

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université des Sciences et de la Technologie Houari Boumediene



Faculté d'informatique
Département de systèmes informatiques

MEMOIRE

en vue de l'obtention du

Diplôme de Master en :
Informatique

Option :
Réseaux et systèmes distribués

Conception d'un système de détection et d'alerte de mauvaise posture assise à l'aide des capteurs d'Electromyographie basé sur l'intelligence artificielle.

Présenté par : OUANDJLI Lyna & KESSAISSIA Meriem

Soutenu le 20/09/2022, devant la commission d'examen :

M. LAICHI Boualem	Président
Mme. ZEBBANE Bahia	Examinateuse
Mme. Roufaida Laidi	Encadrante

Code mémoire :RSD-E-018



DEDICATION

Avec l'expression de ma reconnaissance, je dédie ce modeste travail à ceux qui, quels que soient les termes embrassés, je n'arriverais jamais à leur exprimer mon amour sincère.

A la femme qui n'a jamais dit non âmes exigences et qui n'a épargné aucun effort pour me rendre heureuse, qui m'a comblé d'amour et d'affection et qui n'a jamais cessé de prier pour moi : mon adorable **mère**.

A l'homme, mon précieux offre du dieu, qui doit ma vie, ma réussite et tout mon respect, pour ses encouragements, surtout son amour et son sacrifice afin que rien n'entrave le déroulement de mes études : mon cher **père**.

A mes chères sœurs **Fatima Zahra, Wafaa** et mon cher frère **Wassim** qui n'ont pas cessée de me conseiller, encourager et soutenir tout au long de mes études. Que Dieu les protège et leurs offre la chance et le bonheur.

A ma meilleure amie, **Aridj** qui m'a toujours encouragé, supporté dans les moments difficiles, et à qui je souhaite plus de succès.

Sans oublier mon binôme **Lyna** pour son soutien moral, sa patience et sa compréhension tout au long de ce projet.

KESSAISSIA Meriem



DEDICATION

Je dédie ce travail à :

A ma maman et mon papa avec tous mes sentiments de respect, d'amour, de gratitude et de reconnaissance pour tous les sacrifices déployés pour m' élever dignement et assurer mon éducation dans les meilleures conditions, Puisse le tout puissant leur donne santé, bonheur et longue vie.

A ma sœur Imene et mon frère Amine qui ont partagé avec moi tous les moments d'émotion lors de la réalisation de ce travail.

A mon fiancé Samir ma source de motivation qui m'a énormément soutenu dans les moments les plus difficiles, partagé mes joies et mes peines.

A mon unique binôme Meriem qui m'a épaulé et m'a encouragé tout au long de ma formation.
A toute personne qui a contribué à la réalisation de ce modeste travail.

OUANDJLI Lyna

REMERCIEMENT

Au début, Merci à dieu le tout puissant de nous avoir guidé et donné la force et le courage de tenir jusqu'à la fin de ce travail.

On souhaite adresser nos remerciements les plus sincères aux personnes qui nous ont apporté leur aide et qui ont contribué à l'élaboration de ce mémoire.

On tient à remercier tout particulièrement notre encadrant madame LAIDI ROUFAIDA pour nous avoir suivis et conseillés tout au long de la réalisation de ce mémoire.

On tient à remercier, Madame ZEBBANE BAHIA pour la qualité des renseignements qu'elle nous a offerts.

un grand merci a tous les enseignants de l'USTHB pour leurs efforts afin que notre foemation se termine dans de bonnes conditions.

Ce mémoire n'aurait jamais pu voir le jour sans le soutien actif des membres de nos famille,nos amis, surtout nos parents qu'ils nous ont toujours encouragé moralement et matériellement et à qui on tient a les remercier.

Enfin on tient à exprimer vivement nos remerciements avec une profonde gratitude à toutes les personnes qui ont contribué de près ou de loin à sa realisation, car ce projet ne peut pas être le fruit d'une seule personne.

■ TABLE DES MATIÈRES

LISTE DES FIGURES	x
LISTE DES ABRÉVIATIONS	xi
INTRODUCTION GÉNÉRALE	xiii
1 Généralités	2
1.1 INTRODUCTION	3
1.2 Les troubles musculosquelettiques	3
1.3 Machine Learning	4
1.3.1 Définition	4
1.3.2 Fonctionnement	4
1.3.3 Algorithmes d'apprentissage automatique	5
1.3.3.1 Régression et Classification	6
1.4 Deep Learning	9
1.5 Bluetooth à énergie réduite (Bluetooth Low Energy)	11
1.6 Communication UART (Universal Asynchronous Receiver Transmitter)	15
1.7 Service UART nordique (NUS)	15
1.8 CONCLUSION	16
2 Collecte des données sur la mauvaise posture	17
2.1 INTRODUCTION	18
2.2 Les images en temps réel.	18
2.3 Les informations de poids	18
2.4 Les informations sur l'angle d'inclinaison.	21
2.5 L'information de la courbure de la colonne vertébrale	24
2.6 Information fournie par les capteurs EMG	27
2.7 CONCLUSION	31

TABLE DES MATIÈRES

3 Détection de mauvaise posture par Machine Learning	32
3.1 INTRODUCTION	33
3.2 Application d'approches de machine Learning pour classer la posture assise.	33
3.3 Application d'approches Deep Learning pour classer la posture assise	39
3.4 CONCLUSION	42
4 Conception générale du système	43
4.1 INTRODUCTION	44
4.2 Architecture du système	45
4.3 Collecte de données à l'aide des capteurs EMG	46
4.3.1 Choix du capteur EMG	46
4.3.1.1 Discription de quelques capteurs EMG	46
4.3.1.2 Comparaison des capteurs EMG	51
4.4 Traitement des données avec microcontrôleur	51
4.4.1 choix du microcontrôleur	52
4.4.1.1 Exemples de microcontrôleurs envisagés avec description . .	52
4.4.1.2 Comparaison des microcontrôleurs	54
4.4.1.3 Résultat de la comparaison	55
4.4.2 Convertisseur analogique-numérique à approximations successives SAADC	55
4.5 Classification de la posture assise	56
4.6 Application mobile	57
4.7 CONCLUSION	58
5 Implementation et résultats	59
5.1 INTRODUCTION	60
5.2 Architecture du système	60
5.3 Collecte de données	61
5.3.1 Test du capteur EMG	61
5.3.2 Emplacement des capteurs EMG	62
5.3.3 Assemblage du circuit	63
5.4 Implémentation du périphérique BLE	64
5.5 Classification de la posture	66
5.5.1 Données d'entraînement	66
5.5.2 Prétraitement des données	66
5.5.3 Outils de programmation	68

TABLE DES MATIÈRES

5.5.4	Classification binaire	69
5.5.5	Classification Multiclasse	77
5.6	Application mobile	83
5.6.1	Implémentation du central BLE	85
5.6.2	Interface utilisateur	86
5.6.2.1	La page d'accueil	87
5.6.2.2	Page de connexion BLE-UART	89
5.6.2.3	Page de conseils et d'exercices	94
5.7	Analyse des résultats	94
5.8	CONCLUSION	95
CONCLUSION GÉNÉRALE		96
BIBLIOGRAPHIE		97

LISTE DES FIGURES

1.1	Représentation d'un neurone formel	10
1.2	Tableau : Exemples de fonctions d'activation[71]	11
1.3	Architecture de Bluetooth Low Energy[67]	12
1.4	Les objets basiques d'une transaction GATT dans BLE [66]	14
1.5	communication UART	15
2.1	Types de postures assises [21]	19
2.2	La proposition de concept de système. [33]	20
2.3	Les positions assises définies par cette étude. [33]	21
2.4	Calculs d'angle. (a) Mouvement de rémunération. (b) Degré zéro Étalonnage. (c) Modèle d'étalonnage [25].	22
2.5	Le client dans une bonne et mauvaise posture.	24
2.6	Principe d'un système de surveillance de la posture à base d'inducteurs [1] . . .	26
2.7	Emplacements de mesure sEMG dans le dos[51]	28
2.8	Emplacement des marqueurs[48]	29
3.1	Tableau :Taux de classification des données d'essai selon le classificateur dans chaque sujet [36]	35
3.2	Un coussin intelligent entièrement intégré pour le devant de la chaise[37] . . .	36
3.3	Collecte de données d'entraînement avec différentes positions assises.[41] . . .	38
3.4	Graphe de précision pour les algorithmes d'apprentissage automatique.[41] . .	38
3.5	vue conceptuelle de CVA et FSA.[44]	41
4.1	Architecture de la solution proposé	45
4.2	Kit de développement de capteur musculaire MyoWare-14409 ROHS [52] . . .	47
4.3	Capteur musculaire électrique Sparkfun MyoWare - SEN-13723 (réf SSBIMU4)	48
4.4	Capteur EMG SEN0240	49
4.5	EMG Grove Detector 101020058	50
4.6	Capteur de surface EMG DataLITE	50
4.7	Tableau comparatif des capteurs EMG [52][53][54][55]	51

LISTE DES FIGURES

4.8	Carte Arduino [60]	52
4.9	Le classement des séries STM selon leurs groupes STM32 [63]	53
4.10	carte nRF52	54
4.11	Tableau comparatif des microcontrôleurs[60] [61] [58] [62]	54
4.12	Diagramme d'activité : étapes de classification de la posture	56
4.13	Diagramme de cas d'utilisation de l'application	57
5.1	implémentation du système	60
5.2	Test du capteur : mode repos, début d'une contraction, contraction complète	61
5.3	visualisation du signal numérique : mode repos, début d'une contraction, contraction complète	62
5.4	Emplacement du capteur musculaire sur le dos	63
5.5	Assemblage du circuit	63
5.6	Diagramme d'activité : périphérique BLE	65
5.7	graphe : visualisation des données en cas de bonne posture	66
5.8	graphe : visualisation des données en cas de mauvaise posture	67
5.9	visualisation du dataset	68
5.10	Architecture du réseau de neurones	70
5.11	Multilayer Perceptron :accuracy	73
5.12	Multilayer Perceptron :loss	74
5.13	Support Vector machine :accuracy	74
5.14	k-Nearest Neighbor :accuracy	75
5.15	Decision Tree :accuracy	76
5.16	Random Forest :accuracy	76
5.17	accuracy comparison	77
5.18	Architecture du perceptron multicouche dans le cas d'une classification multiclesse	78
5.19	Multilayer perceptron model on multiclass classification :accuracy	79
5.20	Multilayer perceptron model on multiclass classification :accuracy	80
5.21	Support Vector machine :accuracy	81
5.22	k-Nearest Neighbor :accuracy	81
5.23	Decision Tree :accuracy	82
5.24	Random Forest :accuracy	82
5.25	accuracy comparison	83
5.26	Diagramme d'activités du central BLE.	85
5.27	Page d'accueil	87
5.28	Page d'accueil (bouton Start)	88

LISTE DES FIGURES

5.29	Page de connexion BLE-UART	89
5.30	Page de connexion BLE-UART(scan et connexion avec des périphériques) . . .	90
5.31	Page de connexion BLE-UART (cas d'erreur)	91
5.32	Page de connexion BLE-UART(insertion de l'age et le genre)	92
5.33	Page de connexion BLE-UART(Notifications)	93
5.34	Page de conseils et d'exercices	94



LISTE DES ABRÉVIATIONS

ATT Attribute Protocol

BLE Bleutooth Low Energy

CART Classification And Regression Tree

CNN Convolutional Neural Network

CVA Cranio-Vertebral Angle

DT Decision Tree

EMG Electromyographie

FHP Front Head Posture

HCI Host-Controller Interface

IA Intelligence Artificielle

IEMG Integrated EMG

IOS Iphone Operating System

IDE Integrated Devlopement Environement

KNN K-Nearest Neighbor

LightGBM Light gradient boosting machine

LBP Local Binary Pattern

MQTT Message Queuing Telemetry Transport

MLR Multinomial Logistic Regression

MLP Multi-layer Perceptron

NUS Nordic UART Service

LISTE DES ABRÉVIATIONS

OSI Open Systems Interconnection

PMC perceptron multicouches

PSD Power Spectral Density

PPI Programmable Interferal Connection

QNAP Quality Network Appliance Provider

RF Random Forest

RFEID Radio frequency identificatory

RS Rounded Shoulder

RX Receiver caracteristic

SVM Support Vector Machines

Soc System of chip

SAADC Successive Approximation Analog to Digital Converter

SGDC Stochastic Gradient Descent Classifier

GATT Generic Attribute Profile

GAP Generic Access Profile

TMS Les troubles musculosquelettiques

TX Transmitter caracteristic

USB Universal Serial Bus

UUID Universally Unique Identifier

UART universal asynchronous receiver / transmitter

VDD Voltage Drain Drain

INTRODUCTION GÉNÉRALE

Dans ce monde en évolution rapide, la santé et le bien-être des employés est un sujet réaliste. Les employés de bureau auront inévitablement de mauvaises postures s'ils restent longtemps assis devant l'ordinateur. Une mauvaise posture persistante de la colonne vertébrale est associée au développement et à l'exacerbation de maladies musculosquelettiques, entraînant des douleurs au niveau du cou et du dos, une radiculopathie et des déficits sensorimoteurs. En cas de symptômes sévères, une intervention chirurgicale devient nécessaire. En raison de la hausse des coûts des soins de santé et de la baisse de la productivité du travail, une mauvaise posture finira par entraîner un fardeau économique important. Bien qu'il soit difficile d'estimer le coût total d'une mauvaise posture à lui seul, aux États-Unis seulement, le fardeau prévu des maux de dos est un coût net de 60 milliards de dollars par an [1].

Par conséquent, il est nécessaire de surveiller la posture en temps réel et la corriger. La surveillance de la posture peut prévenir les conséquences ci-dessus grâce à un retour d'information en temps réel, ce qui favorise la correction des mauvaises postures continues et réduit leur durée totale. On suppose que ces systèmes peuvent inculquer des habitudes posturales correctes et réduire l'incidence des maladies musculosquelettiques posturales qui provoquent des maladies musculosquelettiques avec une utilisation à long terme. Traditionnellement, la posture de la colonne vertébrale est évaluée en milieu clinique lors d'examens de routine des patients, de cours de physiothérapie ou d'évaluations formelles en laboratoire. Ces méthodes ont été critiquées comme coûteuses et peu pratiques, gravement incapables de mesurer la posture quotidienne et de fournir une rétroaction en temps opportun. Les solutions basées sur les capteurs peuvent évaluer ces problèmes en proposant des alternatives bon marché et pratiques. De nombreux systèmes de surveillance des gestes dans

INTRODUCTION GÉNÉRALE

la littérature sont basés sur des caméras et le traitement d'images, ce qui compromet la confidentialité. Récemment, la technologie de détection a été continuellement développée et diffusée dans différents domaines, en particulier dans les systèmes médicaux et de santé. Ces capteurs se caractérisent par leur portabilité, leur faible coût, leur facilité d'utilisation, leur petite taille. Dans ce travail, nous nous intéressons particulièrement aux capteurs d'électromyographie (EMG) [2]. Ce projet consiste en la proposition et la mise en place d'un système de détection de mauvaise posture assise. Le système se compose de trois éléments principaux. Le premier composant comprend les capteurs EMG qui collectent des informations représentant la contraction musculaire. Le deuxième analyse les données et détermine la mauvaise posture et son en se basant sur un algorithme d'apprentissage automatique choisi. Le troisième élément est un système de rétroaction, une application mobile, pour alerter l'utilisateur dans cas de mauvaise posture.

Ce document se compose de cinq chapitres, le premier sera consacré aux généralités où nous allons définir quelques notions essentiels pour la compréhension de la suite du projet.

Le deuxième chapitre sera focalisé autours de la collecte de données sur la mauvaise posture où des systèmes existants seront présentés.

Le troisième chapitre quant à lui va exposer les études précédentes qui ont exploré l'application des algorithmes de Machine Learning et du Deep Learning dans la reconnaissance de la posture assise.

L'une des parties les plus importantes de ce mémoire est bien sur la conception de notre solution qui sera traitée dans le quatrième chapitre. Il détaillera entre autres l'architecture de notre système, la définition, les fonctionnalités et la justification du choix de chaque composant, les étapes à suivre pour la classification de la posture et enfin nous concluons la partie avec une description des besoins techniques et fonctionnels de l'application mobile.

Le dernier chapitre sera consacré à la description de manière détaillé le système réalisé en détaillant toutes les configurations et les résultats obtenus dans chaque sous-système.

Enfin nous terminerons par une conclusion générale accompagné d'une liste de perspectives d'avenir pour ce projet.

Généralités

Sommaire

1.1	INTRODUCTION	3
1.2	Les troubles musculosquelettiques	3
1.3	Machine Learning	4
1.3.1	Définition	4
1.3.2	Fonctionnement	4
1.3.3	Algorithmes d'apprentissage automatique	5
1.4	Deep Learning	9
1.5	Bluetooth à énergie réduite (Bluetooth Low Energy)	11
1.6	Communication UART (Universal Asynchronous Receiver Transmitter)	15
1.7	Service UART nordique (NUS)	15
1.8	CONCLUSION	16

1.1 INTRODUCTION

Tout au long de ce chapitre nous tenterons, d'une part, d'aborder brièvement la notion des troubles musculosquelettiques, d'autre part nous définissons le machine Learning en détaillerons notamment son fonctionnement et le Deep Learning en citant quelques algorithmes de ces derniers, de plus nous allons présenter des protocoles de communication à savoir le bluetooth à énergie réduite et un service particulier de ce dernier (service UART nordique) et le protocole UART, ceci va aider à mieux comprendre la suite de notre projet.

1.2 Les troubles musculosquelettiques

Les troubles musculosquelettiques (TMS) sont des lésions qui peuvent apparaître lorsqu'on adopte de mauvaises postures, notamment dans le milieu professionnel. L'augmentation du risque d'affection de l'appareil musculosquelettique est due à l'incapacité du corps à s'adapter aux conditions de travail contraignantes avec l'insuffisance du temps alloué au repos.

Selon la définition du ministère du travail américain, les troubles musculosquelettiques sont des blessures ou des maladies pouvant affecter les muscles, les nerfs, les tendons, les articulations, le cartilage et les disques intervertébraux au niveau de toutes les parties du corps. Ils s'expriment généralement par la douleur, une faiblesse musculaire ou une limitation des articulations et sont principalement causées par les sur-sollicitations des structures musculosquelettiques lors des postures contraintes prolongées et/ou de mouvements répétitifs .[49]

Les mauvaises postures causent généralement des troubles et des problèmes de santé au niveau de la colonne vertébrale. Ces derniers sont omniprésents dans le monde. Il existe trois types courants de troubles de la courbure de la colonne vertébrale : la scoliose, la cyphose et la lordose. La scoliose provoque une courbure latérale de la colonne vertébrale en forme de «S» ou de «C». Il survient souvent pendant la croissance d'un enfant. La courbure varie de

légère à sévère et peut s'aggraver avec l'âge. La cyphose est une condition qui fait que la partie supérieure de la colonne vertébrale se courbe vers l'avant, créant une courbure du dos voûtée à un angle supérieur à 50 degrés. La lordose fait référence à une grande courbure vers l'intérieur de la colonne vertébrale dans le bas du dos.

Pour éviter les TMS, il faut être vigilant quant à la posture qu'on adopte, sa répétition et sa durée.

