuteurs), nous avons attribué des étiquettes à notre ensemble de données : les chuteurs ont été marqués avec le chiffre 1 et les non-chuteurs avec le chiffre 0.

Avant de procéder à la visualisation de nos données, nous avons effectué une étape de prétraitement consistant à normaliser les données. La normalisation est une étape importante pour garantir que toutes les caractéristiques sont à une échelle comparable, ce qui peut améliorer les performances de nombreux modèles d’apprentissage automatique.

Une fois nos données préparées de manière adéquate en utilisant l’algorithme de Gonzalez, nous avons procédé à une série de visualisations pour mieux comprendre ces caractéristiques extraites. Ces visualisations servent non seulement à offrir une compréhension plus profonde de nos données, mais également à guider l’interprétation des résultats futurs de l’apprentissage automatique. Avant d’aborder en détail l’entraînement et le paramétrage de nos modèles d’apprentissage automatique, nous allons donc présenter et visualiser l’ensemble de ces caractéristiques. Les visualisations réalisées comprennent :

**Distribution des données** : ou diagramme de dispersion, nous avons visualisé la distribution de chaque caractéristique dans l’ensemble de données. Cela nous a aidé à comprendre la gamme de valeurs que chaque caractéristique peut prendre, ainsi qu’à identifier d’éventuelles valeurs aberrantes. Comme le montre la figure 3.8.

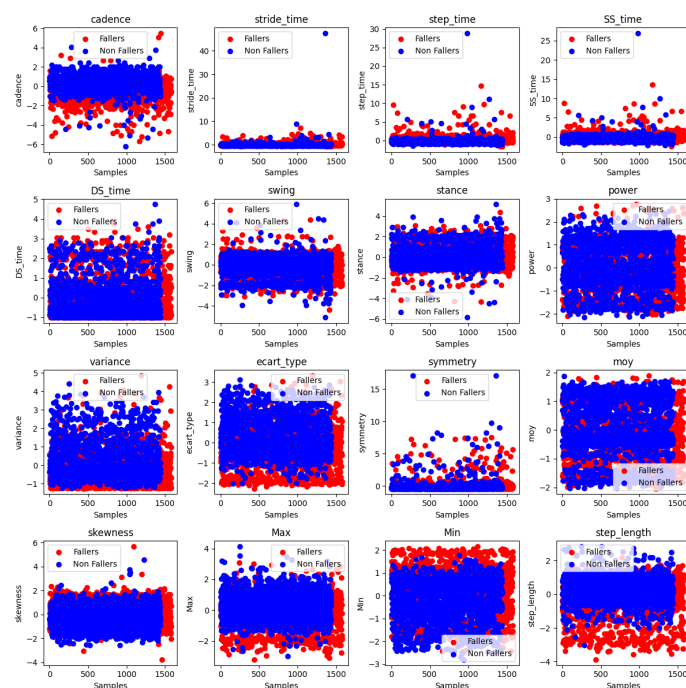


FIGURE 3.8 – Visualisation des données en fonction caractéristiques.

En traçant un diagramme de dispersion, on peut observer la répartition des points des deux classes dans l'espace. Si les points des deux classes se chevauchent considérablement et ne montrent pas de tendance distincte, cela indique une faible capacité de séparation entre les classes pour cette caractéristique. Cela peut rendre difficile la classification des individus en utilisant uniquement cette caractéristique.

En revanche, si les points des deux classes sont clairement séparés et forment des regroupements distincts ou des motifs reconnaissables, cela indique un pouvoir séparatif fort de cette caractéristique. Une telle caractéristique peut être très informative pour la distinction entre les "Chuteurs" et les "non Chuteurs" et peut-être utilisée efficacement dans un modèle de classification. Dans notre cas, les diagrammes de dispersion ne fournissent pas suffisamment d'informations pour tirer des conclusions claires. Les points des deux classes se chevauchent considérablement en raison de la complexité des données, ce qui rend difficile la distinction entre les "Chuteurs" et les "non Chuteurs" en se basant uniquement sur ces diagrammes. Nos données nécessitent d'autres techniques d'analyse pour obtenir des résultats significatifs.

**Diagramme à nuage de points** : nous avons utilisé des diagrammes à nuage de points pour visualiser les relations entre différentes paires de caractéristiques. Comme le montre la figure 3.9.