1.3 Machine Learning

1.3.1 Définition

Le Machine Learning ou apprentissage automatique est un domaine scientifique, et plus particulièrement une sous-catégorie de l'intelligence artificielle, il permet aux ordinateurs d'apprendre sans avoir été programmés explicitement à cet effet. Pour apprendre et se développer, les ordinateurs ont toutefois besoin de données à analyser et sur lesquelles s'entraîner.[72]

1.3.2 Fonctionnement

Le développement d'un modèle de Machine Learning repose sur quatre étapes principales. La première étape consiste à sélectionner et à préparer un ensemble de données d'entraînement. Ces données seront utilisées pour nourrir le modèle de Machine Learning pour apprendre à résoudre le problème pour lequel il est conçu. Les données peuvent être étiquetées pour indiquer au modèle quelles caractéristiques il doit reconnaître. Ils peuvent également être sans étiquette et le modèle doit découvrir et extraire de lui-même des caractéristiques récurrentes. La deuxième étape consiste à choisir un algorithme à exécuter sur l'ensemble de données d'apprentissage. La troisième étape consiste à entraîner l'algorithme. Il s'agit d'un processus itératif. La variable s'exécute dans l'algorithme et compare le résultat avec le résultat qu'elle devrait produire. Ensuite, après des réexécutions des variables sont faites jusqu'à ce que l'algorithme produise

le résultat correct dans la plupart des cas. L'algorithme formé de cette manière est un modèle d'apprentissage automatique. La dernière étape consiste à améliorer le modèle en utilisant ce dernier sur de nouvelles données, et sa source dépend du problème à résoudre.[72]

1.3.3 Algorithmes d'apprentissage automatique

L'apprentissage automatique est divisé en trois sous catégories à savoir l'apprentissage supervisé, apprentissage non supervisé, l'apprentissage par renforcement.

La technique de l'apprentissage non supervisé[68] consiste à entraîner des modèles, sans réaliser d'étiquetage manuel ou automatique des données au préalable. Les algorithmes regroupent les données en fonction de leur similitude, sans aucune intervention humaine.

L'apprentissage par renforcement [69] désigne l'ensemble des méthodes qui permettent à un agent d'apprendre à choisir quelle action prendre, et ceci de manière autonome.

•**Apprentissage supervisé** L'apprentissage supervisé est une méthode qui permet à une intelligence artificielle (IA) d'appliquer ses fonctions en toute autonomie [3]. D'une manière générale, chaque exemple est trié en une paire composée d'un objet d'entrée et d'une valeur de sortie souhaitée.

•**Fonctionnement** [5] Le processus d'apprentissage automatique supervisé est essentiellement un processus fonctionnant comme suit :

•**Importer et préparer un Dataset (x,y)** : il inclut toujours 2 types de variables :

1. Une variable objective (target) y.
2. Une ou plusieurs variables caractéristiques(features) x.

•**Apprentissage et évaluation** : Généralement on divise nos données en deux types :

1. Training Data : sont un ensemble initial de données utilisées pour aider un programme à comprendre comment appliquer des technologies. Il peut être complété par des ensembles de données ultérieurs appelés Testing et Validation Data.
2. Testing Data : dans l'apprentissage automatique c'est un ensemble de données secondaire utilisé pour tester un programme d'apprentissage automatique après qu'il ait été formé sur un ensemble Training Data initial. L'idée est que les modèles prédictifs ont toujours une capacité inconnue qui doit être testée. Avec un tel Dataset, il devient possible de prédire de nouvelles

valeurs à partir de valeurs de x en développant un modèle [4].

•**Développer un Modèle aux paramètres aléatoires :** Un modèle de Machine Learning ne repose pas sur une démonstration mathématique ou une équation physique. A la place, il est construit à partir de données, comme un modèle statistique. Dans les faits, c'est à nous de choisir le type de modèle et c'est à la machine de trouver les paramètres de cette fonction qui donnent les meilleurs résultats[5].

•**Déploiement :** La dernière étape de l'apprentissage automatique est le déploiement, où nous déployons le modèle dans le système réel. Si le modèle préparé produit un résultat précis selon les exigences avec une vitesse acceptable, nous déployons le modèle dans le système réel.

1.3.3.1 Régression et Classification

On distingue deux types de problèmes dans l'apprentissage supervisé, les problèmes de régression et les problèmes de classification. On considère que les problèmes de prédiction d'une variable quantitative sont des problèmes de régression tandis que les problèmes de prédiction d'une variable qualitative sont des problèmes de classification [6].

Régression

•**Régression linéaire :** La régression linéaire est l'un des algorithmes d'apprentissage supervisé les plus populaires. C'est un type d'analyse prédictive de base. La forme la plus simple de l'équation de régression est définie par :

$$y = ax + b \quad (1.1)$$

où y = variable objective (target) ,b = constante, a = coefficient de régression, x = variable caractéristique (feature), On parle ici de régression linéaire simple.

Pour la régression linéaire multiple on écrira

$$y = b + a_1x_1 + \dots + a_nx_n \quad (1.2)$$

Avec x_1 jusqu'à x_n les variables caractéristiques

- a_1 jusqu'à a_n les coefficients de régression respectifs des variables [6].

•**Régression logistique :** La régression logistique, propose le résultat sous forme de probabilités de la classe par défaut.

Le résultat appartient donc à l'intervalle [0 : 1]. La valeur y de sortie est générée par la transformation de la valeur x, à l'aide de la fonction logistique :

$$h(x) = \frac{1}{1 + \exp(-x)} \quad (1.3)$$

Un seuil est ensuite appliqué pour forcer cette probabilité dans une classification binaire [6].

•**Les arbres de classification et de régression :** [6] Les arbres de classification et de régression (CART : Classification And Regression Tree) sont une forme simple d'arbres de décision qui ont deux composantes :

- Les nœuds de branchement, qui représentent une seule variable d'entrée et offrent un seul point de partage sur la variable.
- Les nœuds feuilles, qui représentent les deux variables de sortie.

Classification

•**K-nearest neighbors :** KNN [7] est l'un des algorithmes simples qui peuvent être utilisés pour la classification. Indépendamment de sa simplicité, cette méthode est assez efficace en tant que classificateur. Le concept de base de KNN est d'avoir plusieurs échantillons de formation (entraînement) et des échantillons de test, et ainsi pour attribuer une classe au nouveau point, KNN est capable de prédire que serait sa classe en fonction des k-voisins les plus proches.

Choix de la valeur de K :[8] La valeur de K est directement proportionnelle à la régularité de la frontière. Donc, à partir de cela, nous pouvons comprendre que si la valeur K est définie sur 1, le modèle sera sous-apprentissage (underfitting) et si la valeur K est définie sur un grand nombre, il y aura un sur-apprentissage (overfitting). Ainsi la règle de base est avec n nombre de données : $K = \sqrt{n}$.

GÉNÉRALITÉS

•**Support Vector Machines :**[70] sont des modèles de machine learning supervisés centrés sur la résolution de problèmes de discrimination et de régression mathématiques.

Le principe des SVM consiste à ramener un problème de classification ou de discrimination à un hyperplan dans lequel les données sont séparées en plusieurs classes dont la frontière est la plus éloignée possible des points de données (ou "marge maximale"). D'où l'autre nom attribué aux SVM : les séparateurs à vaste marge. Le concept de frontière implique que les données soient linéairement séparables. Pour y parvenir, les support vector machines font appel à des noyaux, c'est-à-dire des fonctions mathématiques permettant de projeter et séparer les données dans l'espace vectoriel, les "vecteurs de support" étant les données les plus proches de la frontière. C'est la frontière la plus éloignée de tous les points d'entraînement qui est optimale, et qui présente donc la meilleure capacité de généralisation

•**Decision Tree :**[41] L'arbre de décision est un cadre hiérarchique composé de règles de décision qui divise de manière récursive les variables indépendantes en zones homogènes. C'est une approche d'aide à la décision qui utilise une représentation arborescente des options et de leurs résultats potentiels.

Lors de la prise de décisions, un modèle arborescent des décisions et de leurs répercussions probables est utilisé pour faciliter le processus de décision. L'arbre de décision intègre les éléments de manière hiérarchique, la caractéristique la plus significative étant positionnée à la racine de l'arbre. Chaque nœud de l'arborescence représente l'une des caractéristiques et chaque feuille représente la valeur de classe la plus courante. Cela bénéficie de la possibilité d'appliquer rapidement des « règles » de segmentation à des éléments autres que ceux qui composent l'ensemble de données de départ et le groupe représentatif n'est pas identifié

•**Random Forest :**[41] La forêt aléatoire est une méthode d'apprentissage d'ensemble pour la classification, la régression et d'autres tâches qui fonctionne en construisant une multitude d'arbres de décision au moment de l'entraînement et en sortant la classe qui est le mode des classes (classification) ou la prédiction moyenne (régression) des arbres individuels.

Comme les ensembles de données d'entraînement des arbres de décision ont tendance à être surajustées et déséquilibrées, les forêts aléatoires corrigent cette l'habitude.

La forêt aléatoire, comme son nom l'indique, se compose d'un grand nombre d'arbres de décision individuels qui fonctionnent comme un ensemble. Les facteurs qui influencent la consommation de mémoire et le temps de traitement sont la quantité de données et le nombre d'arbres dans la forêt. Pour la classification, l'échantillon de test d'entrée est introduit dans tous les arbres de décision, puis chaque arbre de décision associe l'échantillon à une classe spécifique. Le vote majoritaire est effectué sur le résultat de tous les arbres de décision pour classer l'échantillon de test. Au cours du processus de formation, ces arbres sont générés à l'aide d'échantillons aléatoires à partir de données d'entraînement. Pour diviser davantage chaque nœud de l'arborescence, un nombre aléatoire d'entités est utilisé. Ce caractère aléatoire dans la génération d'arbres, ainsi que le vote majoritaire sur le résultat de tous les arbres de la forêt, ajoutent de la diversification au classificateur aléatoire des forêts et le rendent immunisé contre le surajustement.

1.4 Deep Learning

L'apprentissage en profondeur est le sous-ensemble de l'apprentissage automatique, qui est un sous-ensemble de l'intelligence artificielle. C'est un type d'apprentissage automatique inspiré de la structure du cerveau humain en termes d'apprentissage en profondeur, cette structure s'appelle un réseau de neurones artificiels (ANN).

Il crée des réseaux de neurones artificiels capables d'apprendre et de prendre des décisions intelligentes à l'aide d'algorithmes pour permettre à l'ordinateur de s'entraîner pour effectuer des tâches telles que la reconnaissance de la parole et des images, en appliquant des réseaux neurones multicouches à la grande quantité de données.[71]

•les réseaux de neurones

Un réseau de neurones est un ensemble de méthodes d'analyse et de traitements des données permettant de construire un modèle de comportement à partir de données qui sont des exemples de ce comportement. Il est très adapté pour les problèmes de classification supervisée. En effet, étant construit à partir d'un ensemble d'échantillons provenant des différentes classes (à l'aide

GÉNÉRALITÉS

d'un algorithme d'apprentissage), un réseau de neurones est généralement apte à déterminer la classe d'un nouvel élément à classifier.[71]

•Principe de fonctionnement d'un neurone formel

Un réseau de neurones est construit autour d'un ensemble d'unités de calcul très simples, appelées neurones formels inspirés du fonctionnement des neurones biologiques. Un neurone formel reçoit des informations provenant des sorties de plusieurs neurones du réseau et peut soit s'activer ou soit s'inhiber dans sa sortie. Comme le montre la figure 1.1 son activation ou son inhibition est définie par sa fonction d'activation F et le résultat de cette fonction sera à son tour transmis aux autres neurones du réseau auxquels il est connecté [11][12].

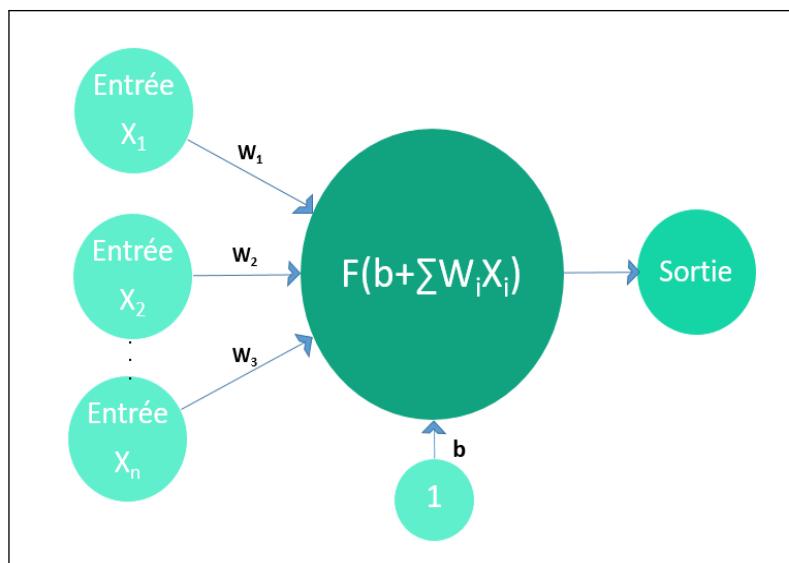


FIGURE 1.1 – Représentation d'un neurone formel

La figure 1.1 illustre une représentation d'un neurone formel où x_i est la valeur de sortie du nœud i représentant une entrée du neurone formel considéré, w_i est le poids associé à la connexion entre le neurone d'entrée i et le neurone observé et b est le biais de ce dernier. La fonction F définit le rendement d'un neurone à partir de ses entrées.

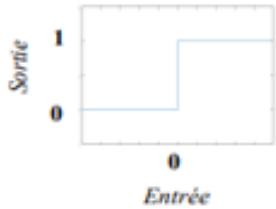
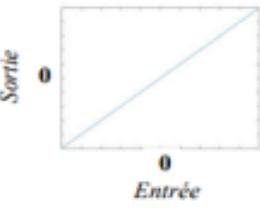
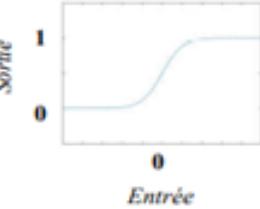
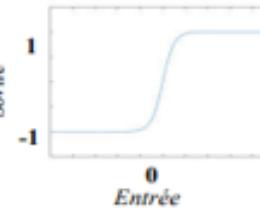
Nom de la fonction	Formule	Représentation
Seuil	$F(x) = \begin{cases} 1 & \text{si } x \geq 0 \\ 0 & \text{sinon} \end{cases}$	 <p>Sortie 1 0 Entrée</p>
Linéaire	$F(x) = x$	 <p>Sortie 0 Entrée</p>
Sigmoïde (ou logistique)	$F(x) = \frac{1}{1 + e^{-x}}$	 <p>Sortie 1 0 Entrée</p>
Tangente hyperbolique	$F(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	 <p>Sortie 1 -1 0 Entrée</p>

FIGURE 1.2 – Tableau : Exemples de fonctions d’activation[71]

La figure 1.2 illustre un tableau représentant les fonctions d’activation les plus utilisées.

1.5 Bluetooth à énergie réduite (Bluetooth Low Energy)

Le Bluetooth est un protocole sans fil bien connu utilisé intensivement dans l’industrie du téléphone portable, il a évolué au fil des années pour répondre aux nouvelles exigences du marché de la technologie. Bluetooth a annoncé dans Bluetooth Version 4.0 un nouveau protocole appelé Bluetooth Low Energy ciblant principalement les appareils IoT alimentés par des batteries de très faible capacité.

Ainsi, à partir de la version 4.0, Bluetooth dispose de trois protocoles le Bluetooth classique, Bluetooth haut débit et Bluetooth à énergie réduite. La spécification Bluetooth 4.0 autorise un ou les deux appareils monomode et bi-mode. Les appareils monomodes ne prennent en charge

GÉNÉRALITÉS

que BLE, tandis que les appareils bi-modes peuvent fonctionner avec BLE ou le protocole Bluetooth classique.[9]

Un module BLE pourra jouer un rôle central ou périphérique :

Central BLE

Un BLE central est un appareil qui recherche des appareils Bluetooth pour se connecter et utiliser les informations qu'ils hébergent. Habituellement, les appareils mainframe sont plus riches en termes de ressources telles que la puissance de calcul par rapport aux appareils périphériques, on parle alors de maître.[9]

Périphérique BLE

Les appareils BLE peuvent être analysés par l'appareil Bluetooth Central. Une fois qu'une connexion BLE a été établie, l'appareil est appelé esclave.[9]

La pile BLE (Stack BLE)

Bluetooth a plusieurs couches de protocoles qui forment une pile.[13] Chaque couche a un travail spécifique. Il ressemble au modèle OSI.

Les couches traitent des signaux radio, d'autres du chiffrement/déchiffrement, de la fragmentation/défragmentation, et d'autres traitent de la façon d'accéder aux données et de distribuer les informations aux appareils. La figure 1.3 illustre ces couches.

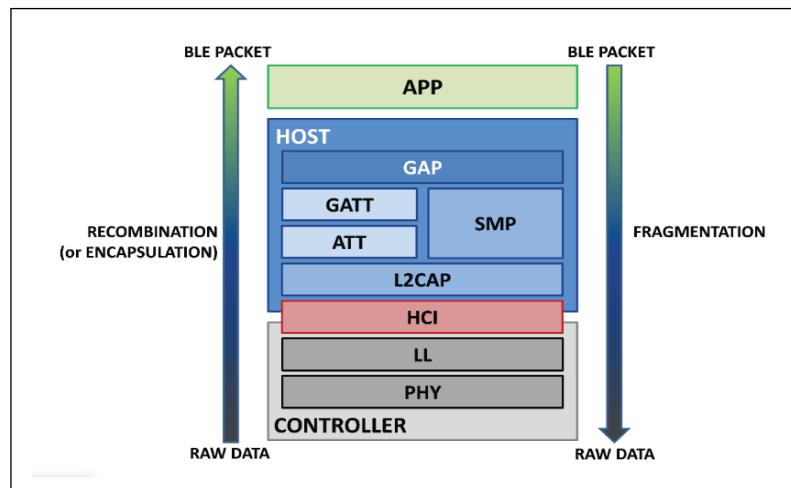


FIGURE 1.3 – Architecture de Bluetooth Low Energy[67]

Les principaux blocs de la pile Bluetooth :[13]

1. Contrôleur : regroupe les couches, physique et lien (Link Layer) qui sont responsables de

GÉNÉRALITÉS

toutes les opérations liées à la liaison radio, transactions, etc. La couche liaison (Link Layer) introduit deux modes opposés : maître et esclave, Le maître est responsable de la synchronisation entre les deux appareils connectés, l'autre appareil est alors l'esclave.

2.Hôte : combine les couches SMP, GAP, ATT et GATT qui gèrent l'accès aux données en direct et découvrent les appareils BLE. L'hôte gère tous les aspects liés à la sécurisation du lien au niveau de la couche Security Manager (SM).

-La couche ATT (Attribute Protocol) gère le support de toutes les données exposées par un appareil.

-La couche GATT (Generic Attribute Profile) fournit quant à elle un ensemble de sous-procédures basées sur la couche ATT pour permettre l'orchestration de toute la gestion des données prise en charge par un appareil. La couche GATT introduit les rôles de serveur et de client.

-La couche GAP est chargée d'établir le lien et de superviser la connexion entre deux appareils utilisant deux modes :

- Mode publicité(Advertising), un appareil émet des trames sur les 3 canaux d'avertissement et ces trames sont accessible à tout autre appareil écoutant sur ces 3 canaux.

- Mode connecté, un lien est établi entre deux appareils et eux seuls peuvent communiquer ensemble sur des canaux négociés connus ; le lien peut éventuellement être authentifié et crypté pour augmenter la sécurité.

3.Interface hôte-contrôleur(HCI) : une couche de liaison (interface) entre l'hôte et le contrôleur. Cette interface fournit une méthode standard d'accès aux capacités de la bande de base Bluetooth (un protocole avec des commandes spécifiques).

4.Application : c'est où se trouve le vrai code, sur une mémoire flash MCU par exemple.[13]

La couche GATT (Generic Attribute Profile)

GATT définit la façon dont deux appareils Bluetooth Low Energy transfèrent des données dans les deux sens à l'aide de concepts appelés Services et caractéristiques illustré dans la figure 1.4. Il utilise un protocole de données générique appelé ATT (Attribute Protocol), qui est utilisé pour stocker les services, les caractéristiques et les données associées dans une table de choix simple à l'aide d'ID 16 bits pour chaque entrée de la table.[66]

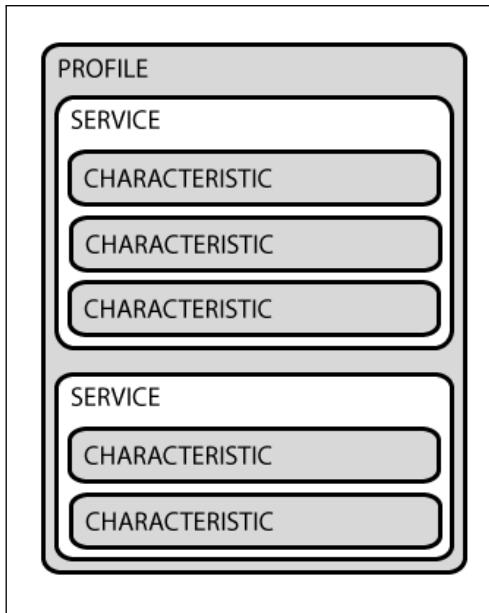


FIGURE 1.4 – Les objets basiques d'une transaction GATT dans BLE [66]

La figure 1.4 illustre des objets imbriqués de haut niveau appelés Profils, Services et Caractéris-

tiques qui sont la base des transactions GATT dans BLE.

Services

Les services sont utilisés pour diviser les données en entités logiques et contiennent des morceaux spécifiques de données appelés caractéristiques. Un service peut avoir une ou plusieurs caractéris-

tiques, et chaque service se distingue des autres services au moyen d'un ID numérique unique appelé UUID, qui peut être soit 16 bits (pour les services BLE officiellement adoptés) ou 128 bits (pour les services personnalisés).[66]

Caractéristiques

Le concept de niveau le plus bas dans les transactions GATT est la caractéristique, qui encapsule un seul point de données. Chaque Caractéristique se distingue par un UUID 16 bits ou 128 bits prédéfini. L'utilisateur est libre d'utiliser les caractéristiques standard définies par le SIG Bluetooth ou de définir ses propres caractéristiques personnalisées que seuls ses périphériques et logiciels comprennent.[66]

Il est possible d'implémenter une interface de type UART simple avec un « service UART » personnalisé et deux caractéristiques, l'une pour le canal TX et l'autre pour le canal RX, où une caractéristique peut être configurée en lecture seule et l'autre a des priviléges d'écriture.[66]

1.6 Communication UART (Universal Asynchronous Receiver Transmitter)

UART représente l'émetteur récepteur asynchrone universel. Il est dédié au matériel lié à la communication série. L'UART est l'une des techniques de communication série les plus couramment utilisées. Il est utilisé dans plusieurs applications telles que les modules Bluetooth. Dans cette communication, il existe deux types d'UART tels que la transmission d'UART TX et la réception d'UART RX.

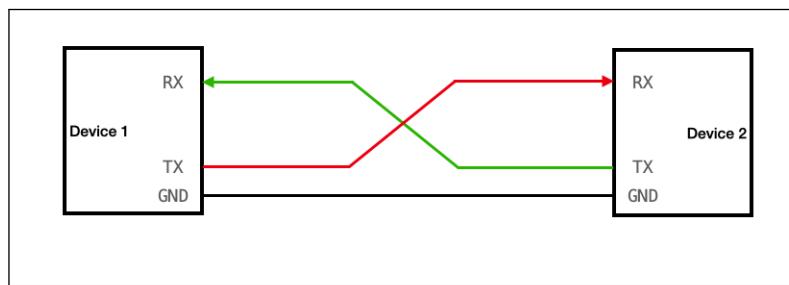


FIGURE 1.5 – communication UART

La figure 1.5 montre que l'interface UART se compose de deux broches : les broches Rx et Tx. La broche Rx est utilisée pour recevoir des données. La broche Tx est utilisée pour transmettre des données. Lorsque deux appareils sont connectés à l'aide d'un UART, la broche Rx d'un appareil est connectée à la broche Tx du deuxième appareil.

1.7 Service UART nordique (NUS)

Le service Bluetooth LE GATT Nordic UART est un service personnalisé qui reçoit et écrit des données et sert de pont vers l'interface UART.[67]

Service UUID : L'UUID de service spécifique au fournisseur 128 bits est

6E400001-B5A3-F393-E0A9-E50E24DCCA9E (décalage 16 bits : 0x0001).[67]

Caractéristiques : Ce service présente deux caractéristiques.

Caractéristique RX (6E400002-B5A3-F393-E0A9-E50E24DCCA9E)

Écrire des données dans la caractéristique RX pour les envoyer à l'interface UART.

Caractéristique TX (6E400003-B5A3-F393-E0A9-E50E24DCCA9E)

Activer les notifications pour que la caractéristique TX reçoit des données de l'application.

L'application transmet toutes les données reçues via UART sous forme de notifications.[67]

1.8 CONCLUSION

Dans ce chapitre, nous avons défini quelques notions utiles pour mieux comprendre le contexte de notre projet et cela en commençant par une description des troubles musculosquelettiques en citant quelques exemples des troubles liés à la colonne vertébrale et en illustrant les différentes régions de cette dernière. Ensuite, nous avons exposé une définition générale du machine Learning et son fonctionnement. Après cela, nous nous sommes familiarisés avec quelques algorithmes de l'apprentissage supervisé et du deep Learning, pour terminer nous avons défini le bluetooth à énergie réduite et on a détaillé la pile BLE et la couche Generic Attribute Profile et présenté privèrement la communication UART et le service NUS.

Collecte des données sur la mauvaise posture

Sommaire

2.1	INTRODUCTION	18
2.2	Les images en temps réel.	18
2.3	Les informations de poids	18
2.4	Les informations sur l'angle d'inclinaison.	21
2.5	L'information de la courbure de la colonne vertébrale	24
2.6	Information fournie par les capteurs EMG	27
2.7	CONCLUSION	31

2.1 INTRODUCTION

Plusieurs recherches se sont portées sur le développement des systèmes de prévention de la douleur de la colonne vertébrale pour les personnes assises, ces derniers surveillent et améliorent la position assise en temps réel. Dans cette section, nous allons étudier les bases et les spécifications des systèmes d'analyse et de surveillance de la position assise, les informations nécessaires pour définir la pose humaine, les limitations des technologies et des systèmes utilisés, postures du corps humain identifiable par les différentes informations fournies par les technologies de détection. Ces systèmes sont classés selon le type d'information fournie par les technologies de détection pour définir la posture de la personne.

2.2 Les images en temps réel.

Ces systèmes basés sur les caméras de vision fournissent des images en temps réel pour surveiller la posture d'une personne au fil du temps.

Le système compare la photo de la posture actuelle avec la photo de la posture correcte et alerte l'utilisateur en cas de mauvaise posture. Une étude récente de **Shin et son équipe** a proposé un système qui utilise une caméra de profondeur infrarouge Kinect pour surveiller la posture [20]. Le système collecte des données visuelles et émet une alarme en cas de mauvaise posture. Ces systèmes aident les utilisateurs à améliorer leur posture et à corriger leur position assise. Cependant, ces systèmes menacent la vie privée en raison de l'utilisation d'informations visuelles.

2.3 Les informations de poids

Dans la littérature, de nombreux systèmes de surveillance de la posture basés sur les informations de poids ont été proposés.

Jongryun et al [21] ont décrit un nouveau système de surveillance de la posture assise. Le système analyse le rapport de poids pour identifier la position assise de la personne. Le système

COLLECTE DES DONNÉES SUR LA MAUVAISE POSTURE

se compose de quatre capteurs de pesage placés sur le dossier et l'assise de la chaise. Les informations collectées sont envoyées à l'ordinateur via la carte Arduino pour analyser et classer la posture assise.

Ce système est capable de définir différents types de postures assises comme illustré à la figure 2.1.

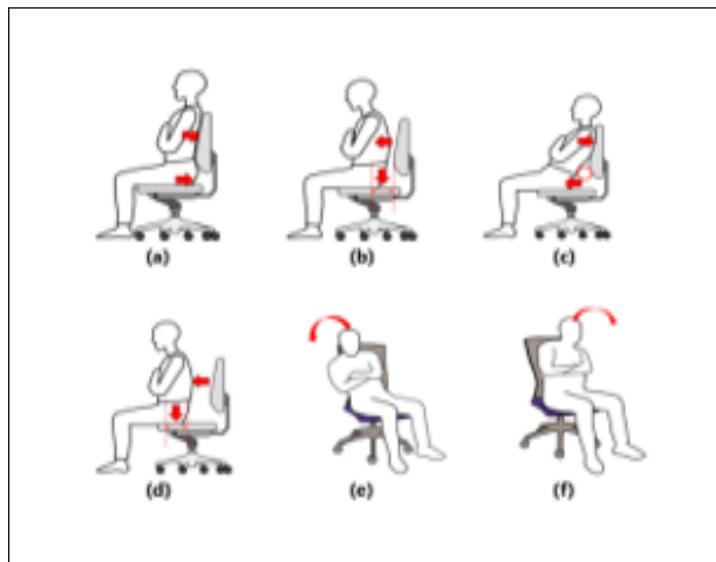


FIGURE 2.1 – Types de postures assises [21]

La figure 2.1 montre les six types de posture (a) représente une position assise verticale avec dossier,(b) position assise verticale sans dossier c) position assise à l'avant avec dossier d) position assise avant sans dossier e) position assise à gauche et f) position assise à droite .

Une nouvelle recherche de **Jullia Birsan et al** [22] est basée sur l'information sur la répartition du poids. Le système est composé de onze capteurs de pression : neuf capteurs sont placés sur l'oreiller et deux capteurs sont montés à l'arrière. Le poids fourni les données sont analysées afin de définir la forme de la colonne vertébrale.

Cheng Jingyuan et al. [23] ont proposé un suivi de la posture du système. Le système fournit le poids total, la répartition du poids et la pression des quatre jambes pour reconnaître la posture et surveiller les mouvements de la tête et des mains. Le capteur de pression est collé sous le pied de la chaise. Le système de surveillance de la posture basé sur le poids peut définir efficacement la posture assise.

COLLECTE DES DONNÉES SUR LA MAUVAISE POSTURE

Slavomir Matuska et al [33] ont conçu un système qui peut être facilement mis en œuvre dans n'importe quel espace de bureau où les gens n'ont pas à utiliser la même chaise tous les jours.

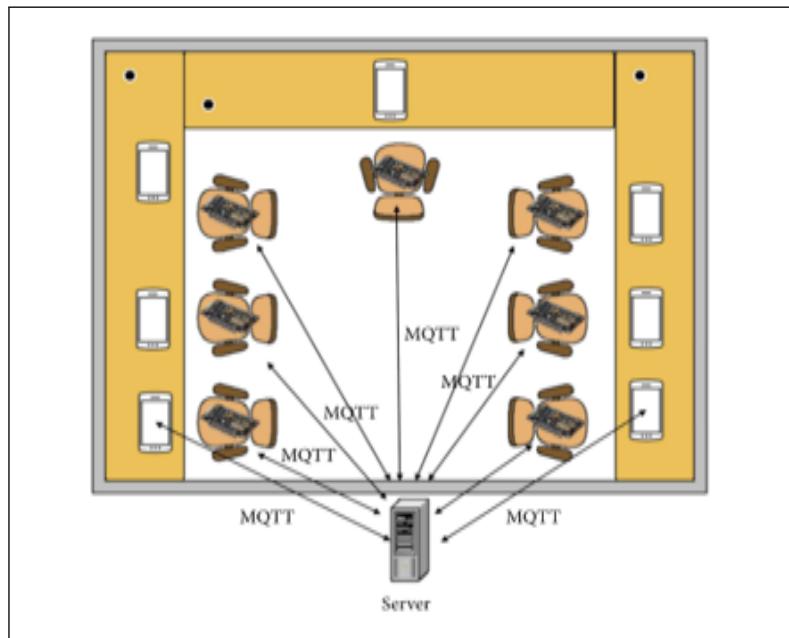


FIGURE 2.2 – La proposition de concept de système. [33]

La figure 2.2 illustre le concept de système recommandé. L'ensemble du système se compose de différents nombres de chaises, de serveurs cloud et de postes clients, et de téléphones intelligents. Chaque chaise possède un dispositif électronique basé sur un microcontrôleur Arduino, une alimentation externe et six capteurs de force flexibles (Le siège est équipé de 4 capteurs et le dossier de 2 capteurs). Les positions appropriées des capteurs sont trouvées empiriquement. Le stockage en réseau de Quality Network Appliance Provider (QNAP) comprend des solutions cloud. Il possède un agent MQTT (Message Queuing Telemetry Transport) pour la communication, Node-RED pour la logique et une base de données Mongo pour le stockage des données.

La routine quotidienne des personnes travaillant dans un bureau intelligent devrait être la suivante : La personne choisit une chaise libre dans le bureau et s'assoit. Le matériel Arduino se réveillera du sommeil à ce stade et se connectera au cloud. La personne allume l'application mobile et se connecte à la chaise. Chaque chaise intelligente a un numéro d'identification pour se connecter.

Ils ont défini 9 postures assises différentes pour examiner l'état de la position de l'utilisateur.



FIGURE 2.3 – Les positions assises définies par cette étude. [33]

La figure 2.3 définit les neuf positions sur lesquelles ce système s'est basé, la position assise numéro 1 est considérée comme une posture assise correcte avec une charge uniformément répartie. Les positions 2 à 5 représentent l'état où le participant est assis, mais son poids n'est pas réparti uniformément et les positions de 6 à 9 sont l'état où le participant est dans une position assise malsaine. Cette dernière signale une charge excessive d'un côté.

Les informations sur la posture assise avec des données supplémentaires sont affichées dans l'application smartphone. Une fois le travail terminé, la personne se déconnecte de la chaise. Enfin, l'utilisateur peut consulter le rapport quotidien.

Les systèmes présentés basés sur les informations du poids utilisent une chaise ou une plate-forme spéciale pour mettre en œuvre les capteurs. Ces systèmes ne sont pas portables et utilisés dans un environnement spécifique.

2.4 Les informations sur l'angle d'inclinaison.

L'angle d'inclinaison par rapport à la gravité et à l'accélération linéaire est fourni par un capteur inertiel, qui est largement utilisé en raison de sa portabilité. Le capteur peut être collé sur les vêtements d'une personne.

COLLECTE DES DONNÉES SUR LA MAUVAISE POSTURE

Wai Yin Wong et Man Sang Wong [24] ont proposé un système de surveillance de la posture assise. Le système fournit des angles d'inclinaison dans les régions du tronc thoracique et lombaire. Le système se compose de capteurs inertIELS, un accéléromètre 3D et trois gyroscopes. Le système suit la courbure de la posture lors du mouvement du tronc.

Wang et al [25] ont proposé un système de surveillance de la posture du patient. Le système est un vêtement de rééducation intelligent. Le capteur inertiel est installé sur les vêtements du patient, le positionnement de ce dernier doit être fait correctement afin que les capteurs puissent être placés dans les bonnes positions dans l'épaule et le torse du patient. Il s'agit d'une condition préalable à ce système car les capteurs sont calibrés pour fonctionner correctement dans des points spécifiques sur le corps du patient. L'angle thoracique à la position d'enfoncement est défini par les éléments inertIELS placés sur les vertèbres T1 et T5, comme le montre la figure 2.4.c

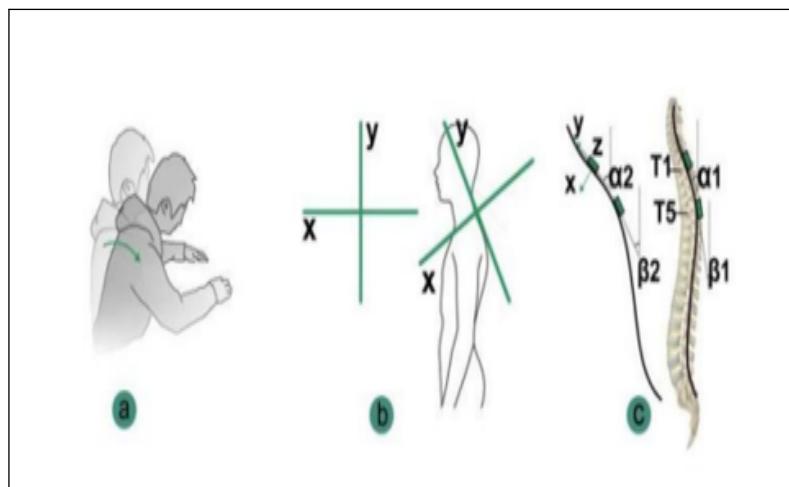


FIGURE 2.4 – Calculs d'angle. (a) Mouvement de rémunération. (b) Degré zéro Étalonnage. (c) Modèle d'étalonnage [25].

La figure 2.4.a illustre la détection d'un mouvement de compensation, la Figure. 2.4.b montre que le capteur de torse est calibré à sa position zéro car le squelette humain n'est pas complètement droit, la figure .2.4.c montre l'angle thoracique correspondant aux mouvements de flexion et d'extension de la partie supérieure du tronc définie par T1 et T5. Le mouvement compensatoire est la moyenne des angles thoracique selon le plan vertical. Les calculs concernant les informations d'angle sont envoyés au processeur Arduino via une communication Bluetooth. Si le patient a une position courbée et une mauvaise posture, le

COLLECTE DES DONNÉES SUR LA MAUVAISE POSTURE

système de rétroaction consiste en des vibrations, de signaux visuels et audio provenant d'un moteur et d'un smartphone pour rappeler le patient et lui permettre d'améliorer sa posture assise.

Maheswaran Shanmugam et al [26] ont proposé un nouveau système de surveillance de la posture assise des travailleurs. Le système classe la posture reconnue comme mauvaise ou bonne posture sur la base des informations d'angle. Le système utilise la fonction d'Arduino (map function) pour convertir l'angle d'accélération et l'accélération du gyroscope pour définir l'angle d'inclinaison. Le système est équipé d'un gyroscope et d'un accéléromètre, placés en bas du dos ou dans la poche de la chemise, le module Bluetooth assure la communication entre les différents composants du système, et l'application Android fonctionnant sur le smartphone permet d'enregistrer et d'analyser informations du capteur et être notifié par des LEDs et un système d'avertissement avec vibrEUR.