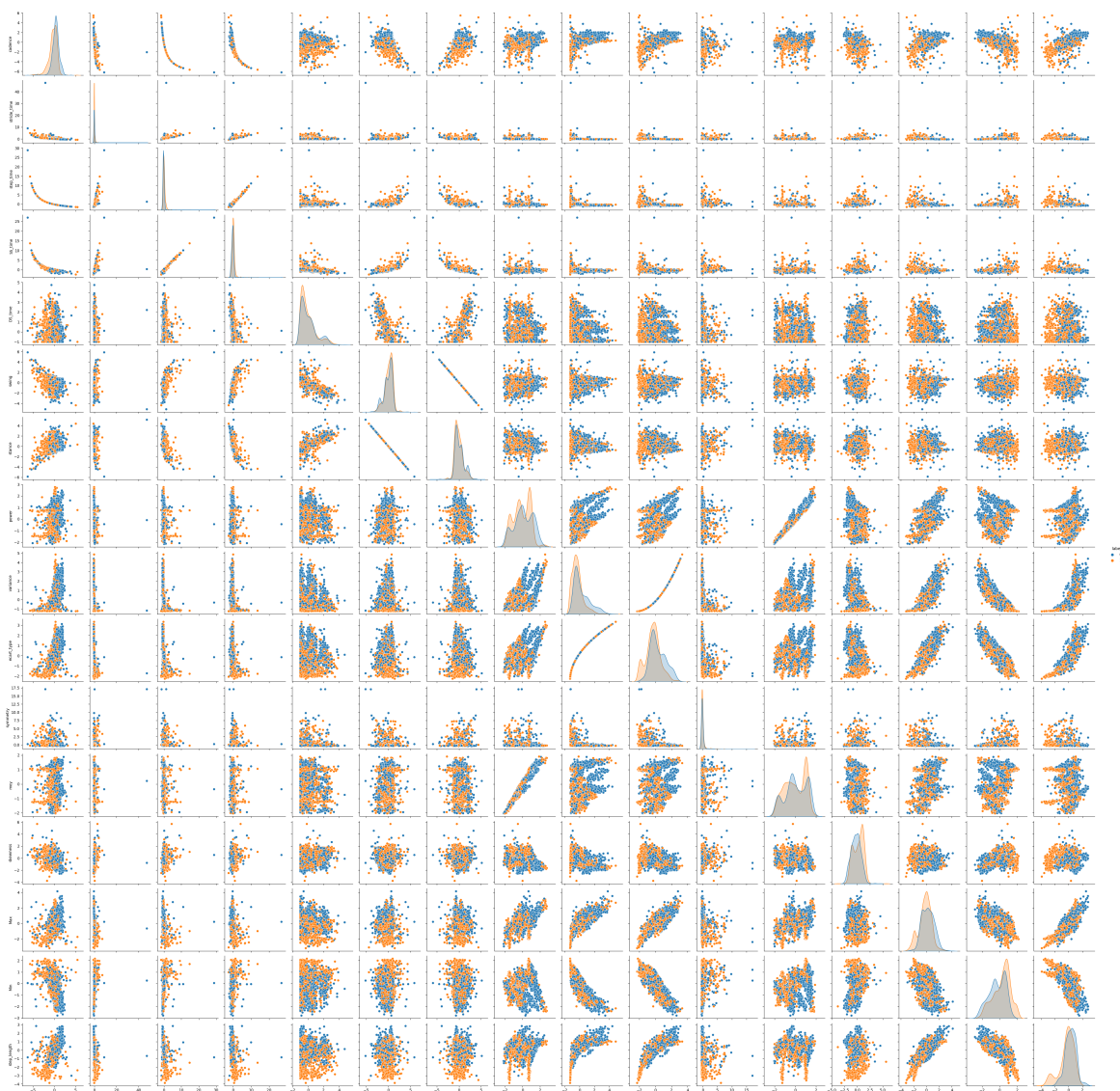


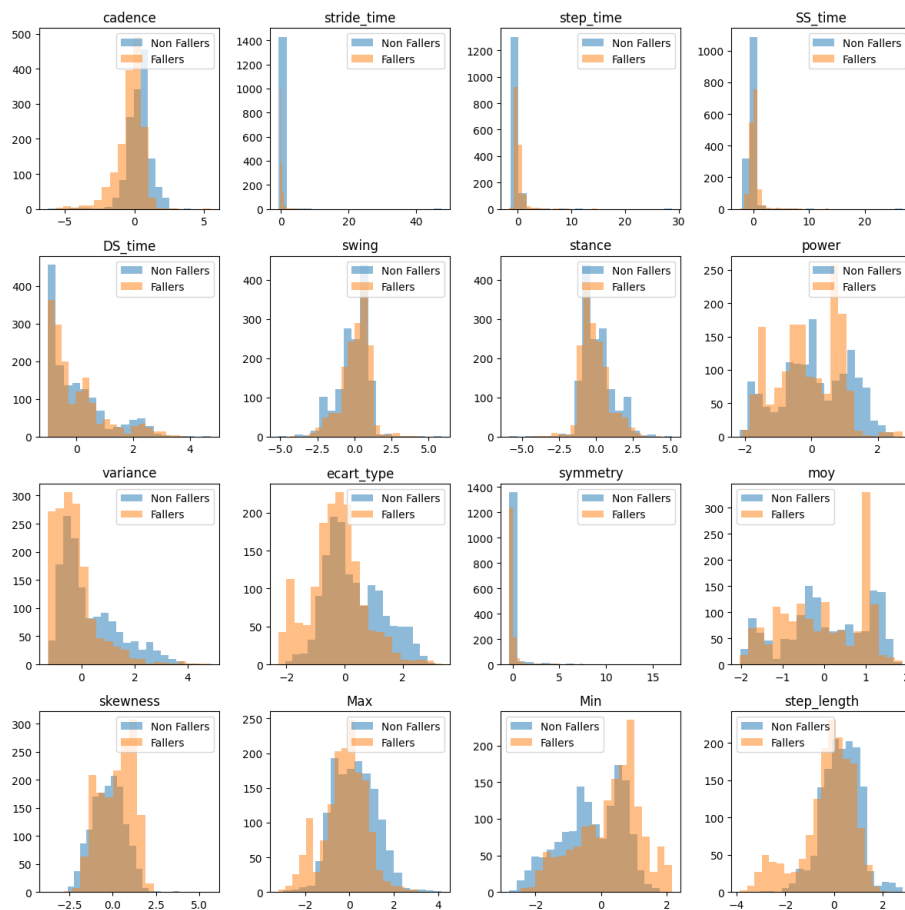
FIGURE 3.9 – Distributions des données en fonctions des caractéristiques.

Le graphique à nuage de points sert à déceler la corrélation entre différentes caractéristiques. L'apparition d'un motif linéaire ou d'une forme de courbe précise au sein de ce graphique peut indiquer une corrélation entre les variables, et la nature de cette corrélation est déterminée par la forme du motif.

En contraste, si les points sont dispersés aléatoirement dans le graphique, cela témoigne de l'absence de corrélation entre les caractéristiques.

Par ailleurs, les points qui se démarquent nettement des autres dans le graphique à nuage de points sont généralement identifiés comme des outliers ou des observations inhabituelles. Ces points pourraient nécessiter une attention particulière lors de l'analyse des données, car ils ont le potentiel d'exercer une influence notable sur les résultats et méritent une investigation plus poussée.

**4- Histogramme** : enfin, nous avons créé des histogrammes pour chaque caractéristique, en séparant les valeurs pour les chuteurs et les non-chuteurs. 3.10.



**FIGURE 3.10** – Histogrammes des caractéristiques : Comparaison entre les chuteurs et les non-chuteurs.

Cette approche nous permet de détecter d'éventuelles différences dans les distributions et de déterminer s'il existe des seuils clairs pour la classification. Par exemple, nous avons constaté que l'histogramme de la caractéristique "temps de cycle" présente des différences significatives entre les classes "Chuteurs" et "non Chuteurs" par rapport aux autres caractéristiques. Cela suggère qu'il existe une relation ou une distinction importante entre ces deux groupes. Cette caractéristique spécifique peut donc jouer un rôle informatif potentiel dans la classification des individus en tant que "Chuteurs" ou "non Chuteurs".

Il existe plusieurs méthodes de visualisation de données. Le choix de la méthode de visualisation dépend du type de données et de leur complexité. Chaque méthode offre un angle de visualisation différent, ce qui peut permettre de mettre en évidence des aspects spécifiques des données. Dans notre étude, nous avons utilisé diverses techniques de visualisation, telles que les histogrammes, les diagrammes de dispersion, et les diagrammes



en boîte, pour analyser les caractéristiques statiques et spatiotemporelles de la marche chez les "Chuteurs" et les "non Chuteurs". Cependant, certaines de ces techniques n'ont pas permis de tirer des conclusions claires. En regroupant les informations fournies par différentes méthodes, nous pouvons obtenir une vision plus complète des données et identifier les relations et les tendances pertinentes. Il est également important de souligner que la visualisation des données ne doit pas être considérée comme une fin en soi, mais plutôt comme un outil pour faciliter la compréhension et l'interprétation des informations contenues dans les données. Elle doit être utilisée en conjonction avec d'autres techniques pour obtenir des résultats significatifs et prendre des décisions éclairées.

Dans la section suivante, notre objectif se déplace vers l'élaboration, l'entraînement et le réglage minutieux de divers modèles d'apprentissage automatique. En utilisant l'ensemble de données que nous avons soigneusement préparé et analysé, ces modèles seront entraînés et ajustés, permettant ainsi d'évaluer leurs performances dans la prédiction des chutes.

## 3.6 Application des techniques de l'apprentissage automatique

Dans cette section, nous allons nous concentrer sur les aspects techniques de l'application de l'apprentissage automatique à notre ensemble de données. Cette section comprend plusieurs sous-sections, chacune abordant une étape cruciale du processus. Nous commencerons par la répartition des données, suivie par la réduction de la dimensionnalité, et enfin, nous aborderons l'entraînement et le paramétrage des modèles d'apprentissage automatique.

### 3.6.1 Répartition des données

Avant de commencer à former nos modèles d'apprentissage automatique, il est essentiel de diviser notre ensemble de données de manière appropriée. Ce partitionnement se fait en trois sous-ensembles distincts :

- Un sous-ensemble de formation utilisé pour concevoir le système de classification, qui se compose d'exemples pour l'apprentissage.
- Un sous-ensemble de validation, qui est un ensemble de données d'exemple utilisé pour régler les hyperparamètres du classificateur et valider l'architecture choisie.
- Un sous-ensemble de test, indépendant de l'ensemble de données d'apprentissage, mais suivant la même distribution de probabilité, utilisé pour évaluer les perfor-

mances de l'architecture du système de classification.

Ces trois sous-ensembles jouent un rôle crucial dans l'évaluation et l'optimisation du système de classification. Le sous-ensemble d'entraînement permet au modèle d'apprendre à partir des exemples fournis, tandis que le sous-ensemble de test fournit une évaluation impartiale des performances du modèle sur des données inédites. Le sous-ensemble de validation permet d'affiner les hyperparamètres du modèle et de s'assurer de la pertinence de l'architecture sélectionnée. L'utilisation de ces sous-ensembles permet d'évaluer et d'améliorer systématiquement l'efficacité et la capacité de généralisation du système de classification.

Durant nos entraînements et évaluations, on a procédé à la répartition présentée dans le tableau 3.4.

Ensembles	Pourcentage (%)	Nombre de cycle
Entraînement	70	2112
Validation	10	302
Test	20	604
Total	100	3018

**TABLEAU 3.3** – Répartition de l'ensemble des caractéristiques.

Cela conclut notre discussion sur la répartition des données. L'étape suivante consiste à réduire la dimensionnalité de nos données.

### 3.6.2 Réduction de la dimension

La réduction de la dimension vise à résoudre ces problèmes en réduisant le nombre de caractéristiques tout en conservant autant d'informations pertinentes que possible. Cela nous permet d'améliorer l'efficacité du processus d'apprentissage et de réduire le risque de surajustement (overfitting) du modèle aux données d'entraînement. Nous avons appliqué une des techniques de réduction de la dimension mentionnée dans le chapitre 1 : l'Analyse en composantes principales.

#### Analyse en composante principale

Dans notre étude, nous avons appliqué l'analyse en composantes principales (ACP) comme technique de réduction de la dimensionnalité sur notre ensemble de caractéristiques extraites.

Pour évaluer l'efficacité de l'ACP, On trace une courbe 3.11 illustrant les variances cumulées en fonction du nombre de composantes principales.

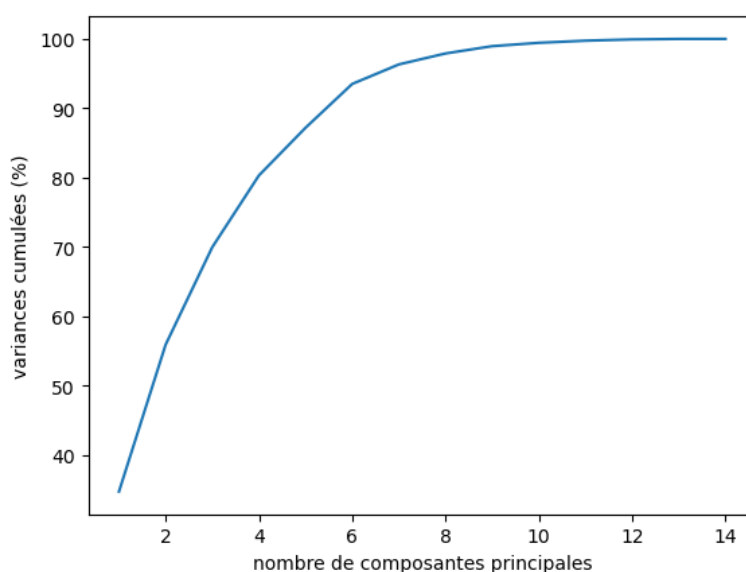


FIGURE 3.11 – Variance cumulée de l'ACP en fonction des caractéristiques extraites.

En fixant le critère de rétention d'au moins 99% de la variance totale, nous avons identifié que 10 composantes principales étaient nécessaires pour atteindre ce seuil. Ces 10 composantes principales ont été sélectionnées en fonction de la quantité d'informations essentielles qu'elles contiennent dans les données réduites.

Composante Principale	Caractéristiques Essentielles	Variance Cumulés
CP1	Ecart-type de l'accélération, Variance de l'accélération, Longueur de pas, Cadence	34.73
CP2	Temps de la phase d'oscillation, Temps de la phase d'appui, Temps de double appui, Temps de simple appui	55.90
CP3	Moyenne de l'accélération, Puissance de l'accélération, Minimum de l'accélération, Longueur de pas	69.97
CP4	Longueur de la foulée, symétrie de la marche, Temps de pas, Temps de double appui	80.32
CP5	Skewness, Maximum de l'accélération, Symétrie de la marche, Longueur de pas	87.19
CP6	Symétrie de la marche, Skewness, Temps de double appui, Temps de pas	93.51
CP7	Longueur de la foulée, Symétrie de la marche, Longueur de pas, Temps de double appui	96.33
CP8	Cadence, Temps de simple appui, Temps de pas, Longueur de pas	97.90
CP9	Longueur de pas, variance de l'accélération, Skewness, Temps de double appui	98.96
CP10	Temps de double appui, Cadence, Temps de la phase d'oscillation, Temps de la phase d'appui	99.44

TABLEAU 3.4 – Tableau des Composantes Principales et Caractéristiques Essentielles.

### Interprétation

L'ACP nous révèle des informations significatives sur la relation entre les caractéristiques de notre ensemble de données réduit.

- La CP1 met en évidence la dynamique et la régularité du mouvement.
- La CP2 est influencée par les différentes phases de la marche, révélant des aspects temporels essentiels.
- La CP3 met en évidence la force et l'amplitude du mouvement de la marche.
- La CP4 souligne la longueur de foulée, la symétrie de la marche et le temps de double appui, reflétant l'amplitude et la symétrie de la marche.
- La CP5 met en évidence la distribution et l'amplitude des mouvements de la marche.
- La CP6 révèle des aspects temporels et de symétrie importants de la marche.
- La CP7 met l'accent sur l'amplitude et la symétrie de la marche.
- La CP8 capture la régularité et le rythme de la marche.
- la CP9 met en évidence la longueur de pas, la variance de l'accélération, le skewness et le temps de double appui.

En résumé, l'ACP nous a permis de comprendre les principales caractéristiques qui

contribuent à la variance des données réduites et nous offre des indications précieuses sur les aspects clés de la marche étudiée. Nous entamons maintenant la modélisation et l'évaluation des algorithmes de machine learning.

Avec la fin de la réduction de la dimensionnalité, nous sommes maintenant prêts à aborder la phase suivante - l'entraînement et le paramétrage des modèles d'apprentissage automatique.

### 3.6.3 Entraînement et paramétrage des modèles d'apprentissage automatique

Nous allons nous intéresser à l'entraînement et au paramétrage des modèles d'apprentissage automatique. Cette étape est cruciale dans le processus d'apprentissage, car elle détermine comment nos modèles vont apprendre à partir des caractéristiques extraites et réduites que nous avons préparées précédemment.

L'entraînement d'un modèle est le processus par lequel il apprend à reconnaître les modèles et à faire des prédictions à partir des données. Cependant, la performance d'un modèle dépend fortement de ses paramètres, que nous devons donc ajuster soigneusement.

Dans cette optique, nous allons utiliser une combinaison de recherche en grille et de validation croisée.

**Recherche en grille (GridSearch)** : la méthode de recherche en grille pour ajuster les hyperparamètres de nos modèles. Cette méthode, parfois appelée "Grid Search", est une technique d'optimisation des hyperparamètres qui consiste à définir une grille d'hyperparamètres potentiels et à évaluer systématiquement la performance du modèle pour chaque combinaison d'hyperparamètres dans cette grille. Cela nous permet de trouver la combinaison d'hyperparamètres qui donne les meilleurs résultats.

**Validation croisée (Cross validation)** : La validation croisée, quant à elle, est une technique d'évaluation de la performance d'un modèle de machine learning. Elle consiste à diviser l'ensemble de données en plusieurs sous-ensembles, appelés plis, et à effectuer plusieurs itérations d'entraînement et de test en utilisant différents plis comme données de test. Cela permet d'estimer de manière plus fiable les performances du modèle, en réduisant les biais liés à la dépendance des données d'entraînement et de test. La validation croisée est particulièrement utile lorsque les données d'apprentissage sont limitées, car elle permet d'utiliser efficacement toutes les observations pour évaluer le modèle. L'ensemble de données sera divisé en un ensemble d'entraînement et un ensemble de test. Les données d'entraînement seront divisées en  $K$  sous-ensembles. Nous formerons ensuite le modèle sur  $K-1$  de ces sous-ensembles et évaluerons ses performances sur le sous-ensemble restant. Ce processus sera répété  $K$  fois, chaque sous-ensemble étant utilisé exactement une fois

comme données de validation. La moyenne des résultats de ces  $K$  évaluations est ensuite calculée pour produire une mesure d'évaluation finale du modèle. Ce processus sera répété pour différents hyperparamètres. Ensuite, la moyenne et l'écart-type des différents modèles sont comparés et un modèle est sélectionné parmi tous les modèles. Le modèle est ensuite testé sur l'ensemble de données de test mis de côté.

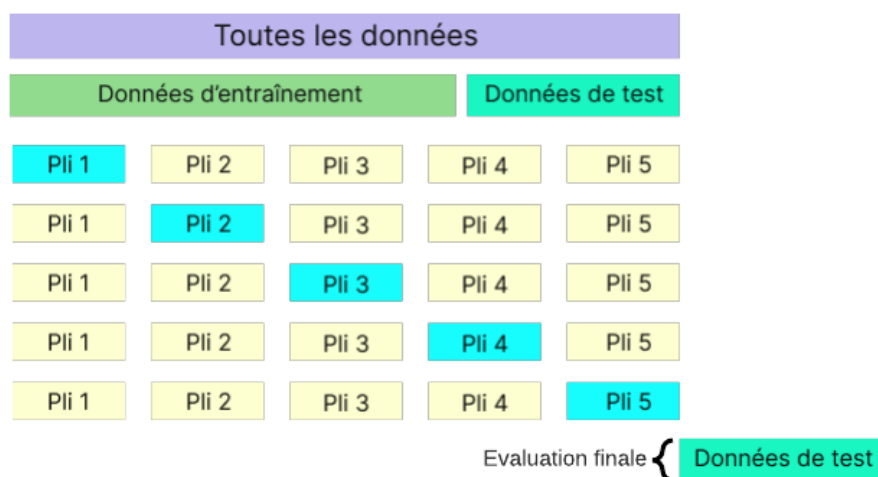


FIGURE 3.12 – Validation croisée

La combinaison des deux est une approche, souvent appelée "Grid Search Cross-Validation", est une technique robuste pour l'optimisation des hyperparamètres et l'évaluation des modèles. La recherche en grille nous permet d'explorer systématiquement une gamme d'hyperparamètres potentiels pour trouver la combinaison qui donne les meilleurs résultats. La validation croisée, quant à elle, améliore l'estimation des performances du modèle en réduisant les biais potentiels associés à la sélection des données d'entraînement et de test. En combinant ces deux techniques, nous pouvons entraîner et ajuster nos modèles, y compris les  $K$ -plus proches voisins, le SVM, l'arbre de décision, les forêts aléatoires, les réseaux de neurones et le réseau neuronal convolutif 1d, de manière à optimiser leurs performances sur l'ensemble des données.

Nous allons examiner en détail la grille d'hyperparamètres que nous utiliserons, et discuter des hyperparamètres spécifiques de chaque modèle que nous allons entraîner, y compris les  $K$ -plus proches voisins, le SVM, l'arbre de décision, les forêts aléatoires, les réseaux de neurones, et le réseau neuronal convolutif 1d. Pour chacun de ces modèles, nous déterminerons les meilleurs hyperparamètres en utilisant la méthode de recherche en grille.

Finalement, nous présenterons les meilleurs hyperparamètres pour chaque modèle

d'apprentissage automatique que nous avons sélectionnés. C'est une étape cruciale pour garantir que nos modèles sont bien paramétrés pour donner les meilleurs résultats possibles sur nos données.

### 3.6.3.1 Grille d'hyperparamètres

La grille d'hyperparamètres est un ensemble de valeurs possibles pour chaque hyperparamètre du modèle. L'objectif de la recherche en grille est de trouver la combinaison d'hyperparamètres qui permettra d'ajuster de manière optimale le modèle aux données d'entraînement, tout en évitant le surajustement.