Alexandra Kanaris, Katherine Lee et Zoë Roecker [46] ont développé un système qui surveille la posture et informe un client atteint de la maladie Parkinson qu'il est affalé. Cet appareil est confortable à porter tout au long de la journée et maintient sa fonctionnalité pendant au moins 15 heures en charge. Il est discret et fonctionne à la fois pour les positions assises et debout. La base de cet appareil est l'utilisation d'un accéléromètre pour mesurer l'angle d'inclinaison. L'appareil est placé sur le dos du client. Lorsque le client se tient debout contre un mur et est dans une posture parfaite, il appuie sur un bouton de leur application nommée One Thing Straight pour calibrer l'appareil selon un réglage « vertical ». Alors qu'il décline dans une posture affaissée, son dos se courbe et l'appareil mesure un angle d'inclinaison différent. Si cette différence d'angle dépasse 10 degrés, le client est alerté par un schéma de vibration aléatoire via une vibration dans son téléphone. Le seuil de 10 degrés a été choisi par l'analyse de la courbure de la colonne vertébrale du même client dans des images de bonne et de mauvaise posture.

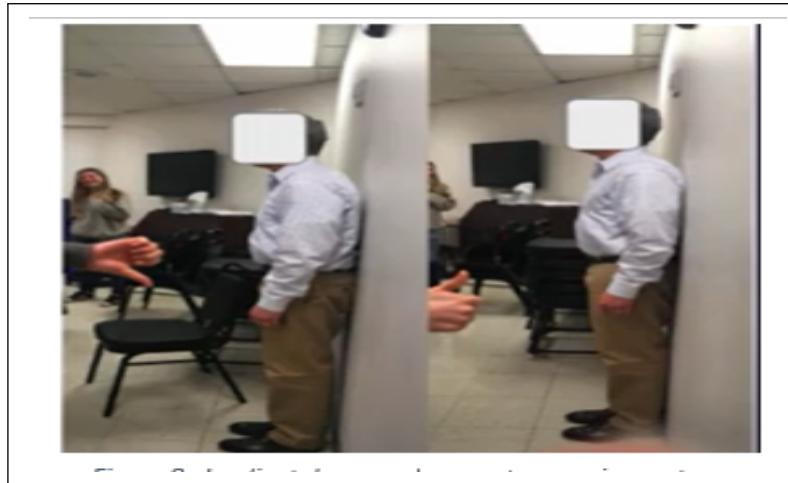


FIGURE 2.5 – Le client dans une bonne et mauvaise posture.

La figure 2.5 illustre deux photos du client, une dans une position correcte l'autre dans une position inadéquate, ceci permet de calculer le seuil de l'angle d'inclinaison. L'appareil se connecte à l'iPhone sans fil à l'aide de Low Energy Bluetooth et signale au téléphone de vibrer si le seuil est dépassé. Le téléphone ne cessera pas de bourdonner jusqu'à ce que l'angle d'inclinaison soit de retour sous la valeur seuil, ce qui indique que la posture a été corrigée.

Les systèmes basés sur l'angle d'inclinaison de la posture sont les systèmes de surveillance de la posture les plus efficaces. Ces systèmes sont capables de surveiller le mouvement de la colonne vertébrale en temps réel et donner une vue réelle de la forme de la colonne vertébrale. Cependant, la précision de l'angle d'inclinaison dépend sur la position des capteurs. En effet, les capteurs doivent être collés au corps de la personne.

2.5 L'information de la courbure de la colonne vertébrale

La mauvaise posture peut être détectée en fonction de la courbure de la colonne vertébrale. De nombreuses technologies de détection sont capables de fournir ces informations au système de surveillance de la posture afin d'élaborer une décision sur la posture de la personne. Dans la littérature les principales technologies utilisées sont les capteurs flexibles [27], les capteurs inducteurs [28] et les capteurs à fibre optique [29].

Capteurs flexibles

Les capteurs flexibles définissent la courbure de la colonne vertébrale en raison de la nature des matériaux. Ces derniers sont sensibles à l'extension sans changer les propriétés [30]. Le changement de résistance du capteur flexible fournit d'excellentes informations pour identifier la courbure de la colonne vertébrale. De plus, ce type de capteurs se caractérise par une longue durée d'utilisation et un faible coût [31].

Un suivi de système utilisant des capteurs flexibles est proposé par **Manju Gopinath et Angeline Kirubha** [27]. Le système est composé d'un capteur flexible et une cellule de charge. Le capteur de flexion est placé sur le mi-thoracique pour définir la posture de la personne. La cellule de charge est composée d'un capteur de poids afin de définir l'effort de la colonne vertébrale. Les capteurs flexibles dépendent de la position du capteur pour détecter la mauvaise posture [20]. En effet le capteur collé au lombaire ou le thoracique inférieur n'est jamais capable d'identifier la mauvaise posture. De plus, la petite courbure de la colonne vertébrale et l'angle de courbure ne sont pas détectable par les capteurs flexibles [31].

Capteurs inducteurs

La déformation du corps entraîne l'allongement et le redressement de la géométrie du capteur inducteur.

Emilio Sardini et al[28] proposent un suivi d'un système basé sur les informations de courbure de la colonne vertébrale définies par la variation d'inductance détectée par les capteurs inducteurs.

En fait les mesures d'impédance et la variation d'allongement sont utiles pour définir la courbure de la posture. Le système proposé dans [28] consiste de T-shirt, d'une unité de lecture, d'une unité de traitement PC et d'un système de retour d'information. Le T-shirt est équipé d'un capteur inducteur cousu à l'arrière et à l'avant, comme illustré dans la figure 2.6.

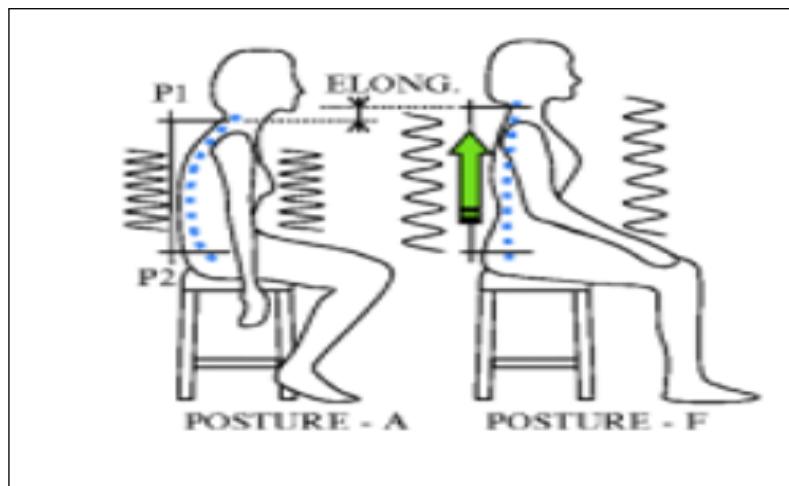


FIGURE 2.6 – Principe d'un système de surveillance de la posture à base d'inducteurs [1]

La figure 2.6 montre deux postures A et F en précisant l'état du capteur inducteur, en effet la mauvaise posture (A) entraîne l'allongement et le redressement de la géométrie du capteur inducteur.

Une unité d'affichage recueille les variations d'inductance et les envoie au PC. Le PC analyse les informations et les classe selon les informations de posture enregistrées. Deux vibro-feedback sont des systèmes de retour (feedback) pour alerter en cas de détection de mauvaise posture. Ce système se limite à définir la courbure de la colonne vertébrale sagittale. La courbure gauche et droite de la posture n'est jamais identifiée par les capteurs inducteurs.

Capteurs à fibre optique

La quantité de tension détectée entre la source de lumière et le capteur de lumière sont les principales informations pour identifier la forme de la colonne vertébrale.

Dunne et al[29] décrivent un système de surveillance de posture assise en utilisant des capteurs à fibre optique. Le système est composé de deux extrémités de fibres optiques collées sur un vêtement. La communication entre les différents composants du système est assurée par une série d'unités Bluetooth. Les informations recueillies sont traitées par un ordinateur qui classe les informations détectées. Une interface logicielle est implémentée pour envoyer un avertissement messages en cas de mauvaise posture. L'exactitude des informations collectées via le capteur à fibre optique dépendent de la position du capteur. Tout changement de la position du capteur peut endommager la précision du système pour définir la courbure de la colonne vertébrale.

COLLECTE DES DONNÉES SUR LA MAUVAISE POSTURE

Les systèmes de surveillance de la posture basés sur la courbe de la colonne vertébrale, l'informa-

tion de valeur définie par les capteurs flexibles [27], les capteurs à inductance [28] ou les capteurs à fibre optique [29] ont une limitation commune qui est leur sensibilité à la position du capteur.

2.6 Information fournie par les capteurs EMG

L'électromyographie (EMG) est une procédure de diagnostic pour évaluer la santé des muscles et des cellules nerveuses qui les contrôlent (neurones moteurs). Les résultats de l'EMG peuvent révéler un dysfonctionnement nerveux, un dysfonctionnement musculaire ou des problè-

mes de transmission des signaux nerf-muscle.

Les motoneurones transmettent des signaux électriques qui provoquent la contraction des muscles. Un EMG utilise de minuscules dispositifs appelés électrodes pour traduire ces signaux en graphiques, sons ou valeurs numériques qui sont ensuite interprétés par un spécialiste. [32]

Kevin Hung et al[51] ont étudié la relation entre l'électromyographie de surface (sEMG) et la posture du tronc.

Ils ont effectué des expériences où ils se sont concentrés sur les activités de quatre groupes de muscles du dos dans différentes conditions de postures et d'angles de la colonne vertébrale. Deux séries d'expériences ont été réalisées, une où le sujet était dans une position où ses mains étaient placées à côté du corps et une autre où le sujet écrivait sur la table devant lui. Une chaise sans dossier a été utilisée.

Trois sujets masculins en bonne santé ont participé à l'étude. Pour chaque sujet, sEMG a été mesuré au niveau du trapèze transversal (position A), du trapèze descendant (position B), du trapèze ascendant (position C) et du deltoïde postérieur (position D) sous différentes situations : assis, assis et écrivant sur une table, marchant et debout. Les emplacements de mesure et de placement des électrodes sur le sujet sont illustrés à la figure 2.7.

COLLECTE DES DONNÉES SUR LA MAUVAISE POSTURE

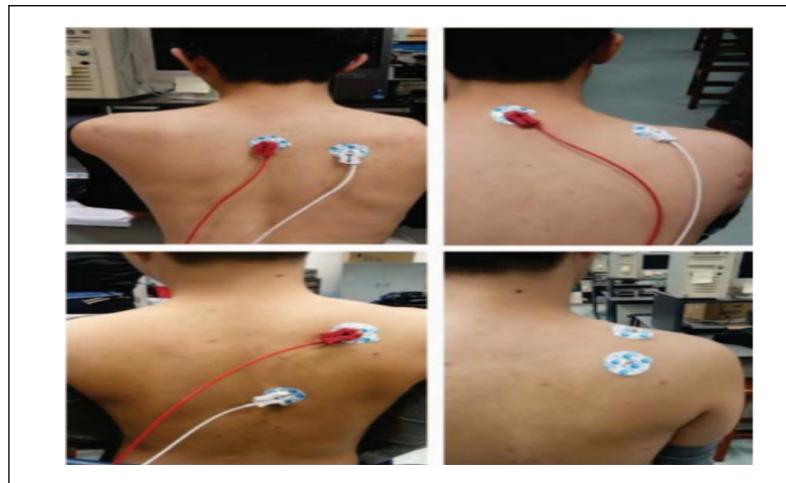


FIGURE 2.7 – Emplacements de mesure sEMG dans le dos[51]

La figure 2.7 illustre les emplacements des capteurs sEMG au dos. En partant du haut à droite (dans le sens des aiguilles d'une montre) : Position A (trapèze transversal), Position B (trapèze descendant), Position C (trapèze ascendant) et Position D (deltoïde postérieur).

Il a été constaté que le sEMG du trapèze transversal sur le dos était la source la plus fiable pour la détection d'une mauvaise posture du tronc. Le deltoïde postérieur n'était pas une source fiable, probablement en raison d'interférences EMG et d'artefacts de mouvement liés aux mouvements du bras. Les ratios integrated EMG (IEMG) et PSD (densité spectrale de puissance) étaient de bons indicateurs pour l'identification de la posture du tronc. Cependant, ils n'étaient précis que lorsque le sujet était assis ou debout, et le mouvement des membres supérieurs et inférieurs affectait la précision de la détection. Les résultats suggèrent qu'il existe des limites à l'utilisation de sEMG uniquement pour la détection d'une mauvaise posture du tronc ; cependant, les performances de détection peuvent être améliorées avec l'ajout d'un deuxième capteur.

Park et al [48] ont étudié les effets d'un dispositif de rétroaction basé sur l'EMG sur le système musculosquelettique humain en évaluant la cinématique des régions lombaire et cervicale pendant le fonctionnement de l'ordinateur. À l'aide d'un appareil EMG, des mesures ont été recueillies dans quatre conditions : avec rétroaction du muscle trapèze supérieur, avec les muscles érecteurs du rachis L4, avec les deux et sans rétroaction (contrôle). Un système d'analyse de mouvement Zebris a été utilisé pour mesurer les angles de la tête et du tronc

COLLECTE DES DONNÉES SUR LA MAUVAISE POSTURE

pendant le fonctionnement de l'ordinateur.

Au total, cinq marqueurs ont été utilisés pour analyser les données cinématiques, qui ont été obtenues à une fréquence d'échantillonnage de 50 Hz. Des marqueurs pour les angles de flexion de la tête vers l'avant et du tronc ont été placés sur la pointe latérale droite de l'acromion et le tragus de l'oreille, le milieu du grand trochanter, le 7e processus épineux cervical (C7) et le niveau des premiers processus épineux lombaires. (L1) par le même enquêteur. L'angle de tête vers l'avant a été défini comme l'angle entre la ligne du tragus à la ligne C7 et l'axe horizontal à C7. L'angle de flexion du tronc était défini comme l'angle entre la ligne allant de l'acromion à L1 et la ligne allant du grand trochanter à L1. Le récepteur du système d'analyse de mouvement 3D était positionné à environ 2 m et perpendiculairement au côté droit du plan sagittal du sujet.

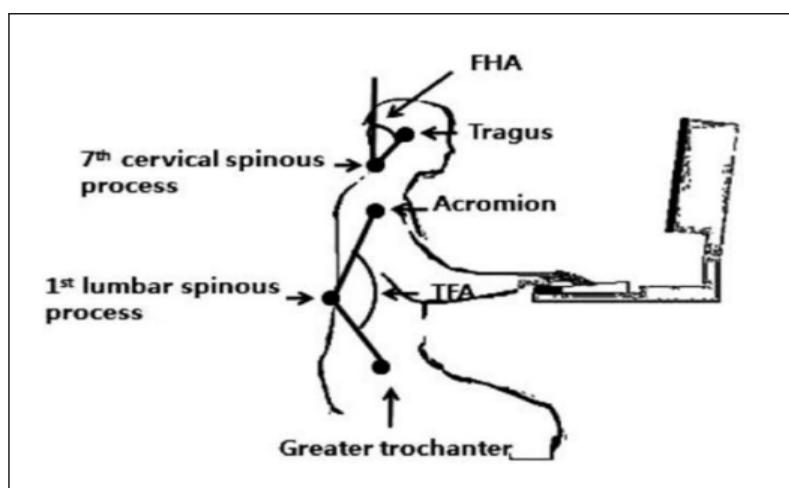


FIGURE 2.8 – Emplacement des marqueurs[48]

La Figure 2.8 illustre l'emplacement des marqueurs et définit des mesures de mouvement à savoir l'angle de tête avant (FHA) qui définit les changements d'angle entre la ligne du tragus au 7ème apophyse épineuse cervicale (C7) et l'axe horizontal en C7. L'angle de flexion du tronc (TFA) qui indique les changements d'angle entre la ligne de L1 à l'acromion et la ligne de L1 au grand trochanter. L'évaluation du feedback a effectivement réduit les changements cinématiques dans le plan sagittal par rapport au contrôle. Le feedback des érecteurs spinaux L4 a réduit la flexion du tronc et les angles de tête en avant, qui étaient significativement différents de ceux sous le feedback du trapèze supérieur et les conditions de contrôle. Ils ont observé que l'utilisation d'un dispositif de rétroaction basé sur l'EMG était efficace.

COLLECTE DES DONNÉES SUR LA MAUVAISE POSTURE

Section	Type de Capteurs	Solution	Informations fournies Par capteurs	Nombre Capteurs	Placement Des capteurs	Avantages	Inconvénients
Informations visuelles	Caméra de profondeur infrarouge Kinect	D.-J. Shin et al. 2017	Des données visuelles Les informations de profondeur du corps humain.	1	Le haut de l'écran du KINECT doit être au niveau des yeux de l'utilisateur.	Sans aucun capteur adhésif inconfortable	Menace la vie privée en raison de l'utilisation d'informations visuelles
Informations de poids	Capteurs de pesage	Jongryun et al 2017	Analyse le rapport du poids	4	4 capteurs de pesage placés sur le dossier et l'assise de la chaise.	Un coût minimal et moins d'éléments de dispositif est nécessaire.	- Nécessite une plateforme spéciale. -Difficulté de la mise en œuvre des capteurs de force dans la chaise.
	Capteurs de pression	Jingyuan Cheng et al 2013	Le poids total et la répartition du poids et la force exercée aux quatre jambes	4	Sous le pied de la chaise.	Déetecter des activités de haut niveau	
	Julia Birsan et all 2017	Julia Birsan et all 2017	Répartition du poids	11	9 capteurs sont placés sur l'oreiller et 2 sont montés à l'arrière du dos.	Efficace et rapide. Optimale pour tous les patients.	-Non portable.
	Capteurs de force	Slavomir Matuska et al 2020	Répartition de la charge	6	Le siège est équipé de 4 capteurs et le dossier de 2 capteurs	Facile à mettre en œuvre dans n'importe quel espace de bureau	
Informations sur l'angle d'inclinaison	Capteurs inertiels	Q. Wang et all.2015	Angle thoracique	2	Vertèbres T1 et T5 de la colonne vertébrale	la portabilité et la facilité d'utilisation	-La précision dépend de la position des capteurs
	Capteurs Inertiels, accéléromètre 3D et trois gyroscopes	Wai Yin Wong et all 2008	Angles d'inclinaison les angles thoracique et lombaires	3	Capteurs intégrés Aux vêtements sur le dessus et tronc et au niveau pelvien	Facile à mettre en place. Ne prend pas beaucoup de temps à utiliser.	
	Gyroscope et accéléromètre	Maheswaran Shamugam et all.2018	Angle d'accélération converti en angle de courbure	1	Le capteur est placé sur le bas du dos ou au niveau de la chemise poche	Simple, Economique et très efficace.	-les capteurs doivent être collés au corps de la personne.
	Accéléromètre	Alexandra Kanaris, Katherine Lee et Zoë Roecker. 2017	Angle d'inclinaison	1	L'appareil est placé sur le dos du client	Petit, confortable à porter. Discret et fonctionner à la fois pour les positions assises et debout.	
Information de La courbure de la colonne vertébrale	Capteur flexible Et charge cellulaire	Manju Gopinath and Angeline Kirubha.2015	Valeur de tension causée par variation de résistance pendant flexion de la colonne vertébrale et corps charge	2	Le capteur Flex placé Sur la région médiо-thoracique et la cellule de charge est placée entre des plates-formes sur lesquelles le sujet peut se tenir debout	Facilement mis en place sur le lieu de travail	-la petite courbure de la colonne vertébrale et l'angle de courbure ne sont pas détectable par les capteurs flexibles.
	Capteur inductif	Emilio Sardini et all.2015	Valeur d'inductance	1	Collé au t-shirt dans tout le dos du patient et la poitrine	Indépendance par rapport à l'unité distante, utilisation facile, légèreté et confort de port	-Définir la courbure de la colonne vertébrale sagittale uniquement.
	Capteur à fibre optique en plastique	Dunne et all. 2007	Degré de courbure	1	Fibre optique plastique intégrée au vêtement. Une source lumineuse et des capteurs de lumière sont placés sur les bords du POF. POF est collé au sujet en arrière.	Fiable, simple et portable surveillance à long terme	-Sensibilité à la position du capteur.
Information fournie par les capteurs EMG	Capteur EMG	Kevin Hung et al	Contractions du muscle	1	Trapèze transversal, descendant, ascendant et deltoïde postérieur	Efficace, faible coût	-L'utilisation de sEMG uniquement pour la détection d'une mauvaise posture du tronc
		Park et al		5	Angle de la tête vers l'avant (FHA). Angle de flexion du tronc (TFA).	Efficace de réduire la posture de la tête vers l'avant ainsi que la posture fléchie-relâchée	-Nombre de capteurs Très élevés ce qui est Gênant pour l'utilisateur.