Dans le code, la grille d'hyperparamètres peut être définie en créant un dictionnaire où chaque clé est le nom d'un hyperparamètre et chaque valeur est une liste de valeurs à explorer.

```
param_grid = {
'paramètre1': [valeur1, valeur2, valeur3],
'paramètre2': [valeur4, valeur5, valeur6]
}
```

En ce qui suit, nous discuterons pour chaque modèle les hyperparamètres possible.

Faute noter que Le K-plus proches voisins est un algorithme non paramétrique. Cependant, il comporte un seul paramètre qui est le nombre de voisins  $k$ , qui doit être spécifié avant l'entraînement du modèle.

### 3.6.3.2 Les hyperparamètres du SVM

Les hyperparamètres pouvant être variés, dans les classeurs de SVM de décision sont :

- $C$  : le paramètre de régularisation.
- Kernel : le noyau utilisé pour la transformation des données.
- Gamma : le coefficient du noyau.

Voici le tableau correspondant à l'exemple de la grille d'hyperparamètres du Support Vector Machine (SVM) :

Paramètre	Valeurs possibles
$C$	[1, 3, 5, 10, 15, 100]
kernel	['linear', 'rbf']
gamma	[0.1, 1, 10]

**TABLEAU 3.5** – Exemple de la grille d'hyperparamètres du Support Vector Machine (SVM)

### 3.6.3.3 Les hyperparamètres de l'algorithme Arbre de Décision

Les hyperparamètres les plus importants dans l'algorithme Decision Tree (DT) sont les suivants :

- Critère de division (criterion) : il détermine la mesure utilisée pour évaluer la qualité de la division d'un nœud. Les critères couramment utilisés sont "gini" pour l'indice de Gini et "entropy" pour l'entropie.
- Profondeur maximale des arbres (max\_depth) : cela contrôle la profondeur maximale de chaque arbre dans la forêt. Une profondeur plus élevée peut conduire à un surajustement (overfitting), tandis qu'une profondeur plus faible peut entraîner un sous-ajustement (underfitting). Il est courant de laisser ce paramètre à sa valeur par défaut (None), permettant aux arbres de se développer jusqu'à ce que toutes les feuilles soient pures ou jusqu'à ce que le nombre minimum d'échantillons par feuille soit atteint.
- Nombre minimum d'échantillons par feuille (min\_samples\_leaf) : cela indique le nombre minimum d'échantillons requis pour qu'un nœud de l'arbre soit considéré comme une feuille. L'augmentation de cette valeur peut aider à éviter le surajustement.
- Nombre minimum d'échantillons requis pour effectuer une division (min\_samples\_split) : cela définit le nombre minimum d'échantillons requis pour diviser un nœud interne de l'arbre. Une valeur plus élevée peut éviter le surajustement en limitant les divisions supplémentaires dans l'arbre.

On peut observer dans le tableau ci-dessous un exemple de grille d'hyperparamètres pour l'arbre de décision.

Paramètre	Valeurs possibles
'criterion'	['gini', 'entropy']
'max_depth'	[5, 10, 15, 20]
'min_samples_split'	[2, 5, 10]
'min_samples_leaf'	[1, 3, 5]
'max_features'	None
'splitter'	['random', 'best']

TABLEAU 3.6 – Exemple de la grille d'hyperparamètres du l'arbre de décision.

### 3.6.3.4 Les hyperparamètres de l'algorithme Forêts Aléatoires

Pour le modèle Forêts Aléatoires (Random Forest), il utilise les mêmes hyperparamètres que l'arbre de décision, mais avec un hyperparamètre supplémentaire :



- Nombre d'arbres (`n_estimators`) : spécifie le nombre d'arbres de décision indépendants dans la forêt aléatoire. Un nombre plus élevé d'arbres peut améliorer les performances, mais cela augmente également le temps d'entraînement du modèle.

Le tableau ci-dessous détaille les valeurs des hyperparamètres pour le modèle Random Forest

Paramètre	Valeurs possibles
<code>n_estimators</code>	[2, 5, 10, 15]
<code>'max_depth'</code>	[5, 10, 15, 20],
<code>'criterion'</code>	['gini', 'entropy'],
<code>'min_samples_split'</code>	[2, 5, 10],
<code>'min_samples_leaf'</code>	[1, 3, 5],
<code>'max_features'</code>	None,
<code>'splitter'</code>	['random', 'best'],

TABLEAU 3.7 – Exemple de la grille d'hyperparamètres du Random forest (RF).

### 3.6.3.5 Les hyperparamètres dans les réseaux de neurones (DNN)

Les hyperparamètres couramment utilisés dans les réseaux de neurones sont :

- `Activation_function` : c'est la fonction d'activation. Chaque couche du réseau peut utiliser une fonction d'activation pour introduire une non-linéarité. Les fonctions d'activation courantes incluent la fonction sigmoïde, la fonction ReLU (Rectified Linear Unit), la fonction tangente hyperbolique, etc.
- `Learning_Rate` : c'est le taux d'apprentissage auquel les poids du réseau sont mis à jour pendant l'entraînement.
- `Batch_size` : c'est le nombre d'échantillons d'entraînement utilisés pour mettre à jour les poids du réseau à chaque itération.
- `Epochs` : c'est le nombre de fois que le réseau passe par l'ensemble des données d'entraînement complet.
- `Optimizer` : il s'agit de l'algorithme utilisé pour optimiser les poids du réseau pendant l'entraînement. Des optimiseurs populaires incluent l'optimiseur stochastique de descente de gradient (SGD), l'optimiseur Adam, l'optimiseur RMSprop, etc.

Paramètre	Valeurs possibles
'hiddenLayerOne'	[64, 128, 256],
'hiddenLayerTwo'	[16, 32, 64],
'learning_Rate'	[0.1, 0.005, 0.01, 0.001],
'epochs'	[100,200],
'Batch_size'	[16,32],
'activation_function'	['relu', 'tanh'],
'optimizer'	['Adadelata', 'Adagrad', 'Adam', 'RMSprop', 'SGD'],

TABLEAU 3.8 – Exemple de la grille d’hyperparamètres du réseau de neurones (DNN).

### 3.6.3.6 Les hyperparamètres dans le réseau neuronal convolutif 1d (CNN1D)

Les hyperparamètres couramment utilisés dans le réseau neuronal convolutif 1d sont :

- Filters : le nombre de filtres de convolution.
- Kernel\_size : la taille des filtres de convolution.
- Strides : le pas utilisé lors de la convolution.
- Activation : la fonction d’activation utilisée.
- Pooling : le type de pooling appliqué après la convolution (max pooling, average pooling, etc.).
- Dense\_units : le nombre d’unités dans les couches denses.
- Optimizer : l’optimiseur utilisé pour la mise à jour des poids du modèle.
- Learning\_rate : le taux d’apprentissage utilisé par l’optimiseur.
- Nombre de couches : le nombre total de couches dans le modèle, y compris les couches de convolution, les couches de pooling et les couches denses. Cela peut inclure à la fois les couches visibles (entrée) et les couches cachées.

Voici un exemple de grille de valeurs des hyperparamètres pour l’algorithme CNN1D.

Paramètre	Valeurs possibles
'filters'	[32, 64, 128],
'kernel_size'	[2, 3, 5],
'strides'	[1, 2],
'activation'	['relu', 'tanh'],
'pooling'	['max', 'average'],
'dense_units'	[128, 256, 512],
'optimizer'	['adam', 'rmsprop'],
'learning_rate'	[0.001, 0.01, 0.1]

TABLEAU 3.9 – Exemple de la grille d’hyperparamètres CNN1D.

### 3.6.3.7 Sélection des meilleurs hyperparamètres pour les modèles d'apprentissage automatique

Après avoir effectué une exploration approfondie des hyperparamètres à l'aide de la méthode du GridSearchCV pour les six modèles d'apprentissage automatique mentionnés précédemment, nous avons identifié les meilleures configurations pour chaque modèle.

Modèle	Hyperparamètres
SVM	Noyau : RBF
	Paramètre C : 4
	Paramètre gamma : 1
K-plus proches voisins	Nombre de voisins : 3
Arbre de décision	Critère de division : gini
	Profondeur maximale : 10
	Nombre minimum d'échantillons par feuille : 2
	Nombre minimum d'échantillons pour une division : 1
Random Forrest	Nombre d'arbres : 100
	Profondeur maximale des arbres : 15
	Critère de division : gini
	Nombre minimum d'échantillons par feuille : 1
	Nombre minimum d'échantillons pour une division : 2

**TABLEAU 3.10** – Tableau des hyperparamètres optimaux pour les modèles d'apprentissage automatique

En ce qui concerne le DNN et 1D-CNN, plusieurs architectures différentes ont été conçues. Les hyperparamètres optimale de ces deux modèles sont visibles dans le tableau [3.11](#) et [3.12](#)

	Hyperparamètre	
Couche 1	Type de couche	Dense
	Nombre d'unités	128
	Fonction d'activation	Relu
Couche 2	Type de couche	Dense
	Nombre d'unités	16
	Fonction d'activation	Relu
Couche 3	Type de couche	Dense
	Nombre d'unités	8
	Fonction d'activation	Relu
Couche 4	Type de couche	Dense
	Nombre d'unités	1
	Fonction d'activation	Sigmoïde
Optimiseur	Adam	
Taux d'apprentissage	0.01	
Epoque	100	
Batch size	16	

TABLEAU 3.11 – Meilleurs hyperparamètres pour le modèle DNN.

	Hyperparamètre	
Couche 1	Type de couche	Conv1D
	Nombre de filtres	8
	Taille de noyau	2
	Type de couche	Dense
Couche 2	Nombre d'unités	16
	Fonction d'activation	Relu
	Dropout (0.5)	
Couche 3	Type de couche	Dense
	Nombre d'unités	4
	Fonction d'activation	Relu
	Dropout (0.5)	
Couche 4	Type de couche	Dense
	Nombre d'unités	1
	Fonction d'activation	Sigmoïde
Optimiseur	Adam	
Taux d'apprentissage	0.01	
Epoque	100	
Batch size	16	

TABLEAU 3.12 – Meilleurs hyperparamètres pour le modèle CNN1D.

Avec l'achèvement de l'entraînement et du paramétrage des modèles d'apprentissage automatique, nous sommes maintenant prêts à présenter et à discuter des résultats dans

la section suivante.

## 3.7 Présentation des résultats

Après le paramétrage minutieux de notre pipeline de traitement de données, depuis la sélection initiale de la base de données jusqu'à la configuration des hyperparamètres de nos modèles d'apprentissage automatique, nous sommes maintenant en mesure d'analyser et de discuter des résultats obtenus.

Dans cette section, nous allons discuter de la performance de nos modèles en utilisant différentes métriques pertinentes. Nous présenterons également les résultats obtenus à partir de deux stratégies distinctes de répartition des données.

Nous commencerons par définir les métriques d'évaluation pour mesurer les performances des modèles.

### 3.7.1 Métriques d'évaluation pour évaluer les performances des modèles

Les différents modèles d'apprentissage automatique ont besoin d'une ou de plusieurs mesures pour évaluer leurs performances. Elles nous aideront à quantifier les performances d'un modèle d'apprentissage et à identifier rapidement les points faibles de modèle.

**Matrice de confusion :** La matrice de confusion est un élément essentiel qui peut être utilisé pour mesurer la performance du modèle de classification ML, mais elle n'est pas considérée comme une métrique. Elle fournit une visualisation de la façon dont un modèle de classification fait des erreurs lors de ses prédictions. Elle permet de comprendre non seulement les erreurs commises, mais aussi le type spécifique d'erreurs commises.

- **Vrai positif (VP) :** une classe est prédite comme vraie et qu'elle est vraie réalité.
- **Vrai négatif (VN) :** une classe est prédite comme fausse et qu'elle fausse en réalité.
- **Faux positif (FP) :** une classe est prédite vraie, mais elle est fausse en réalité.
- **Faux négatif (FN) :** une classe est prédite fausse, mais elle est vraie en réalité.

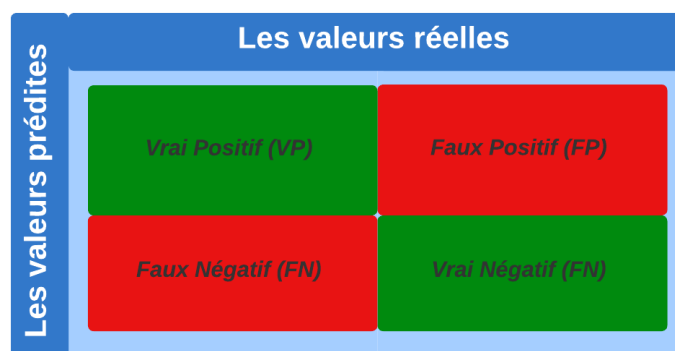


FIGURE 3.13 – Matrice de confusion.

Il existe plusieurs métriques d'évaluation pour évaluer les performances des modèles de machine learning. Voici quelques métriques les plus couramment utilisées :

1. **Précision (Precision)** : cette métrique mesure la proportion de prédictions positives correctes parmi toutes les prédictions positives. Elle est utile lorsque le coût des faux positifs est élevé.

$$Precision = \frac{\text{Vrai positif}}{\text{Vrai positif} + \text{Faux positif}} \quad (3.1)$$

2. **Rappel (Recall)** : le rappel mesure la proportion de prédictions positives correctes parmi toutes les observations positives réelles. Cette métrique est utile lorsque le coût des faux négatifs est élevé.

$$Rappel = \frac{\text{Vrai positif}}{\text{Vrai positif} + \text{Faux négatif}} \quad (3.2)$$

3. **F1-score** : le F1-score est la moyenne harmonique de la précision et du rappel. Il est utilisé lorsque vous souhaitez avoir une mesure équilibrée entre la précision et le rappel.

$$F1 - score = 2 * \frac{\text{Précision} * \text{Rappel}}{\text{Précision} + \text{Rappel}} \quad (3.3)$$

4. **Exactitude (Accuracy)** : l'exactitude mesure la proportion de prédictions correctes parmi toutes les prédictions. C'est une métrique générale qui fonctionne bien lorsque les classes sont équilibrées.

$$Exactitude = \frac{\text{Vrai positif} + \text{Vrai négatif}}{\text{Vrai négatif} + \text{Vrai positif} + \text{Faux négatif} + \text{Faux positif}} \quad (3.4)$$

5. **Aire sous la courbe ROC (AUC-ROC)** : une courbe ROC (Receiver Opera-

ting Characteristic) est un graphique qui illustre les performances d'un modèle de classification à différents seuils de classification. Elle est créée en traçant le taux de vrais positifs (TPR) en fonction du taux de faux positifs (FPR) à différents seuils. La courbe ROC permet d'évaluer la capacité du modèle à discriminer entre les classes positives et négatives. Plus la courbe se rapproche du coin supérieur gauche du graphique, meilleure est la performance du modèle. Une courbe diagonale représente une performance aléatoire, tandis qu'une courbe parfaitement au-dessus de la diagonale indique une performance idéale.

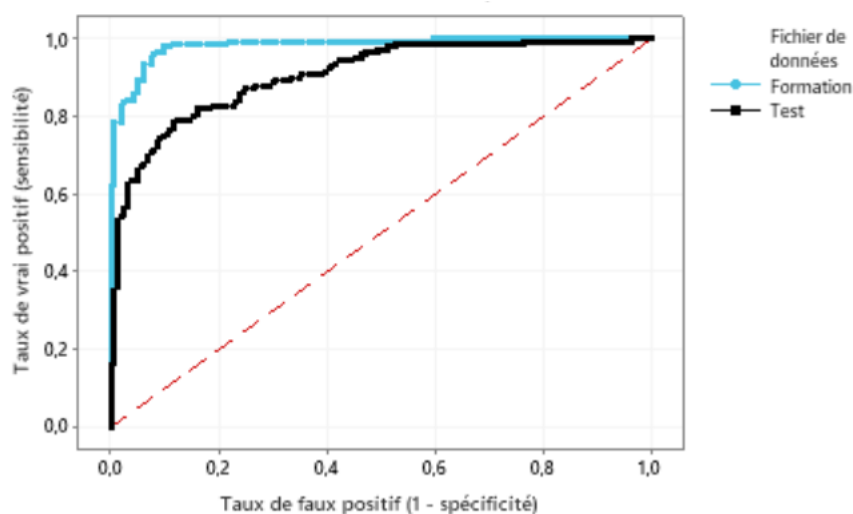


FIGURE 3.14 – La courbe ROC.

### 3.7.2 Performances des modèles

Dans cette section, nous analyserons les performances de différents algorithmes d'apprentissage automatique à travers deux perspectives distinctes.

Le premier tableau 3.13 fournit l'exactitude globale des algorithmes. Nous examinerons les indicateurs d'exactitude, la courbe ROC et le temps d'exécution pour une bonne compréhension de la manière dont ces modèles se comportent en général.

Algorithme	Exactitude entraînement [%]	Exactitude validation [%]	Exactitude test [%]	ROC [%]	Temps d'exécution [s]
<b>SVM</b>	99.52	87.90	87.6	88	0.25
<b>KNN</b>	93.62	88.54	89.7	90	0.008
<b>DT</b>	94.43	79.62	77.2	78	0.05
<b>DT avec adaboost</b>	100	86.62	85.8	86	1.8
<b>RF</b>	100	85.67	84.9	85	1.08
<b>RF avec adaboost</b>	100	86.62	84.9	85	1.12
<b>DNN</b>	96.38	88.85	89.4	89	29
<b>CNN 1D</b>	90.9	80.51	85.7	86	31.08

**TABLEAU 3.13** – Performances Comparatives des Modèles d'Apprentissage Automatique.

On constate que tous les modèles ont une exactitude d'entraînement élevée, les algorithmes ensemblistes atteignant même une exactitude parfaite de 100%. Cependant, l'exactitude en validation et en test, est nettement plus basse pour tous les modèles. Cela pourrait indiquer un surapprentissage lors de l'entraînement, où les modèles se sont trop bien adaptés aux données d'entraînement et ne généralisent pas bien à de nouvelles données.