TABLE 2.1 – Tableau récapitulatif des études précédentes sur la collecte des données de mauvaise posture

2.7 CONCLUSION

Dans ce chapitre, nous avons défini des systèmes de surveillance de posture existant basés sur des caméras de vision, qui fournissent des images en temps réel, des capteurs de pression qui fournissent des informations sur la répartition du poids, des capteurs inertiels et inducteurs ou fibre optique qui fournissent des informations sur la courbure de la colonne vertébrale, des capteurs inertiels qui déterminent l'angle d'inclinaison, ainsi que les systèmes basés sur les informations fournis par les capteurs d'électromyographie , en précisant les performances et les limitations de chaque système. On a conclu cette partie par un tableau récapitulatif. Cette étude pourra servir comme référence pour des futures recherches. Dans le chapitre suivant nous allons présenter les travaux existant concernant les modèles d'apprentissages utilisés pour classifier le type d'une posture assise.

Détection de mauvaise posture par Machine Learning

Sommaire

3.1	INTRODUCTION	33
3.2	Application d'approches de machine Learning pour classer la posture assise.	33
3.3	Application d'approches Deep Learning pour classer la posture assise	39
3.4	CONCLUSION	42

3.1 INTRODUCTION

Les effets néfastes d'une mauvaise posture assise et d'une position assise prolongée sur la santé ont été largement étudiés et les solutions pour enrayer cette épidémie sédentaire ont reçu une attention particulière ces dernières années. Avec les récent progrès des technologies de détection et de l'intelligence artificielle (IA), la surveillance et la correction de la posture assise sont parmi les problèmes clés à résoudre pour améliorer le bien-être humain.

Dans ce chapitre, nous présentons des études existantes qui ont exploré l'application des algorithmes de Machine Learning et du Deep Learning dans la reconnaissance de la posture assise tout en précisant leurs technologies et résultat obtenues.

3.2 Application d'approches de machine Learning pour classer la posture assise.

Plusieurs études ont montré l'efficacité des algorithmes d'apprentissage automatique dans la détection de la mauvaise position.

Zemp et al. [34] ont développé une chaise instrumentée avec des capteurs de force et d'accélération pour déterminer la précision de l'identification automatique de la position assise de l'utilisateur.

Ils ont appliqué cinq méthodes d'apprentissage automatique différentes à savoir la méthode Support Vecteur Machines (SVM), Régression multinomiale (RML), Réseaux de neurones et forêts aléatoires (RF).

Quarante et un sujets ont été invités à s'asseoir quatre fois dans sept positions assises prescrites différentes (total de 1148 échantillons). Seize valeurs de capteur de force et l'angle du dossier ont été utilisés comme variables caractéristiques pour la classification. Les différentes méthodes de classification ont été comparées au moyen d'une approche de validation croisée. La meilleure performance a été obtenue en utilisant l'algorithme de classification des forêts

DÉTECTION DE MAUVAISE POSTURE PAR MACHINE LEARNING

aléatoires (Random Forest), produisant une précision de classification moyenne de 90,9% pour les sujets avec lesquels l'algorithme n'était pas familier. La précision de la classification variait entre 81% et 98% pour les sept positions assises différentes.

La présente étude a montré la possibilité de classer avec précision différentes positions assises au moyen de la chaise de bureau instrumentée introduite combinée à des analyses d'apprentissage automatique.

L'utilisation de ces nouvelles approches pour l'évaluation précise de l'utilisation de la chaise pourrait donner un aperçu des relations entre la position assise, le comportement assis et l'apparition de troubles musculosquelettiques.

Une étude de **Ma et al.** [35] a utilisé 12 capteurs de pression textiles montés sur le repose-pied et le dossier d'un fauteuil roulant et un algorithme d'arbres de décision (J48) pour classer 5 postures assises spécifiques au fauteuil roulant. Cinq techniques de classification supervisées à savoir Decision Tree (J48), Support Vector Machines (SVM), Multilayer Perceptron (MLP), Naive Bayes et k-Nearest Neighbor (k-NN) qui sont comparées en termes d'exactitude de classification, de précision, de rappel et de mesure. Les résultats ont montré que l'algorithme J48 léger peut être efficace et précis à 99.47%, il est utilisable pour la reconnaissance de la posture des utilisateurs assis. Un avantage de l'approche basée sur l'arbre de décision est que les modèles entraînés sont très légers et faciles à mettre en œuvre sur les appareils embarqués.

Roh et al. [36] ont utilisé un système de cellule de charge composé de quatre cellules de charge montées sur le reste inférieur d'une chaise. Divers algorithmes d'apprentissage automatique ont été appliqués au rapport de poids corporel mesuré par le système développé pour identifier la méthode qui classait le plus précisément la posture assise réelle de la personne assise. L'exactitude de la classification a été calculée à l'aide des données du test, qui contiennent les 84 changements de posture assise.

DÉTECTION DE MAUVAISE POSTURE PAR MACHINE LEARNING

Objet	SVM _{Rbf}	SVM _{lin}	LDA	QDA	NB	RF	DT
1	0.9736	0.7650	0.8197	0.8502	0.7642	0.9182	0.6009
2	0.9777	0.8579	0.8830	0.8774	0.7618	0.9443	0.7145
3	0.9678	0.8860	0.9020	0.9035	0.9006	0.9327	0.8728
4	0.9722	0.8944	0.9042	0.9042	0.8806	0.9361	0.8125
5	0.9721	0.8830	0.9164	0.9206	0.8733	0.9415	0.7604
6	0.9624	0.8790	0.8887	0.8915	0.7928	0.9263	0.6787
7	0.9721	0.8607	0.9011	0.9248	0.8482	0.9248	0.8301
8	0.9794	0.8971	0.9000	0.9191	0.9235	0.9515	0.9176
9	0.9705	0.8215	0.8555	0.8687	0.8451	0.9100	0.7153
Moyenne	0.9720	0.8627	0.8956	0.8956	0.8433	0.9317	0.7679

FIGURE 3.1 – Tableau :Taux de classification des données d'essai selon le classificateur dans chaque sujet [36]

La figure 3.1 montre l'exactitude de la classification de chaque classificateur pour les données d'essai, la précision moyenne de la classification variait de 76,79% à 97,20%. De plus, le tableau ci-dessus montre que le taux de classification de la méthode de l'arbre de décision était le plus faible. Le taux de classification de la SVM utilisant le noyau RBF variait de 96,31% à 97,94%. La précision du classificateur de forêt aléatoire avec 30 arbres était la plus proche de celle de la SVM utilisant le noyau RBF, ne montrant qu'une légère différence.

Le système proposé a pu classer six postures assises, y compris la posture avec chargement sur le dossier, et a montré la possibilité de classer la posture assise même si le nombre de capteurs est réduit.

K. Bourahmoune et T.Amagasa [37] ont mis en place un coussin intelligent appelé LifeChair pour collecter des données relatives à la posture en temps réel. Le LifeChair est un coussin intelligent pour le devant la chaise qui utilise une nouvelle technologie de détection de pression spécialement développée pour la détection de la posture humaine.

DÉTECTION DE MAUVAISE POSTURE PAR MACHINE LEARNING

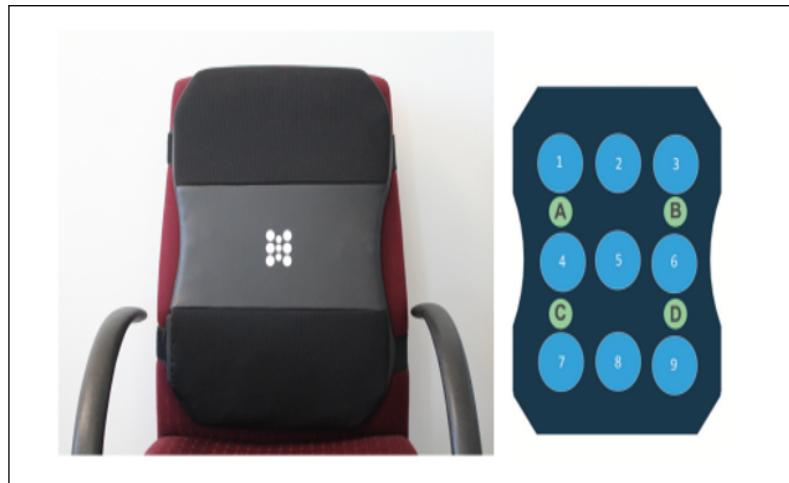


FIGURE 3.2 – Un coussin intelligent entièrement intégré pour le devant de la chaise[37]

La figure 3.2 illustre le coussin ‘LifeChair’ qui utilise neuf capteurs de pression répartis sur une interface coussin réalisée de cuir et de maille dans des endroits bien définis numérotés de 1 à 9, la figure montre également la position des quatre actionneurs de vibration (A-D) qui fournissent un retour haptique pour la correction de la posture.

En utilisant différents modèles de vibration ,le coussin intelligent communique à l’utilisateur et lui envoie des informations spatiales, comme comment corriger leur posture, Le type de données générées par le système LifeChair est caractéristiquement et temporellement pertinent pour l’application de l’apprentissage automatique pour l’automatisation et l’amélioration des performances dans des tâches telles que la reconnaissance de la posture, la classification des poses d’étirement et les méthodes basées sur les habitudes .

En utilisant les données du capteur et données corporelles saisies par l’utilisateur ils ont atteint 98,93% précision dans la détection de plus de 13 postures assises différentes à l’aide d’un modèle d’apprentissage machine supervisé « Random Forest» qui est facilement déployable adapté aux implémentations ML basées sur les mobiles et le cloud.

P. Rosero-Montalvo et al. [38] présentent une approche basée sur un système embarqué pour acquérir des variables liées à la position et des techniques d’apprentissage, à savoir la réduction de dimensionnalité (DR) et la classification. Puisque les informations acquises par les capteurs sont de grande dimension et peuvent donc ne pas être enregistrées dans la mémoire

DÉTECTION DE MAUVAISE POSTURE PAR MACHINE LEARNING

système intégrée, pour cette raison, le système a une étape réduction de dimensionnalité basée sur le principe d'analyse des composants est effectuée. Par la suite, la détection de la posture est effectuée par le classificateur k-plus proches voisins (KNN) entre la matrice stockée dans le système et les nouvelles données acquises par les capteurs de pression et de distance. Ainsi, en ce qui concerne l'utilisation de l'ensemble des données, le coût de calcul est diminué de 33% ainsi que la lecture des données est réduite de 10 ms. Ensuite, la tâche de détection de pose assise prend 26 ms et atteint 75% de précision.

Vikas Kumar Singha et al.[42] ont utilisé un capteur inertiel pour surmonter les comportements assis humains malsains de l'employé de bureau. Pour surveiller, six volontaires sont considérés dans la tranche d'âge de 26 ± 3 ans. Le capteur inertiel est fixé en haut du tronc arrière du corps et des données sont générées pour cinq activités différentes effectuées par les utilisateurs assis sur la chaise du bureau. La technique de sélection d'entités basée sur la corrélation et les méthodes d'optimisation de l'essaim de particules sont utilisées conjointement pour sélectionner des vecteurs d'entités. Les fonctionnalités optimisées sont transmises à des classificateurs super-visés par l'apprentissage automatique tels que Bayes naïfs, SVM et KNN pour reconnaissance. Enfin, le classificateur SVM a atteint une précision globale de 99,90% pour différents comportements assis humains à l'aide d'un accéléromètre, d'un gyroscope et de capteurs magnétométriques.

Jawad ahmed et al.[41] présentent un système de reconnaissance gestuelle conçu pour détecter la posture assise d'un utilisateur de fauteuil roulant. L'objectif principal du système proposé est d'identifier les postures irrégulières et inappropriées afin de prévenir les problèmes de santé associés à la position assise. Dans le système de surveillance proposé, un réseau de 16 capteurs de pression sérigraphiés est utilisé pour acquérir des données de pression, qui sont échantillonnées et traitées en temps réel à l'aide d'une électronique de lecture. La reconnaissance de posture a été effectuée pour quatre positions assises.

DÉTECTION DE MAUVAISE POSTURE PAR MACHINE LEARNING

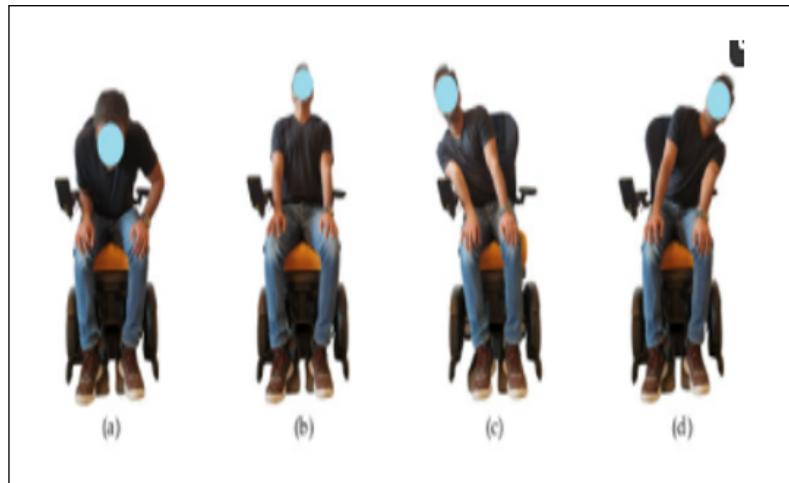


FIGURE 3.3 – Collecte de données d’entraînement avec différentes positions assises.[41]

La figure 3.3 illustre les différents positions assises utilisées pour la collecte de données du Training Test , (a) représente une inclinaison vers l'avant; b) une inclinaison vers l'arrière, c) une inclinaison à droite, d) une Tendance à gauche.

Ils se sont basés sur les algorithmes de classification k-plus proches voisins (k-NN), machines à vecteurs de support (SVM), forêts aléatoires (Random Forest), arbres de décision (Decision Tree) et apprentissage automatique Light gradient boosting machine (LightGBM).

Les classificateurs sont formés sans aucune réduction de dimensionnalité.

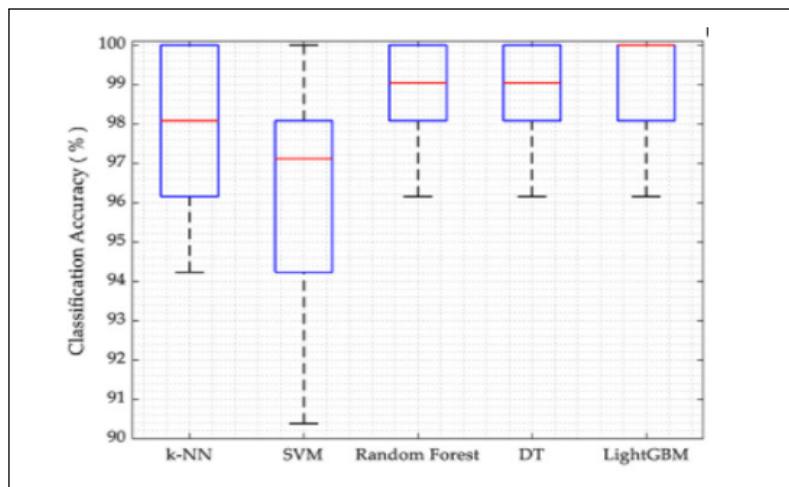


FIGURE 3.4 – Graphe de précision pour les algorithmes d’apprentissage automatique.[41]

La figure 3.4 illustre sous forme de diagramme les précisions moyennes des cinq classificateurs K-NN, SVM, RF, DT, LightGBM. Comme on peut l’observer l’algorithme k-NN

a fourni une précision moyenne suffisamment meilleure pour la classification de la posture que le SVM qui est le moins précis parmi les autres classificateurs en raison de sa tendance à la classification binaire. La précision de DT est à celle de RF et sont supérieur à celle de K-NN. Cependant Light GBM a obtenu les meilleurs résultats.

3.3 Application d'approches Deep Learning pour classer la posture assise

Divers algorithmes d'apprentissage automatique ont été appliqués pour reconnaître les activités et les modèles humains à l'aide de plusieurs types de capteurs, bien que ces algorithmes présentent des résultats satisfaisants dans certains environnements, ils souffrent de plusieurs limitations. Par exemple, les algorithmes nécessitent des extractions de caractéristiques artisanales pour un apprentissage efficace et ne peuvent apprendre que des caractéristiques superficielles. Les progrès récents des algorithmes d'apprentissage en profondeur peuvent fournir des solutions potentielles à ces problèmes. Voici quelques solutions de classification de posture assise basées sur le deep learning.

Kim min et al [39] ont développé un système de classification des postures assises pour les enfants à l'aide d'algorithmes d'apprentissage automatique. L'algorithme du réseau neuronal convolutif (CNN) a été utilisé en plus des algorithmes conventionnels : classificateur de Bayes naïf (NB), arbre de décision (DT), réseau neuronal (NN), régression logistique multinomiale (MLR) et machine vectorielle de support (SVM). Pour recueillir des données pour classer les postures assises, un coussin de détection a été développé en installant un tapis de capteur de pression (8×8) à l'intérieur du coussin du siège de la chaise pour enfants. Dix enfants ont participé, et les données des capteurs ont été recueillies en prenant une posture statique pour les cinq postures prescrites. La précision de CNN s'est avérée être la plus élevée par rapport à celles des autres algorithmes.

DÉTECTION DE MAUVAISE POSTURE PAR MACHINE LEARNING

Kulikajevas et al[43] proposent nouveau modèle de réseau hiérarchique récurrent profond qui est une extension du réseau neuronal MobileNetV2.