Le tableau se concentre davantage sur des mesures spécifiques, à savoir le F1-score, le rappel et la précision. Ces mesures sont particulièrement importantes pour comprendre comment ces modèles distinguent entre les classes "chute" et "non chute".



	Précision [%]		Rappel [%]		F1-score [%]	
	Chute	Non-Chute	Chute	Non-Chute	Chute	Non-Chute
<b>SVM</b>	91	89	91	88	91	89
<b>KNN</b>	91	89	91	88	91	88
<b>DT</b>	74	81	83	71	79	76
<b>DT avec adaboost</b>	86	85	88	83	87	84
<b>RF</b>	86	84	87	82	86	83
<b>RF avec adaboost</b>	84	85	88	81	86	83
<b>DNN</b>	91	87	90	88	91	88
<b>CNN 1D</b>	88	83	87	84	87	84

**TABLEAU 3.14** – Précision, Rappel et Score F1 des Modèles d'Apprentissage Automatique pour la Classification de Chute et Non-Chute

Le tableau 3.14 donne une image plus détaillée de la performance de chaque modèle en termes de précision, de rappel et de score F1 pour la classification de "chute" et "non-chute". En général, les scores F1 pour la classe "chute" sont légèrement supérieurs à ceux de la classe "non-chute" pour tous les modèles, ce qui signifie qu'ils sont légèrement meilleurs pour identifier correctement les chutes.

## 3.8 Discussion

- L'algorithme SVM, KNN et DNN se démarquent en termes de performances générales. Ces modèles ont montré la meilleure précision sur les données d'entraînement et de test, ainsi qu'un excellent score ROC, indicateur de leur capacité à distinguer entre les classes positives et négatives.
- Surapprentissage : Le fait que tous les modèles affichent une précision d'entraînement très élevée, certains atteignant même une précision parfaite de 100%, tandis que la précision de validation et de test est sensiblement plus faible, indique un possible surapprentissage. Cela signifie que les modèles se sont très bien adaptés aux données d'entraînement, mais n'ont pas réussi à généraliser de manière aussi

efficace aux nouvelles données. C'est un aspect crucial à prendre en compte lors de l'amélioration de ces modèles, car un modèle idéal devrait être capable de bien généraliser à de nouvelles données non vues pendant l'entraînement.

- Comparaison des modèles : Les modèles SVM, KNN et DNN semblent offrir les meilleures performances globales en termes de précision, de rappel et de score F1, pour les classes "chute" et "non-chute". Ces modèles sont donc potentiellement les plus utiles pour notre objectif d'identifier correctement les chutes.
- Temps d'exécution : Le temps d'exécution est un facteur important à considérer pour des applications en temps réel ou pour des systèmes avec des capacités de calcul limitées. Ici, le KNN se distingue par son temps d'exécution extrêmement rapide.
- Modèles ensemblistes : Les modèles ensemblistes (DT avec Adaboost et RF avec Adaboost) atteignent une précision d'entraînement de 100% mais ne parviennent pas à maintenir une telle performance sur les données de validation et de test. Cela peut suggérer que ces modèles sont peut-être trop complexes et s'ajustent trop bien aux données d'entraînement, entraînant un surapprentissage.
- Courbe ROC : Les scores ROC sont généralement bons pour tous les modèles, ce qui indique une bonne performance en termes de taux de vrais positifs par rapport aux faux positifs. Cependant, il est important de noter que même si un score ROC élevé est généralement une bonne indication, il ne doit pas être utilisé seul pour évaluer la performance d'un modèle.
- Identification des chutes : L'identification correcte des chutes est cruciale dans cette étude car toute erreur peut avoir des conséquences significatives sur la fiabilité du système et la sécurité des utilisateurs. Les modèles ont tendance à obtenir des scores F1 légèrement plus élevés pour la classe "chute" par rapport à la classe "non-chute", indiquant une performance globalement meilleure dans la détection précise des chutes. Les modèles les plus performants en termes d'identification des chutes sont le SVM, le KNN et le DNN, qui présentent à la fois une haute précision et un bon rappel. En revanche, les modèles basés sur les arbres de décision (DT et RF, avec ou sans Adaboost) et le CNN 1D semblent moins performants. Malgré une précision d'entraînement élevée, ils peuvent souffrir d'un surajustement nuisant à leur capacité à généraliser sur de nouvelles données. En conclusion, bien que le SVM, le KNN et le DNN se distinguent en termes d'identification des chutes, le choix du modèle optimal nécessite également de considérer d'autres facteurs, tels que le temps d'exécution, en particulier pour une utilisation en temps réel.

## 3.9 Conclusion

Dans ce chapitre, nous avons mené une étude approfondie en utilisant la base de donnée "Long Term Movement Monitoring" de Physionet pour l'application des techniques d'apprentissage automatique pour l'analyse de la marche, spécifiquement orientée vers personnes âgées. Cette base de données, avec ses caractéristiques distinctes et sa richesse en informations, s'est avérée être un choix approprié pour notre étude.

Nous avons examiné en détail quatre algorithmes de segmentation - ceux de Shin, de Gharsi, de Gonzalez et de CWT - afin de détecter deux événements clés dans la marche, à savoir le CI et le CF. Une analyse comparative de ces algorithmes, appuyée par une revue de la littérature pertinente, nous a permis de constater que chaque algorithme a ses propres forces et faiblesses en termes de mesure des paramètres de marche. Malgré cela, l'algorithme de Gonzalez a été choisi pour notre système en raison de son compromis équilibré entre la précision et la rapidité, une considération essentielle pour notre objectif de mise en œuvre en temps réel sur des dispositifs aux capacités de calcul limitées.

Après la segmentation, nous avons effectué une extraction détaillée des caractéristiques spatio-temporelles et statistiques de notre base de données, créant ainsi un ensemble de données enrichi pour nos modèles d'apprentissage automatique. Cet ensemble a ensuite été visuellement exploré pour une meilleure compréhension.

Pour l'application des algorithmes d'apprentissage automatique, nous avons commencé par diviser notre ensemble de données de manière appropriée, suivie par une réduction de la dimensionnalité pour améliorer l'efficacité de nos modèles. Nous avons ensuite entraîné et paramétré divers modèles d'apprentissage automatique, en veillant à choisir les meilleurs hyperparamètres.

L'évaluation des performances de nos modèles a révélé que les algorithmes SVM, KNN et DNN se sont particulièrement distingués. Ces modèles ont démontré une excellente précision sur les données d'entraînement et de test, un score ROC impressionnant, et ont atteint le score F1 le plus élevé pour les deux classes de notre ensemble de données. Ces résultats soulignent leur capacité à identifier correctement les chutes et les non-chutes, faisant d'eux des choix solides pour notre objectif de détection des chutes en temps réel.

# Conclusion Générale

En conclusion, notre travail a mis en évidence l'importance cruciale de la détection des chutes chez les personnes âgées, une question de santé publique préoccupante. Les données provenant de la base de données Physionet "Long Term Movement Monitoring" ont été analysées, en utilisant une seule unité de mesure inertielle placée dans le bas du dos. Différents algorithmes de segmentation, dont l'algorithme de GHARSI [56], l'algorithme de SHIN [53], l'algorithme de CWT [51] et l'algorithme de Gonzales [54] ont été employés pour extraire les caractéristiques spatio-temporelles et statistiques. Bien que l'algorithme GHARSI se soit révélé être le plus précis, nous avons finalement opté pour l'algorithme Gonzales, compte tenu de son compromis optimal entre rapidité et précision, facteur essentiel pour l'implémentation pratique de la solution.

Les différents algorithmes de classification, notamment le KNN et le SVM, ont ensuite été utilisés, avec des résultats prometteurs. Le classificateur KNN a obtenu une exactitude de 89.7% et un F1-score de 90%, tandis que le SVM a atteint une exactitude de 87.6% et un F1-score de 90%.

Ces résultats démontrent l'efficacité et la faisabilité de notre approche. Cela suggère des perspectives prometteuses pour des applications futures dans le domaine de la santé publique, en particulier dans la prévention des chutes chez les personnes âgées. Notre travail pourrait donc servir de base à des interventions efficaces et opportunes, améliorant ainsi la qualité de vie des personnes âgées.

## Perspective

les résultats obtenus ouvrent la voie à une multitude d'opportunités pour l'avenir. Les modèles SVM, KNN et DNN ayant démontré leur efficacité dans la détection des chutes, l'enjeu majeur demeure leur intégration en temps réel. Cela passe par l'exploitation de technologies comme le TinyML qui permet d'exécuter des modèles d'apprentissage automatique sur des dispositifs à ressources de calcul limitées. Cette approche pourrait avoir une valeur ajoutée significative, notamment pour les

systèmes embarqués dédiés à la détection des chutes. Par ailleurs, ces techniques pourraient être appliquées à d'autres groupes à haut risque de chute, impliquant l'utilisation de nouvelles bases de données et l'adaptation de nos approches à ces contextes spécifiques. Au-delà de la détection, une autre perspective intéressante serait la prédiction du risque de chute à partir de l'analyse de la marche, permettant des interventions préventives. Malgré les avancées significatives réalisées jusqu'ici, ces perspectives montrent qu'il reste encore un vaste champ d'exploration pour maximiser le potentiel de l'apprentissage automatique dans la détection et la prévention des chutes.