Ce modèle permet l'utilisation des données vidéo séquentielles comme entrée, permettant ainsi au réseau neuronal profond d'extraire des caractéristiques temporelles importantes des images vidéo, qui seraient autrement perdues par rapport à une classification à une seule image tout en étant capable de prédiction d'une seule image en raison de son biais vers la dernière image. Ils ont amélioré la couche supérieure de l'architecture MobileNetV2 en ajoutant la représentation hiérarchique des données, qui agit comme un verrou sémantique pour la classification des étiquettes de niveau supérieur en filtrant rapidement les étiquettes de classe non valides.

De plus, ils ont réalisé une étude pilote basée sur laquelle ils suggèrent la méthodologie requise pour collecter l'ensemble de données de formation et les ensembles de données de validation. L'approche de classification de posture proposée est très extensible en raison de sa représentation d'arbre aplatie, qui peut être facilement adaptée aux tâches de classification de posture déjà existantes, la profondeur du modèle de posture sémantique ontologique étant l'un des facteurs déterminants de la qualité de la classification. Sur la base de leurs données de validation, ils ont une précision de classification de 91,47% dans la prédiction de trois classes principales de posture assise (posture arrière, posture avant et posture droite) à un taux de 10 fps.

Enfin, contrairement aux travaux connexes leur méthode ne dépend pas des prédicteurs squelettiques, ils ont pu donc effectuer une prédiction de la posture humaine assise lorsque seulement 30% du torse humain est visible dans le cadre.

Jang et al [44] ont proposé une méthode de surveillance simultanée de la mauvaise posture, y compris la posture de la tête vers l'avant (FHP), l'épaule arrondie (RS) et l'épaule surélevée (ES). Ces postures et les symptômes qui en résultent sont de plus en plus répandus, et une analyse complète, simultanée et approfondie de ces troubles de la posture est nécessaire.

Cette méthode consiste à recueillir des données de posture à partir d'une nouvelle combinaison d'accéléromètres et de magnétomètres jumelés à des aimants miniatures. Les emplacements des capteurs sont choisis de manière optimale pour calculer de manière fiable les angles du cou et

DÉTECTION DE MAUVAISE POSTURE PAR MACHINE LEARNING

des épaules représentant l'angle cranio-vertébral (CVA) pour FHP, l'angle d'épaule avant (FSA) pour RS et l'angle de symétrie (SA) pour ES.

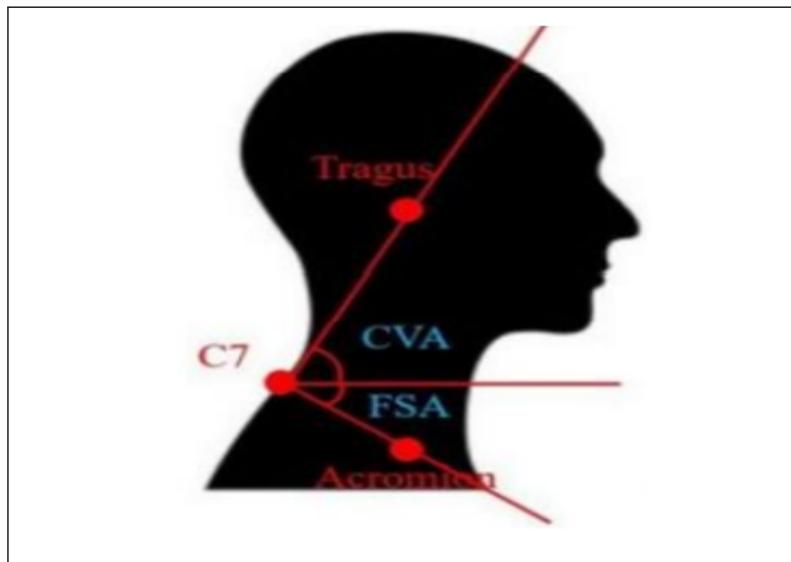


FIGURE 3.5 – vue conceptuelle de CVA et FSA.[44]

La figure 3.5 illustre une vue conceptuelle de CVA et FSA ; les mots en rouge désignent des parties du corps humain à savoir le Tragus, C7 et acromion. Le CVA est l'angle entre une ligne horizontale et la ligne reliant le tragus et l'apophyse épineuse C7. FSA est l'angle entre une ligne horizontale et la ligne reliant la vertèbre C7 et l'acromion.

Le traitement des données de capteur collectées est réalisé par des algorithmes de réseau neuronal profond (DNN) et de réseau neuronal convolutif (CNN).

Les résultats expérimentaux démontrent une classification réussie des mauvaises postures avec une grande précision (DNN : 88,1%, CNN : 88,7%) même avec l'analyse simultanée de FHP, RS et ES.

DÉTECTION DE MAUVAISE POSTURE PAR MACHINE LEARNING

Section	Meilleure Technique D'apprentissage	Type de Capteurs	Solution	Informations fournies Par capteurs	Placement Des capteurs	Avantages	Inconvénients	Precision
Machine Learning	Random Forest	Capteurs de force et d'accélération	Zemp et al.	angles d'inclinaison	dans le dossier et les coussins de la casserole du siège ainsi que dans les accoudoirs	Coût minimal Moins complexe	Non portable	90,9%
		Capteurs de pression	K. Bourahmoune et T.	Informations spatiales	répartis sur une interface coussin	Efficace	Non portable	98,93%
	Support Vector Machines (SVM)	Capteurs de pesage	Roh et al	Information Sur le poids	une charge sur la plaque de dossier, avec quatre capteurs montés uniquement sur la plaque de siège	Efficace	Non Portable, complexe	96,31% à 97,94%
		Capteur inertiel	Vikas Kumar Singha et al.	Courbure de la colonne vertébrale	fixé en haut du tronc arrière du corps	faible coût Le jeu de données sur les comportements assis est auto-généré dans le présent article	complexe	99%
	k-Nearest Neighbor (k-NN)	Capteurs de pression	P. Rosero-Montalvo et al.	angle d'inclinaison	Installés à l'intérieur du siège de la chaise	Coût de calcul faible Vitesse de lecture des données minimale.	charge de calcul	75%
	Light GBM		Jawad ahmed et al.	Des données de pression	intgré dans un coussin de siège de fauteuil roulant	appareil portable léger de surveillance de la santé	le programme de surveillance de la pression s'exécute manuellement	99,8%
	Decision Tree (J48),		Ma et al	l'indice de masse corporelle	montés sur le repose-pied et le dossier d'un fauteuil roulant	les modèles entraînés sont très légers et faciles à mettre en œuvre sur les appareils embarqués	Non Portable, beaucoup de capteurs	99,47%
Deep Learning	Réseau de neurones convolutif (CNN)	Capteurs de pression	Kim min et al	Information Sur la taille	à l'intérieur du coussin de siège de la chaise	concentrés sur les enfants	Non Portable, Installation compliquée	/
	réseau de neuronal artificiel (ANN)	Caméra de profondeur	Kulikajevas et al	Informations de couleur et de profondeur	Camera placée en face à l'utilisateur	Ne dépend pas des prédicteurs squelettiques	menace la vie privée. un petit nombre de sujets	91,74%
	Réseau Neuronal Profond (DNN) Et Convolutif (CNN).	Accélémètre et magnétomètre	Jang et al	Angle d'inclinaison	l'angle crano-vertébral (CVA) pour FHP, l'angle d'épaule avant (FSA) pour RS et l'angle de symétrie (SA)	Meilleur choix en matière de portabilité	Position du capteur trop précise, difficile à implémenter	DNN : 88,1 CNN : 88,7

TABLE 3.1 – Tableau récapitulatif des études précédentes de la détection de la mauvaise posture par le machine learning et le deep learning

3.4 CONCLUSION

Dans cette section, nous avons présenté quelques solutions déjà existantes dans la littérature concernant la détection de la posture humaine assise, et sa classification à l'aide de l'intelligence artificielle à savoir les algorithmes de machine Learning tel que KNN, SVM, DT, RF, NF, RML, qui ont montré des résultats satisfaisants et les algorithmes de deep learning tel que CNN et DNN. Enfin un tableau comparatif a été réalisé afin de permettre d'avoir une vue globale ou un résumé sur l'intégralité du chapitre. Le prochain chapitre sera consacré à notre proposition concernant la conception du système de détection et d'alerte de mauvaise posture assise.

Conception générale du système

Sommaire

4.1	INTRODUCTION	44
4.2	Architecture du système	45
4.3	Collecte de données à l'aide des capteurs EMG	46
4.3.1	Choix du capteur EMG	46
4.4	Traitemet des données avec microcontrôleur	51
4.4.1	choix du microcontroleur	52
4.4.2	Convertisseur analogique-numérique à approximations successives SAADC	55
4.5	Classification de la posture assise	56
4.6	Application mobile	57
4.7	CONCLUSION	58

4.1 INTRODUCTION

Comme présenté précédemment la plupart des systèmes de détection de mauvaise posture assise utilisent des technologies qui les rends moins pratiques à utiliser dans notre vie quotidienne, parmi ces dernières l'utilisation des caméras qui menacent la vie privée de l'individus, l'utilisation d'une chaise spéciale qui rend le système non portable. Les systèmes utilisant le capteur EMG étaient trop compliqués a implémenté en effet ils utilisent un grand nombre de capteurs placées à des endroits trop précis alors que les résultats n'étaient pas très satisfaisants où chaque changement d'emplacement influence négativement sur tout le système. Cela nous a motivé à concevoir un système simple et facile à réaliser et efficace au même temps. Ce chapitre consacré à la conception de cette solution. Dans ce qui suit, nous présentons brièvement les différentes étapes de la conception en exposant les composants et leurs fonctionnalités.

4.2 Architecture du système

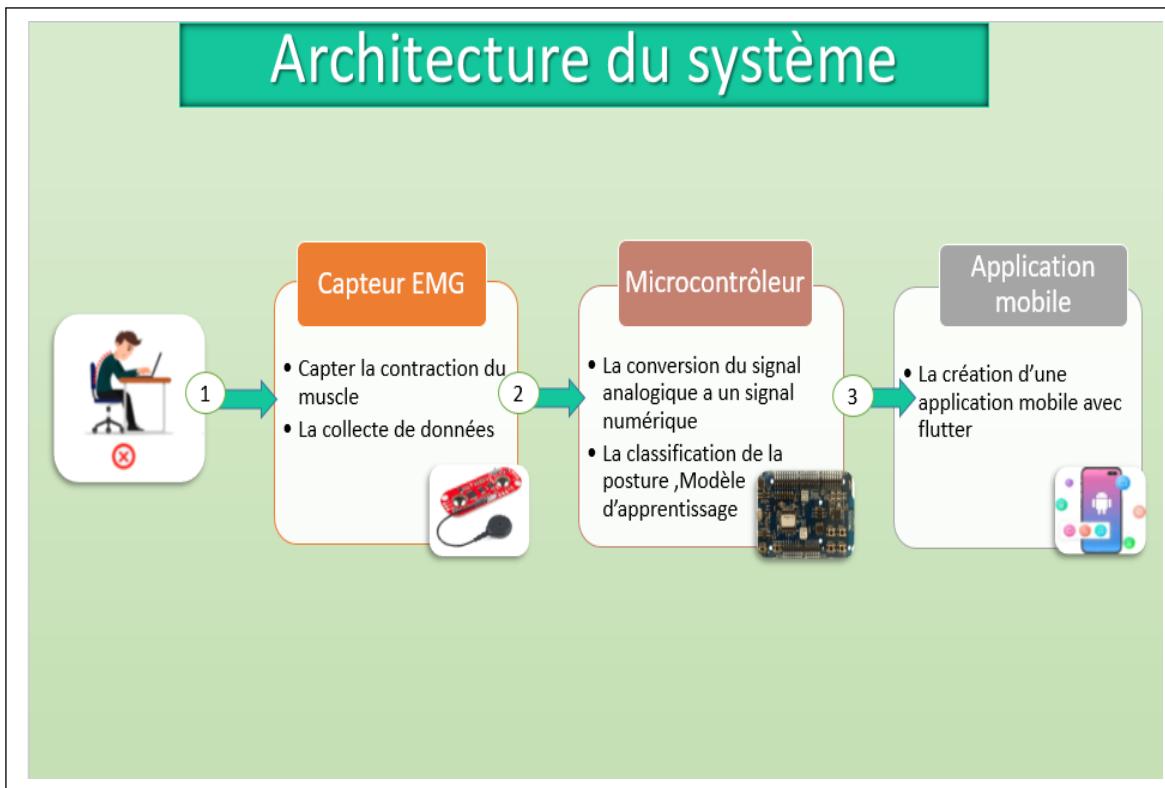


FIGURE 4.1 – Architecture de la solution proposé

L'architecture de notre solution illustrée dans la figure 4.1 se compose de trois composants physiques essentiels à savoir un capteur EMG, un microcontrôleur, une application mobile implémentée sur un téléphone portable. Tout d'abord, le capteur EMG placé sur le dos d'une personne capte le signal représentant la contraction musculaire. Ce capteur sera branché à un microcontrôleur, par conséquent les signaux vont être envoyés vers ce dernier.

Ensuite, le signal analogique capté sera converti en un signal numérique au niveau du microcontrôleur.

La phase suivante consiste à utiliser les données numériques comme entrées d'un modèle d'apprentissage qui classifiera l'état de la posture assise.

Enfin le résultat contenant le type de la posture assise sera transmis vers une application mobile qu'on doit développer, cette dernière va servir comme système de rétroaction et d'alerte qui

envoie des notifications à l'utilisateur lorsqu'il se trouve dans une position inadéquate. Dans les sections qui suivent, on va détailler chaque composant du système proposé.

4.3 Collecte de données à l'aide des capteurs EMG

Electromyogramme EMG

L'électromyogramme, a pour objectif d'analyser l'influx nerveux dans les nerfs moteurs, les nerfs sensitifs et les muscles. C'est un examen clef en neurologie qui permet d'évaluer le fonctionnement des nerfs et des muscles.[65]

les capteurs EMG permettent d'enregistrer l'activité électrique des nerfs ainsi que la contraction d'un muscle soit en piquant une aiguille dans le muscle ou à côté du nerf, soit en collant une électrode sur la peau si le nerf ou le muscle sont superficiels. L'activité électrique est analysée au repos, après une stimulation électrique artificielle ou après un effort de contraction volontaire du patient.[65]

4.3.1 Choix du capteur EMG

Avant de réaliser notre système qui permet de récolter les données, nous commençons par mener une étude comparative des capteurs qui nous permettent de mesurer la contraction des muscles à partir des signaux EMG .

4.3.1.1 Description de quelques capteurs EMG

Kit de développement de capteur musculaire MyoWare KIT-14409 ROHS

La carte MyoWare incluse dans ce kit agit en mesurant l'activité électrique filtrée et rectifiée d'un muscle produisant 0-5 Volts en fonction de la quantité d'activité dans le muscle sélectionné, où Vs signifie la tension de la source d'alimentation. Il est très simple de brancher ce kit, d'attacher des boucliers au capteur musculaire, de coller quelques électrodes incluses, de lire la tension et de fléchir certains muscles.

Ce kit comprend également tout les outils nécessaires pour configurer le capteur musculaire

CONCEPTION GÉNÉRALE DU SYSTÈME

MyoWare : tous les boucliers associés (alimentation, câble, LED et Proto), les coussinets et câbles de capteur biomédical, ainsi que les en-têtes qui peuvent être soudés sur chaque bouclier afin qu'ils puissent être fixés ensemble.

Pour faciliter l'empilement, le kit de développement de capteurs musculaires MyoWare comprend des en-têtes empilables à 3 broches. L'utilisation de ceux-ci sur tous les boucliers permet un empilement facile dans plus de configurations.

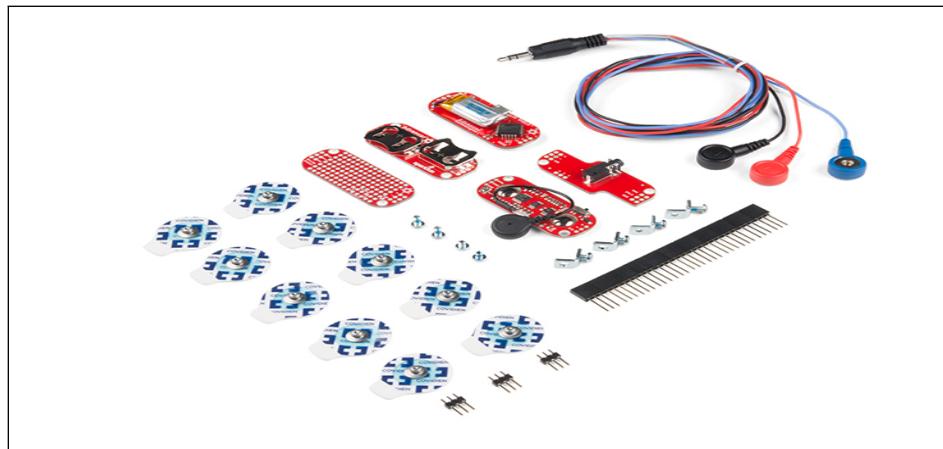


FIGURE 4.2 – Kit de développement de capteur musculaire MyoWare-14409 ROHS [52]

Le capteur musculaire MyoWare permet d'attacher des coussinets de capteurs biomédicaux directement à la carte elle-même en se débarrassant de ces câbles embêtants. Cette carte comprend également une multitude d'autres nouvelles fonctionnalités, notamment une tension d'alimentation unique de + 3,1 V à + 5 V, une sortie RAW EMG, des broches d'alimentation protégées contre la polarité, des voyants LED et un interrupteur marche/arrêt. De plus, des boucliers tel que des câble et une alimentation peuvent se fixer au capteur musculaire Myoware pour aider à augmenter sa polyvalence et sa fonctionnalité.

La mesure de l'activité musculaire par détection de son potentiel électrique, appelée électromyographie (EMG), est traditionnellement utilisée pour la recherche médicale. Cependant, avec l'avènement de microcontrôleurs et de circuits intégrés toujours plus petits et plus puissants, les circuits et capteurs EMG ont trouvé leur place dans toutes sortes de systèmes de contrôle [52].

CONCEPTION GÉNÉRALE DU SYSTÈME

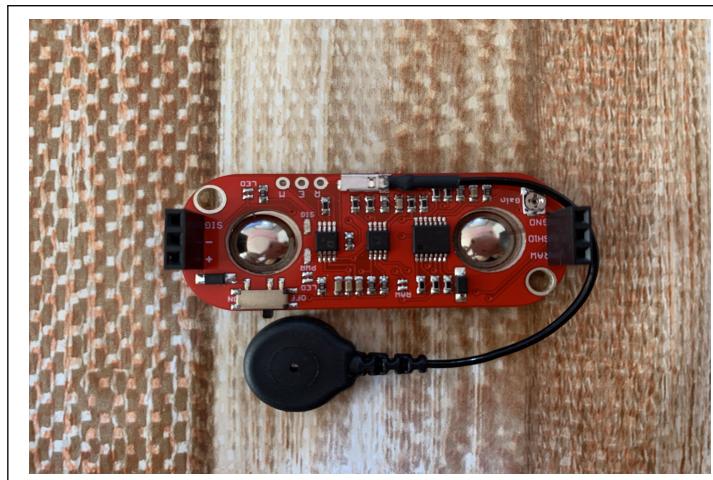


FIGURE 4.3 – Capteur musculaire électrique Sparkfun MyoWare - SEN-13723 (réf SSBIMU4)

Capteur EMG SEN0240

Capteur d'électromyogramme de DFRobot et OYMotion pour détecter l'activité musculaire et nerveuse chez un être humain. Le capteur est livré avec un module amplificateur pour le raccordement à un microcontrôleur compatible Arduino. Ce type de module peut être utilisé en interaction entre un humain et un ordinateur. Il se connecte à une entrée analogique d'une carte compatible Arduino ou directement au shield Gravity I/O via le câble fourni[53].