# Bibliographie

- [1] Chute. [bit.ly/3Ne8TDU](https://bit.ly/3Ne8TDU).
- [2] Jacquelin Perry, Jon R Davids, et al. Gait analysis : normal and pathological function. *Journal of Pediatric Orthopaedics*, 12(6) :815, 1992.
- [3] Chandra Prakash, Rajesh Kumar, and Namita Mittal. Recent developments in human gait research : parameters, approaches, applications, machine learning techniques, datasets and challenges. *Artificial Intelligence Review*, 49 :1–40, 2018.
- [4] MG Benedetti, A Merlo, and A Leardini. Inter-laboratory consistency of gait analysis measurements. *Gait & posture*, 38(4) :934–939, 2013.
- [5] Alice Nieuwboer, René Dom, Willy De Weerdt, Kaat Desloovere, Steffen Fieuws, and Eva Broens-Kaucsik. Abnormalities of the spatiotemporal characteristics of gait at the onset of freezing in parkinson’s disease. *Movement disorders : official journal of the Movement Disorder Society*, 16(6) :1066–1075, 2001.
- [6] Mary Roberts, David Mongeon, and Francois Prince. Biomechanical parameters for gait analysis : a systematic review of healthy human gait. *Phys Ther Rehabil*, 4(1) :6, 2017.
- [7] Muqing Deng, Cong Wang, Fengjiang Cheng, and Wei Zeng. Fusion of spatial-temporal and kinematic features for gait recognition with deterministic learning. *Pattern Recognition*, 67 :186–200, 2017.
- [8] Meg E Morris, Jennifer McGinley, Frances Huxham, Janice Collier, and Robert Ianseck. Constraints on the kinetic, kinematic and spatiotemporal parameters of gait in parkinson’s disease. *Human Movement Science*, 18(2-3) :461–483, 1999.
- [9] Nissan Kunju, Neelesh Kumar, Dinesh Pankaj, Aseem Dhawan, and Amod Kumar. Emg signal analysis for identifying walking patterns of normal healthy individuals. *Indian journal of biomechanics*, 118, 2009.

- [10] Faisal Hussain, Fawad Hussain, Muhammad Ehatisham ul Haq, and Muhammad Awais Azam. Activity-aware fall detection and recognition based on wearable sensors. *IEEE Sensors Journal*, 19 :4528–4536, 2019.
- [11] Ahmad Jalal, Majid Ali Khan Quaid, and Kibum Kim. A wrist worn acceleration based human motion analysis and classification for ambient smart home system. *Journal of Electrical Engineering & Technology*, 14 :1733–1739, 2019.
- [12] Jeong-Kyun Kim, Myung-Nam Bae, Kangbok Lee, Jae-Chul Kim, and Sang Gi Hong. Explainable artificial intelligence and wearable sensor-based gait analysis to identify patients with osteopenia and sarcopenia in daily life. *Biosensors*, 12(3) :167, 2022.
- [13] David H Sutherland. The evolution of clinical gait analysis : Part ii kinematics. *Gait & posture*, 16(2) :159–179, 2002.
- [14] Ravi Kant Avvari, Mirza Khalid Baig, Thirugnanam Arunachalam, et al. Gait analysis : An effective tool to measure human performance. In *Advances in Computational Approaches in Biomechanics*, pages 65–87. IGI Global, 2022.
- [15] WI Schöllhorn, JM Jäger, and D Janssen. Artificial neural network models of sports motions. In *Routledge handbook of biomechanics and human movement science*, pages 50–64. Routledge, 2008.
- [16] Alex Smola and SVN Vishwanathan. Introduction to machine learning. *Cambridge University, UK*, 32(34) :2008, 2008.
- [17] Eduardo Casilari, José-Antonio Santoyo-Ramón, and José-Manuel Cano-García. Analysis of public datasets for wearable fall detection systems. *Sensors*, 17(7) :1513, 2017.
- [18] Anuradha Singh, Saeed Ur Rehman, Sira Yongchareon, and Peter Han Joo Chong. Sensor technologies for fall detection systems : A review. *IEEE Sensors Journal*, 20(13) :6889–6919, 2020.
- [19] Anita Ramachandran and Anupama Karuppiyah. A survey on recent advances in wearable fall detection systems. *BioMed research international*, 2020, 2020.
- [20] Sérgio Barroso Gonçalves, Siddhi Bianca Camila Lama, and Miguel Tavares da Silva. Three decades of gait index development : A comparative review of clinical and research gait indices. *Clinical Biomechanics*, 96 :105682, 2022.
- [21] Eduardo Casilari, José-Antonio Santoyo-Ramón, and José-Manuel Cano-García. Analysis of public datasets for wearable fall detection systems. *Sensors*, 17(7), 2017.

- [22] Eduardo Casilari, Jose A. Santoyo-Ramón, and Jose M. Cano-García. Umafall : A multisensor dataset for the research on automatic fall detection. *Procedia Computer Science*, 110 :32–39, 2017. 14th International Conference on Mobile Systems and Pervasive Computing (MobiSPC 2017) / 12th International Conference on Future Networks and Communications (FNC 2017) / Affiliated Workshops.
- [23] George Vavoulas, Matthew Padiaditis, Emmanouil Spanakis, and Manolis Tsiknakis. The mobifall dataset : An initial evaluation of fall detection algorithms using smartphones. pages 1–4, 11 2013.
- [24] Angela Sucerquia, José David López, and Jesús Francisco Vargas-Bonilla. Sisfall : A fall and movement dataset. *Sensors*, 17(1), 2017.
- [25] Aner Weiss, Marina Brozgol, Moran Dorfman, Talia Herman, Shirley Shema, Nir Giladi, and Jeffrey M Hausdorff. Does the evaluation of gait quality during daily life provide insight into fall risk? a novel approach using 3-day accelerometer recordings. *Neurorehabilitation and neural repair*, 27(8) :742–752, 2013.
- [26] Marta Neira Álvarez, Antonio R Jiménez Ruiz, Guillermo García-Villamil Neira, Elisabet Huertas-Hoyas, Maria Teresa Espinoza Cerda, Laura Pérez Delgado, Elena Reina Robles, Antonio J Del-Ama, Luisa Ruiz-Ruiz, Sara García-de Villa, et al. Assessing falls in the elderly population using g-stride foot-mounted inertial sensor. *Scientific Reports*, 13(1) :9208, 2023.
- [27] Giulia Pacini Panebianco, Maria Cristina Bisi, Rita Stagni, and Silvia Fantozzi. Analysis of the performance of 17 algorithms from a systematic review : Influence of sensor position, analysed variable and computational approach in gait timing estimation from imu measurements. *Gait & posture*, 66 :76–82, 2018.
- [28] Abdul Saboor, Triin Kask, Alar Kuusik, Muhammad Mahtab Alam, Yannick Le Moullec, Imran Khan Niazi, Ahmed Zoha, and Rizwan Ahmad. Latest research trends in gait analysis using wearable sensors and machine learning : A systematic review. *Ieee Access*, 8 :167830–167864, 2020.
- [29] Can Tunca, Gulustu Salur, and Cem Ersoy. Deep learning for fall risk assessment with inertial sensors : Utilizing domain knowledge in spatio-temporal gait parameters. *IEEE journal of biomedical and health informatics*, 24, 12 2019.
- [30] Benoit Caby, Suzanne Kieffer, Marie de Saint Hubert, Gerald Cremer, and Benoit Macq. Feature extraction and selection for objective gait analysis and



- fall risk assessment by accelerometry. *Biomedical engineering online*, 10(1) :1–19, 2011.
- [31] Michael Marschollek, A Rehwald, KH Wolf, M Gietzelt, G Nemitz, H Meyer Zu Schwabedissen, and R Haux. Sensor-based fall risk assessment—an expert ‘to go’. *Methods of information in medicine*, 50(05) :420–426, 2011.
- [32] Ying Liu, Stephen J Redmond, Michael R Narayanan, and Nigel H Lovell. Classification between non-multiple fallers and multiple fallers using a triaxial accelerometry-based system. In *2011 annual international conference of the IEEE engineering in medicine and biology society*, pages 1499–1502. IEEE, 2011.
- [33] Matthias Gietzelt, Florian Feldwieser, Mehmet Gövercin, Elisabeth Steinhagen-Thiessen, and Michael Marschollek. A prospective field study for sensor-based identification of fall risk in older people with dementia. *Informatics for health and social care*, 39(3-4) :249–261, 2014.
- [34] Jianning Wu, Bin Wu, et al. The novel quantitative technique for assessment of gait symmetry using advanced statistical learning algorithm. *BioMed research international*, 2015, 2015.
- [35] Yuqing Chen and Yang Xue. A deep learning approach to human activity recognition based on single accelerometer. In *2015 IEEE international conference on systems, man, and cybernetics*, pages 1488–1492. IEEE, 2015.
- [36] Tahmina Zebin, Patricia J Scully, and Krikor B Ozanyan. Human activity recognition with inertial sensors using a deep learning approach. In *2016 IEEE sensors*, pages 1–3. IEEE, 2016.
- [37] Yuwen Chen, Kunhua Zhong, Ju Zhang, Qilong Sun, and Xueliang Zhao. Lstm networks for mobile human activity recognition. In *2016 International conference on artificial intelligence : technologies and applications*, pages 50–53. Atlantis Press, 2016.
- [38] Tianyi Zhen, Lilei Mao, Jiawei Wang, and Qiang Gao. Wearable preimpact fall detector using svm. In *2016 10th International Conference on Sensing Technology (ICST)*, pages 1–6. IEEE, 2016.
- [39] Changzhan Gu. Short-range noncontact sensors for healthcare and other emerging applications : A review. *Sensors*, 16(8) :1169, 2016.
- [40] Yongjia Zhao and Suiping Zhou. Wearable device-based gait recognition using angle embedded gait dynamic images and a convolutional neural network. *Sensors*, 17(3) :478, 2017.