- Caractéristiques[53] :

Alimentation : 3,3 à 5,5 Vdc, Plage de détection : +/- 1,5 mV, Tension de sortie : 0 à 3 Vcc, Température de fonctionnement : 0 à 50°C.

- Dimensions [53] :

Amplificateur : 22 x 35 mm, Capteur : 22 x 35 mm.

- Poids du capteur : 36 g

- Remarque[53] :

Ce produit ne doit en aucun cas être utilisé dans des applications médicales.

Il est recommandé d'étalonner le capteur avant chaque mesure.

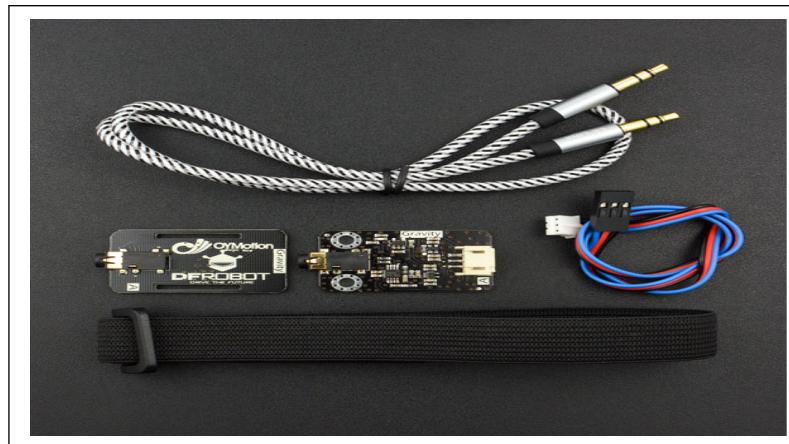


FIGURE 4.4 – Capteur EMG SEN0240

EMG Grove Detector 101020058

Ce détecteur EMG compatible Grove permet de mesurer les courants qui accompagnent l’activité musculaire et d’interfacer cette activité musculaire avec un Arduino. En mode veille la tension de sortie est de 1,5 VDC. Lorsque le muscle est activé, la tension augmente pour atteindre 3,3 VDC maximum.

- Livré avec 6 électrodes[54].

Ce module est connecté à une entrée analogique du Grove Base Shield ou du Mega Shield via un Câble 4 conducteurs inclus, interface compatible Grove[54].

-Alimentation : 3,3 ou 5 Vdc [54].

-Longueur du cordon : 1 mètre [54].

-Électrodes : 6 [54].

-Dimensions modules : 42 x 25 x 11 mm[54].

-Référence Seeedstudio : 101020058 (remplace 811006001) [54]

Remarque : Nous ne pouvons pas utiliser ce produit dans des applications médicales ou de sécurité. [54].

CONCEPTION GÉNÉRALE DU SYSTÈME



FIGURE 4.5 – EMG Grove Detector 101020058

Capteur de surface EMG DataLITE

Disponibles en tant qu'options sans fil ou filaires, les deux capteurs sont uniques à leurs conceptions en tant que l'impédance d'entrée est de > 100 Mohms, cela signifie que peu ou pas de préparation de la peau est nécessaire et sans avoir besoin de gels conducteurs.[55]

Le capteur EMG sans fil DataLITE est petit, léger et utilise des algorithmes de saut de fréquence progressif, de détection d'erreur et de récupération de données, permettant l'activité musculaire les lectures doivent être fluides et robustes, même dans un environnement de travail typique avec une plage allant jusqu'à 30 mètres de son récepteur [55]



FIGURE 4.6 – Capteur de surface EMG DataLITE

4.3.1.2 Comparaison des capteurs EMG

Capteurs	MyoWare	Capteur EMG SEN0240	EMG Grove Detector 101020058	Capteur de surface EMG sans fil DataLITE	Capteur de surface EMG filaire DataLITE
Vendeurs	Sparkfun	DFRobot and OYMotion	Seed	Biometrics Ltd	Biometrics Ltd
Prix(euro)	37.95- 56.10	40,90	32,80	52,99€	46,90
Poids	26g	36g	19g	17g	5g
Microcontrôleur Pris en charge	N'importe quel microcontrôleur	Arduino	Arduino, Raspberry Pi	N'importe quel microcontrôleur	N'importe quel microcontrôleur
Tension d'alimentation	2.9V-5.7V	+3.3V- 5.5V	3.3V-5V	+/- 6 mV	+/- 3 mV
Cable	Câble extensible	50 Cm	Cables de 1 m	Portée jusqu'à 30m	Câble de 1,25 m
Electrodes de surface jetables	3 électrodes de surface jetables	Une électrode sèche en métal	6 électrodes de surface jetables	Intégrale à sec réutilisable	Intégrale à sec réutilisable

FIGURE 4.7 – Tableau comparatif des capteurs EMG [52][53][54][55]

Résultat de la comparaison

Après avoir visualisé les principales propriétés de chaque capteur, notre choix est porté sur le capteur MyoWare à cause de sa disponibilité, son adaptation avec tout type de microcontrôleurs, son prix raisonnable, son poids léger, sa capacité à répondre aux exigences de notre projet.

4.4 Traitement des données avec microcontrôleur

Le microcontrôleur est un circuit intégré , conçu pour réaliser une opération précise dans un système intégré. Il comprend un processeur, une mémoire et des périphériques d'entrée .

4.4.1 choix du microcontrôleur

4.4.1.1 Exemples de microcontrôleurs envisagés avec description

ARDUINO Uno

L'Arduino Uno est une carte microcontrôleur open-source basée sur le microcontrôleur Microchip AT mega328P et développée par Arduino.cc. La carte est équipée d'ensembles de broches d'entrée/sortie (E/S) numériques et analogiques pouvant être interfacées avec divers extensions de cartes et d'autres circuits. La carte dispose de 14 broches d'E/S numériques (six capables de sortie PWM), 6 broches d'E/S analogiques, et est programmable avec l'IDE Arduino

(environnement de développement intégré), via un câble USB de type B. Il peut être alimenté par le câble USB ou par un 9 volts de batterie externe, bien qu'elle accepte des tensions comprises entre 7 et 20 volts[56].

Le mot "uno" signifie "un" en italien et a été choisi pour marquer la sortie initiale du logiciel d'Arduino. La carte Uno est la première d'une série de cartes Arduino basées sur USB[56].



FIGURE 4.8 – Carte Arduino [60]

STM32

Le STM32 ne fait pas référence à un microcontrôleur spécifique. C'est le nom que ST-Microelectronics a donné à sa famille de microcontrôleurs ARM Cortex-M. La famille STM32

CONCEPTION GÉNÉRALE DU SYSTÈME

elle-même peut être divisée en plusieurs groupes, qui à leur tour peuvent être divisés en séries[57].

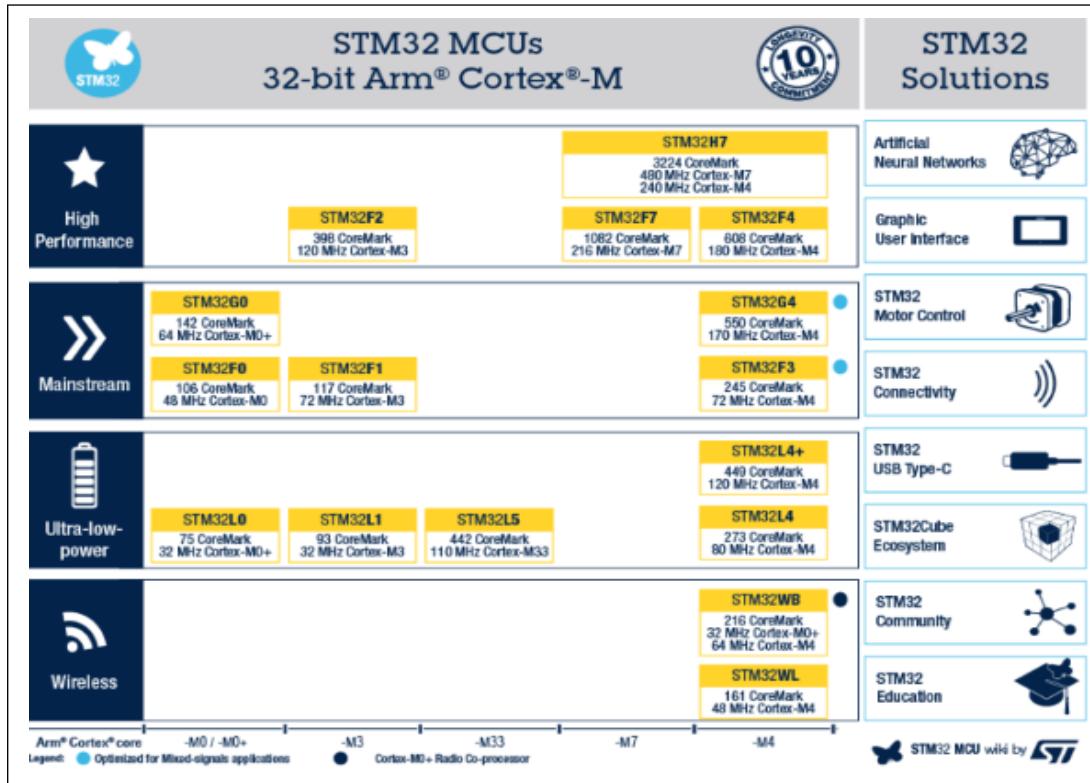


FIGURE 4.9 – Le classement des séries STM selon leurs groupes STM32 [63]

Kit de développement Bluetooth Low Energy pour les séries Nrf52

Le nRF52 DK est un kit de développement monocarte polyvalent pour le développement propriétaire

taire Bluetooth Low Energy, Bluetooth Mesh, NFC, ANT et 2,4 GHz sur les SoC nRF52805, nRF52810 et nRF52832. Il facilite le développement en exploitant toutes les fonctionnalités des nRF52805, nRF52810 et nRF52832 SoC. Il comprend une antenne NFC qui permet d'utiliser rapidement le périphérique d'étiquette NFC-A sur le nRF52832. Tous les GPIO sont disponibles via des connecteurs et des en-têtes de bord, et 4 boutons et 4 LED simplifient la sortie et l'entrée depuis et vers le SoC[58].

Il est compatible avec Arduino Uno Revision 3, il est livré avec un débogueur SEGGER J-Link intégré permettant la programmation et le débogage à la fois du SoC embarqué et des SoC externes via l'en-tête de débogage[58].

CONCEPTION GÉNÉRALE DU SYSTÈME

Il peut être alimenté par USB, mais comprend également un support de pile CR2032, permettant des tests sur le terrain de prototypes. SEGGER Embedded Studio, Keil, GCC, les IDE et les chaînes d'outils sont pris en charge[58].

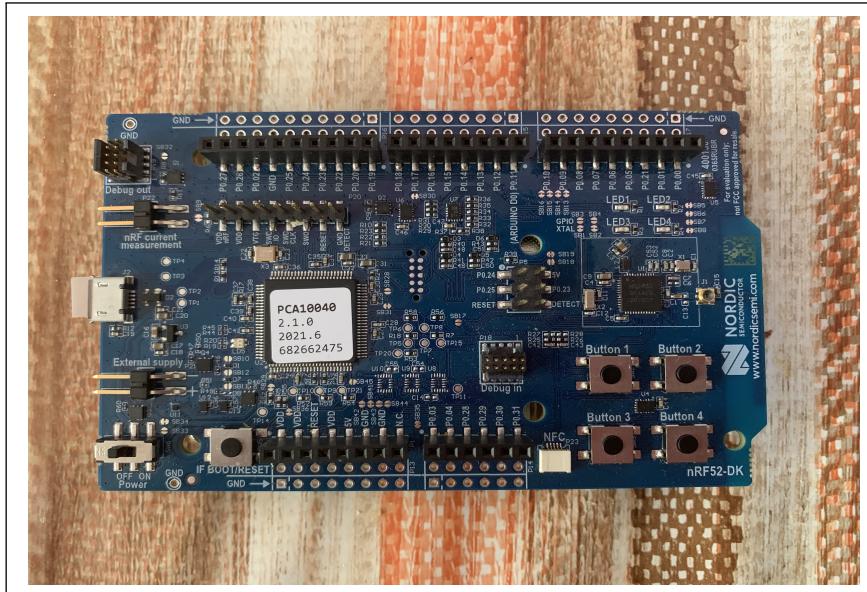


FIGURE 4.10 – carte nRF52

4.4.1.2 Comparaison des microcontrôleurs

	Arduino Uno (ATMega328)	STM32 F303VCT6	NRF52-DK nRF52832
Fréquence	16MHz	72 MHz	2.4 GHz
Type et taille de la mémoire du programme	Flash : 32 Ko	Flash: 256 kB	Flash: 512/256 Ko
Type et taille de la RAM	SRAM :2Kb	SRAM: 40kB	RAM: 64/32 Ko
Tension	5V	2V à 3.6V	1.7 V à 3.6 V
Prix(euro)	19.5	28,48	33,03
Poids	25 g	200 g	40 g
Taille	68,6x53,4 mm	200x180 mm	63x100 x15mm

FIGURE 4.11 – Tableau comparatif des microcontrôleurs[60] [61] [58] [62]

4.4.1.3 Résultat de la comparaison

Après avoir comparé certains microcontrôleurs couramment utilisés, nous avons décidé d'opter pour le microcontrôleur NRF52, d'abord en raison de sa puissance et du fait qu'il s'agit d'un kit Bluetooth à faible consommation d'énergie de plus qu'il est de petite taille et léger. Pour programmer notre microcontrôleur nrf52, nous avons besoin de KEIL comme environnement de développement, c'est la solution de développement logiciel la plus complète pour les microcontrôleurs Arm et comprend tous les composants dont l'utilisateur a besoin pour créer et déboguer des applications embarquées.

4.4.2 Convertisseur analogique-numérique à approximations successives SAADC

L'ADC est un convertisseur analogique-numérique différentiel à registre d'approximations successives (SAR) [59]

-Résolution 8/10/12 bits, résolution 14 bits avec suréchantillonnage -Jusqu'à huit canaux d'entrée.

-Plage d'entrée pleine échelle (0 à VDD).

-Échantillonnage déclenché via une tâche depuis un logiciel ou un PPI (Programmable Peripheral Interconnect) canal pour une flexibilité totale sur les sources de fréquence d'échantillonnage.

-Mode de conversion unique pour échantillonner un seul canal.

-Mode de balayage pour échantillonner une série de canaux est séquentiel.

Le délai d'échantillonnage entre les canaux varie selon la configuration de l'utilisateur.

- Échantillons stockés sous forme de valeurs complémentaires de 16 bits pour l'échantillonnage différentiel et asymétrique.

4.5 Classification de la posture assise

Notre but est de classifier la position assise en utilisant les données recueillis à partir du capteur EMG comme données d'entraînement d'un modèle de classification choisis. voici un organigramme qui explique les étapes de cette procédure.

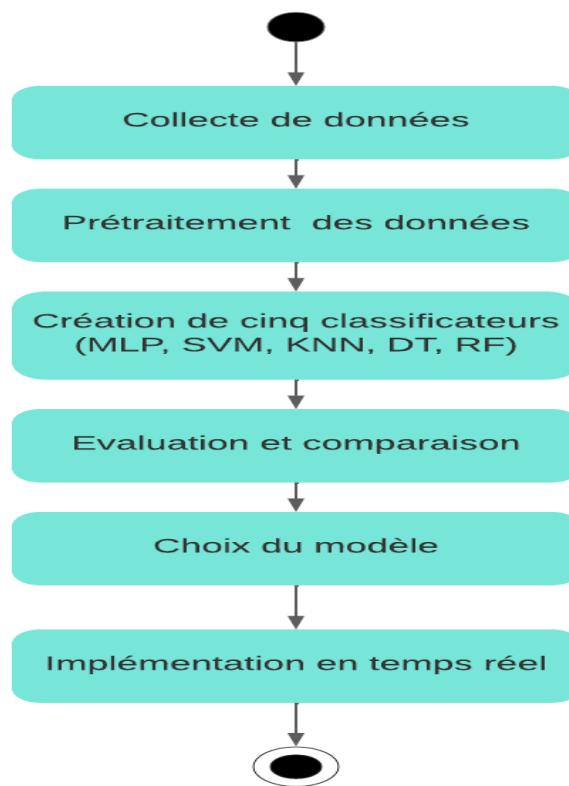


FIGURE 4.12 – Diagramme d’activité : étapes de classification de la posture

Le diagramme d'activité de la figure 4.12 montre les étapes de la classification de la posture tout d'abord une lecture de valeurs représentant la contraction musculaire d'un individu est effectué pendant 1 minute ensuite un pré-traitement est réalisé sur les données en extrayant des caractéristiques précises au lieu d'utiliser des données brutes, après cela cinq modèles d'apprentissage vont être utilisés et comparés entre eux à savoir multilayer perceptron , K-nearest neighbor, decision tree, random forest , le meilleur de ces derniers sera adopté à la fin dans le système final en temps réel. Les détails de ces modèles seront donnés dans le chapitre implémentation avec la présentation des résultats.

4.6 Application mobile

L'objectif principal de ce système est que l'utilisateur assis qui travaille dans son bureau ait une vision de sa posture en temps réel à l'aide d'une application mobile. notre application reçoit les données concernant l'état de la posture à partir du microcontrôleur puis elle alerte l'utilisateur par le biais des notifications. Ci-dessous, une liste des besoins fonctionnels concernant l'interface utilisateur de l'application mobile.

Connexion Bluetooth : le client doit activer son Bluetooth pour qu'il puisse recevoir des informations concernant l'état de sa posture de la part du microcontrôleur.

Notifications : l'utilisateur pourra recevoir des alertes en lui indiquant qu'il est dans une mauvaise position et qu'il doit la corriger. Ces notifications vont apparaître malgré que l'utilisateur soit en dehors de l'application mobile ou en mode verrouillé.

Un besoin technique : essentiel pour notre implémentation est que l'application mobile devra être développée de manière à être compatible (dans la mesure du possible) avec un maximum de versions des systèmes d'exploitation iOS et Android :

Pour Android : version 5 et plus.

Pour iOS : version 10 et plus.