- [41] Wei-Yi Cheng, Alf Scotland, Florian Lipsmeier, Timothy Kilchenmann, Liping Jin, Jens Schjodt-Eriksen, Detlef Wolf, Yan-Ping Zhang-Schaerer, Ignacio Fernandez Garcia, Juliane Siebourg-Polster, et al. Human activity recognition from sensor-based large-scale continuous monitoring of parkinson's disease patients. In *2017 IEEE/ACM International Conference on Connected Health : Applications, Systems and Engineering Technologies (CHASE)*, pages 249–250. IEEE, 2017.
- [42] Eftim Zdravevski, Biljana Risteska Stojkoska, Marie Standl, and Holger Schulz. Automatic machine-learning based identification of jogging periods from accelerometer measurements of adolescents under field conditions. *PLoS one*, 12(9) :e0184216, 2017.
- [43] Yi Xia, Jun Zhang, Qiang Ye, Nan Cheng, Yixiang Lu, and Dexiang Zhang. Evaluation of deep convolutional neural networks for detection of freezing of gait in parkinson's disease patients. *Biomedical Signal Processing and Control*, 46 :221–230, 2018.
- [44] Enas Abdulhay, N Arunkumar, Kumaravelu Narasimhan, Elamaran Vellaiappan, and V Venkatraman. Gait and tremor investigation using machine learning techniques for the diagnosis of parkinson disease. *Future Generation Computer Systems*, 83 :366–373, 2018.
- [45] Chia-Yeh Hsieh, Wan-Ting Shi, Hsiang-Yun Huang, Kai-Chun Liu, Steen J Hsu, and Chia-Tai Chan. Machine learning-based fall characteristics monitoring system for strategic plan of falls prevention. In *2018 IEEE International Conference on Applied System Invention (ICASI)*, pages 818–821. IEEE, 2018.
- [46] Hamza Abujrida, Emmanuel Agu, and Kaveh Pahlavan. Machine learning-based motor assessment of parkinson's disease using postural sway, gait and lifestyle features on crowdsourced smartphone data. *Biomedical Physics & Engineering Express*, 6(3) :035005, 2020.
- [47] Han Byul Kim, Hong Ji Lee, Woong Woo Lee, Sang Kyong Kim, Hyo Seon Jeon, Hye Young Park, Chae Won Shin, Won Jin Yi, Beomseok Jeon, and Kwang S Park. Validation of freezing-of-gait monitoring using smartphone. *Telemedicine and e-Health*, 24(11) :899–907, 2018.
- [48] Reed D Gurchiek, Rebecca H Choquette, Bruce D Beynnon, James R Slauterbeck, Timothy W Tourville, Michael J Toth, and Ryan S McGinnis. Remote gait analysis using wearable sensors detects asymmetric gait patterns in patients recovering from acl reconstruction. In *2019 IEEE 16th International*

- Conference on Wearable and Implantable Body Sensor Networks (BSN)*, pages 1–4. IEEE, 2019.
- [49] Chao Wang, Peter PK Chan, Ben MF Lam, Sizhong Wang, Janet H Zhang, Zoe YS Chan, Rosa HM Chan, Kevin KW Ho, and Roy TH Cheung. Real-time estimation of knee adduction moment for gait retraining in patients with knee osteoarthritis. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 28(4) :888–894, 2020.
- [50] Pablo Aqueveque, Britam Gomez, Paulina Ortega-Bastidas, Guisella Pena, Gustavo Retamal, and Roberto Cano-de-la Cuerda. Predicting risk of falls in elderly using a single inertial measurement unit on the lower-back by estimating spatio-temporal gait parameters. In *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 2390–2394. IEEE, 2022.
- [51] Rodrigo Capobianco Guido, Fernando Pedroso, André Furlan, Rodrigo Colnago Contreras, Luiz Gustavo Caobianco, and Jogi Suda Neto.  $Cwt \times dwt \times dtwt \times sdtwt$  : Clarifying terminologies and roles of different types of wavelet transforms. *International Journal of Wavelets, Multiresolution and Information Processing*, 18(06) :2030001, 2020.
- [52] Minh H Pham, Morad Elshehabi, Linda Haertner, Silvia Del Din, Karin Srujijes, Tanja Heger, Matthis Synofzik, Markus A Hobert, Gert S Faber, Clint Hansen, et al. Validation of a step detection algorithm during straight walking and turning in patients with parkinson’s disease and older adults using an inertial measurement unit at the lower back. *Frontiers in neurology*, 8 :457, 2017.
- [53] Seung Hyuck Shin and Chan Gook Park. Adaptive step length estimation algorithm using optimal parameters and movement status awareness. *Medical engineering & physics*, 33(9) :1064–1071, 2011.
- [54] Rafael C Gonzalez, Diego Alvarez, Antonio M Lopez, and Juan C Alvarez. Modified pendulum model for mean step length estimation. In *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 1371–1374. IEEE, 2007.
- [55] Meinard Müller. Dynamic time warping. *Information retrieval for music and motion*, pages 69–84, 2007.
- [56] Ignacio Ghersi, Maria H Ferrando, Carlos G Fliger, Cristhian F Castro Arenas, Diego J Edwards Molina, and Mónica T Miralles. Gait-cycle segmentation

- method based on lower-trunk acceleration signals and dynamic time warping. *Medical Engineering & Physics*, 82 :70–77, 2020.
- [57] JC Alvarez, AM Lo, J Rodriguez-uri, A Diego, and RC Gonza. Real-time gait event detection for normal subjects from the lower trunk. *Gait and Posture.*, 31 :322–325, 2010.
- [58] Ngoc-Huynh Ho, Phuc Huu Truong, and Gu-Min Jeong. Step-detection and adaptive step-length estimation for pedestrian dead-reckoning at various walking speeds using a smartphone. *Sensors*, 16(9) :1423, 2016.
- [59] Aleksandr Mikov, Alex Moschevikin, Alexander Fedorov, and Axel Sikora. A localization system using inertial measurement units from wireless commercial hand-held devices. In *International Conference on Indoor Positioning and Indoor Navigation*, pages 1–7. IEEE, 2013.
- [60] Inge Bylemans, Maarten Weyn, and Martin Klepal. Mobile phone-based displacement estimation for opportunistic localisation systems. In *2009 Third International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, pages 113–118. IEEE, 2009.
- [61] Ian Sharp and Kegen Yu. Sensor-based dead-reckoning for indoor positioning. *Physical Communication*, 13 :4–16, 2014.
- [62] William S Noble. What is a support vector machine? *Nature biotechnology*, 24(12) :1565–1567, 2006.
- [63] Carl Kingsford and Steven L Salzberg. What are decision trees? *Nature biotechnology*, 26(9) :1011–1013, 2008.
- [64] Decision tree. <https://bit.ly/3qEH4gb>.
- [65] Ensemble learning. <https://bit.ly/43QoyQL>.
- [66] Jason Brownlee. *Ensemble learning algorithms with Python : Make better predictions with bagging, boosting, and stacking*. Machine Learning Mastery, 2021.
- [67] Aakash Parmar, Rakesh Katariya, and Vatsal Patel. A review on random forest : An ensemble classifier. In *International Conference on Intelligent Data Communication Technologies and Internet of Things (ICICI) 2018*, pages 758–763. Springer, 2019.
- [68] Reseau de neurones. <bit.ly/3pcW4BJ>.
- [69] Miguel A Carreira-Perpinán. A review of dimension reduction techniques. *Department of Computer Science. University of Sheffield. Tech. Rep. CS-96-09*, 9 :1–69, 1997.

- [70] statpca. [bit.ly/3PidTK4](https://bit.ly/3PidTK4).
- [71] Pca. [bit.ly/468pCkA](https://bit.ly/468pCkA).
- [72] Pablo Aqueveque, Britam Gomez, Paulina Ortega-Bastidas, Guisella Pena, Gustavo Retamal, and Roberto Cano-de-la Cuerda. Predicting risk of falls in elderly using a single inertial measurement unit on the lower-back by estimating spatio-temporal gait parameters. In *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 2390–2394. IEEE, 2022.