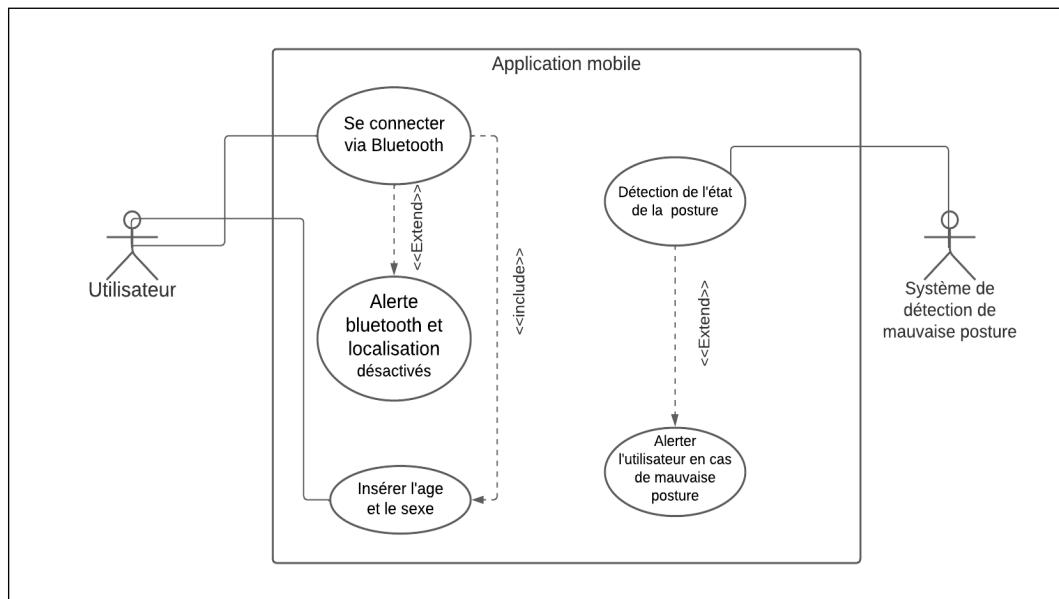


FIGURE 4.13 – Diagramme de cas d'utilisation de l'application

Le diagramme de cas d'utilisation de la figure 4.13 explique les actions que l'utilisateur doit effectuer pour que le système alerte l'utilisateur en cas de mauvaise posture. En effet, une connexion au Bluetooth et nécessaire sans oublier l'activation de la localisation, sans ces deux conditions un message d'erreur s'affichera à l'écran de l'application. De plus, une insertion de l'age et du sexe de l'utilisateur est indispensable pour l'affichage de l'état de la posture de la part du système.

4.7 CONCLUSION

Dans ce chapitre nous avons exposé brièvement l'architecture de notre solution à savoir les composants et leurs rôles, ensuite nous avons justifié le choix des outils matériels à savoir les capteurs EMG et le microcontrôleur en effectuant des recherches comparatifs sur ces derniers. De plus, nous avons présenté un aperçu sur le convertisseur analogique-numérique à approximation successive SAADC. Ensuite, nous avons expliqué la démarche suivie afin de classifier la posture à l'aide du machine learning.

Enfin nous concluons la partie avec une description des besoins techniques et fonctionnels de l'application.

Dans le chapitre qui suit nous allons présenter le travail réalisé en détaillons toutes les configurations et les résultats obtenus.

Implementation et résultats

Sommaire

5.1	INTRODUCTION	60
5.2	Architecture du système	60
5.3	Collecte de données	61
5.3.1	Test du capteur EMG	61
5.3.2	Emplacement des capteurs EMG	62
5.3.3	Assemblage du circuit	63
5.4	Implémentation du périphérique BLE	64
5.5	Classification de la posture	66
5.5.1	Données d'entraînement	66
5.5.2	Prétraitement des données	66
5.5.3	Outils de programmation	68
5.5.4	Classification binaire	69
5.5.5	Classification Multiclasse	77
5.6	Application mobile	83
5.6.1	Implémentation du central BLE	85
5.6.2	Interface utilisateur	86
5.7	Analyse des résultats	94
5.8	CONCLUSION	95

5.1 INTRODUCTION

Dans cette partie, nous allons décrire de manière concise le système réalisé en détaillant toutes les configurations et les résultats obtenus dans chaque sous-système.

5.2 Architecture du système

Après quelques recherches, on a découvert que la carte NRF522832-DK qu'on a programmé pour recevoir les signaux du capteur et les convertir en signal numérique ne bénéficie pas des fonctionnalités standard d'IA. Pour faire face à ce problème, on décide de modifier l'architecture de notre système.

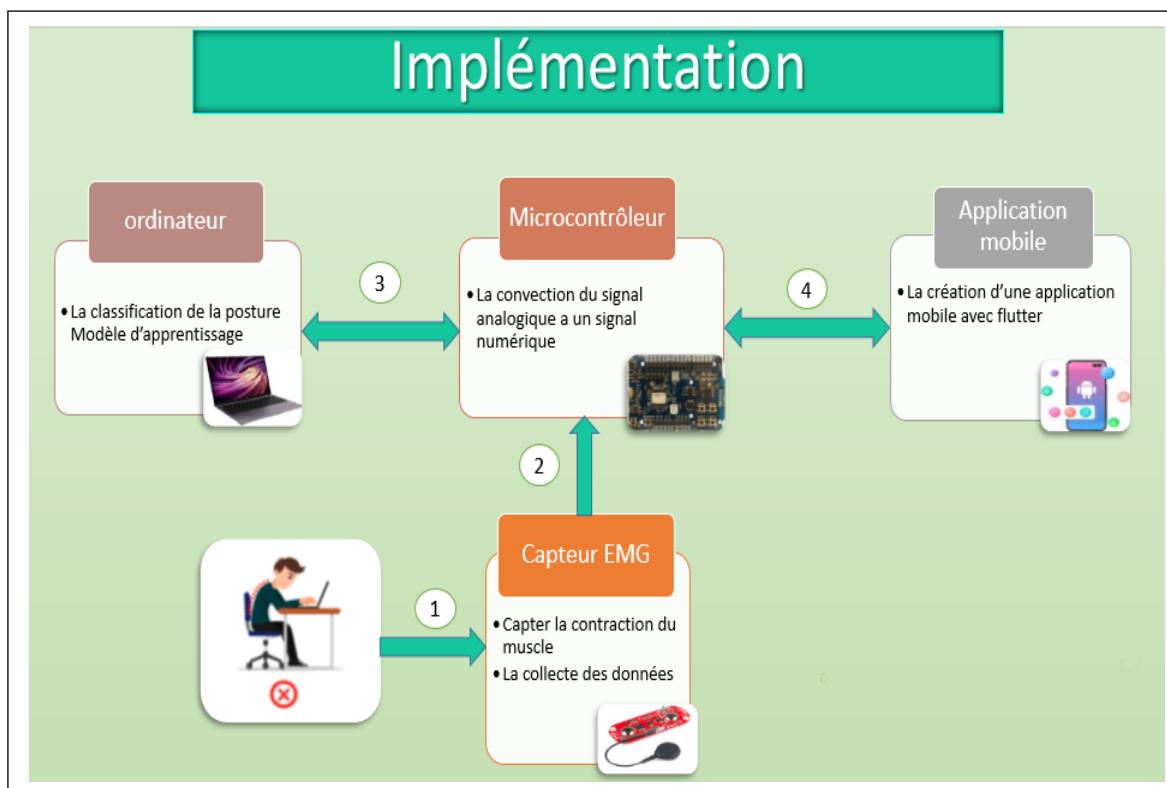


FIGURE 5.1 – implémentation du système

La figure 5.1 illustre la nouvelle architecture, la partie classification de la posture basée sur les algorithmes d'apprentissage automatique et profond sera traitée sur ordinateur portable branché avec cable USB avec le microcontrôleur. Le microcontrôleur exécute un code qui a

IMPLEMENTATION ET RÉSULTATS

pour rôle principale de convertir les signaux désignant de la contraction musculaire en données numériques(en utilisant le convertisseur SAADC) puis les transmet vers le PC via une communication UART série (lien 2 sur la figure 5.2). le resultat du machine learning sera envoyer du PC vers l'application mobile en passant par le NRF52 via communication UART série entre le PC et le NRF52 (lien 3) et via une communicatio BLE avec le service NUS entre le microcontrôleur et l'application mobile (lien 4).

5.3 Collecte de données

5.3.1 Test du capteur EMG

Tout d'abord, le capteur doit être testé s'il fonctionne bien, s'il transmet correctement les contractions d'un muscle, pour cela nous allons placer le capteur musculaire sur la main d'une personne, le capteur sera relié à LED afin de visualiser l'intensité des contractions du muscle.

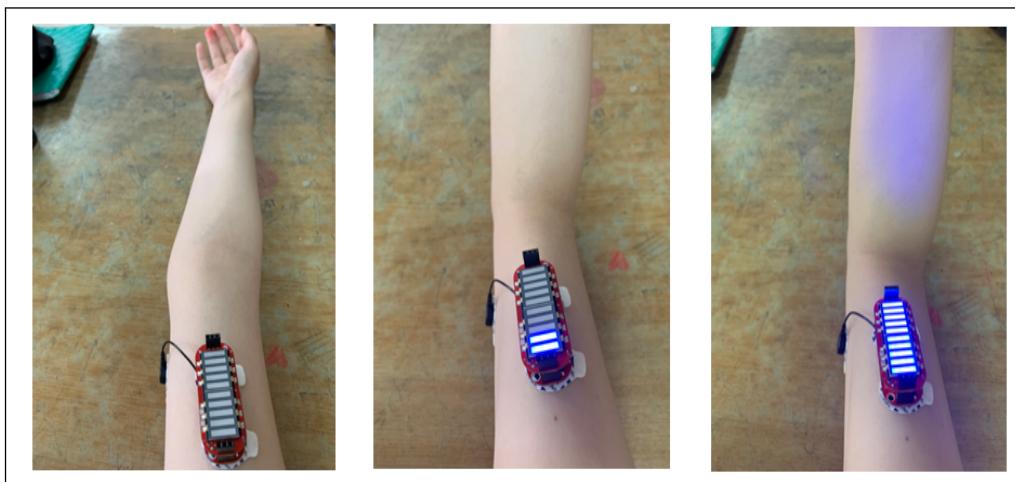


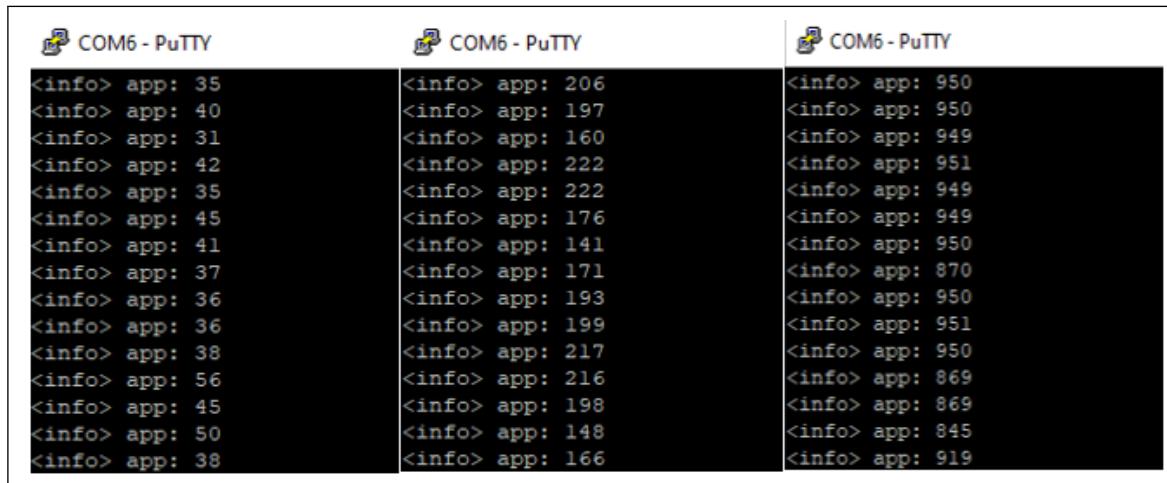
FIGURE 5.2 – Test du capteur : mode repos, début d'une contraction, contraction complète

Dans la figure 5.2 on aperçoit que lorsque le muscle est dans un état de repos les barres de la LED restent éteintes, dans le cas où le muscle est dans un état début de contractions deux à trois barres de la LED s'allument, dans le cas d'une contraction complète toute la LED est allumée.

conversion du signal

IMPLEMENTATION ET RÉSULTATS

Après avoir connecté le capteur au microcontrôleur, le convertisseur analogique-numérique à approximations successives (SAADC) convertit le signal reçu en signal numérique. Pour tester le bon déroulement de cette conversion on fait visualiser la variation des valeurs, le signal est visualisé à l'aide de PUTTY qui est un émulateur de terminal logiciel qui supporte les connexions par port série, et permet d'afficher les valeurs sur l'écran de l'ordinateur.



Terminal 1 (Left)	Terminal 2 (Middle)	Terminal 3 (Right)
<info> app: 35	<info> app: 206	<info> app: 950
<info> app: 40	<info> app: 197	<info> app: 950
<info> app: 31	<info> app: 160	<info> app: 949
<info> app: 42	<info> app: 222	<info> app: 951
<info> app: 35	<info> app: 222	<info> app: 949
<info> app: 45	<info> app: 176	<info> app: 949
<info> app: 41	<info> app: 141	<info> app: 950
<info> app: 37	<info> app: 171	<info> app: 870
<info> app: 36	<info> app: 193	<info> app: 950
<info> app: 36	<info> app: 199	<info> app: 951
<info> app: 38	<info> app: 217	<info> app: 950
<info> app: 56	<info> app: 216	<info> app: 869
<info> app: 45	<info> app: 198	<info> app: 869
<info> app: 50	<info> app: 148	<info> app: 845
<info> app: 38	<info> app: 166	<info> app: 919

FIGURE 5.3 – visualisation du signal numérique : mode repos, début d'une contraction, contraction complète

La figure 5.3 montre que les valeurs exprimant l'intensité de l'activité musculaire augmentent proportionnellement à l'intensité de la contraction musculaire.

5.3.2 Emplacement des capteurs EMG

Après plusieurs essais des emplacements du capteur EMG dans différents muscles dans le dos, on a déduit que le muscle trapèze est le plus sensible aux changements de la posture assise .pour cela on a choisi de placer les électrodes du capteur EMG Myoware sur ce muscle de la manière montrée dans la figure 5.3.

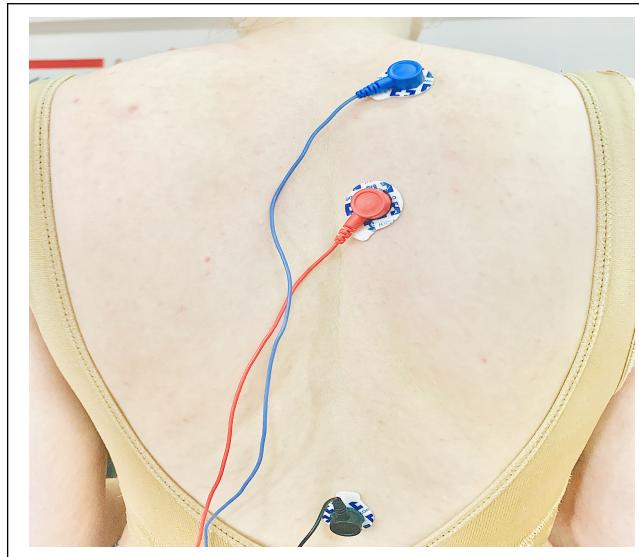


FIGURE 5.4 – Emplacement du capteur musculaire sur le dos

Comme la figure 5.4 le montre l'emplacement des électrodes dans le muscle .L'électrode de référence ayant la couleur noir est placé dans une zone osseuse près du muscle ciblé, tandis que l'électrode rouge à l'extrémité du muscle et le bleu au centre du muscle .

5.3.3 Assemblage du circuit

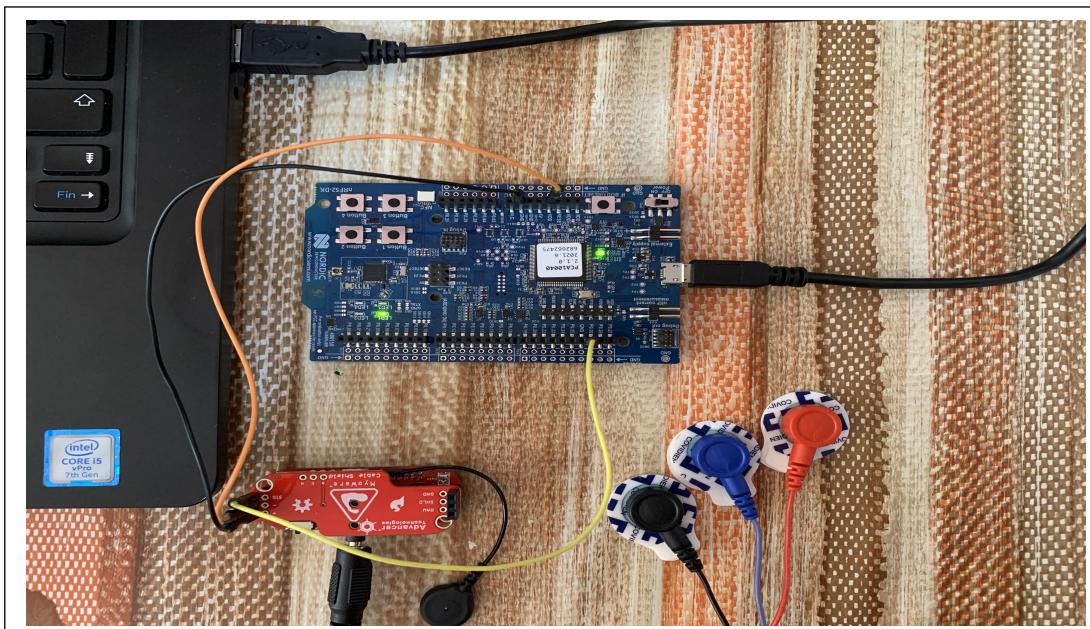


FIGURE 5.5 – Assemblage du circuit

IMPLEMENTATION ET RÉSULTATS

Comme la figure 5.5 l'illustre, pour notre assemblage de circuits, nous avons besoin d'une carte nRF52 et d'un capteur MyoWare, la carte est relié par câble USB vers le pc. La carte et le capteur sont reliés l'un à l'autre par 3 broches comme suit :

La broche VDD de la carte est connectée à la broche + du capteur par le fil noir.

La broche GND de la carte est connectée à la broche - du capteur par le fil orange.

Le SIG est connecté à la broche 2 de la carte par un fil jaune pour récupérer le signal analogique du Capteur.

Après avoir récupéré notre signal analogique à partir du capteur, nous devons le convertir en un signal numérique en utilisant un convertisseur analogique-numérique à approximation successive pour faire l'échantillonnage et toutes les 2 secondes nous aurons 10 échelles afin de le manipuler.

5.4 Implémentation du périphérique BLE

Le microcontrôleur se comporte comme un périphérique BLE car c'est l'application mobile qui va initier la communication BLE. Lors de la configuration du microcontrôleur on a opté pour l'utilisation d'une communication bidirectionnelle entre ce dernier et le pc via UART série et entre le microcontrôleur et l'application mobile via un Nordic UART service (NUS). Ces communications permettent le transfert des données nécessaires pour le bon fonctionnement de notre système de détection de mauvaise posture.

Le diagramme d'activité de la figure 5.6 montre les étapes que le microcontrôleur suit pour assurer les

communications. Tout d'abord, on initialise les services BLE (Nus ,Gatt..) et la communication UART, le microcontrôleur lit le signal analogique venant du capteur EMG à partir du port Ain0, puis le convertit en signal numérique en utilisant un convertisseur analogique-numérique à approximation successive pour faire l'échantillonnage et chaque deux secondes nous aurons 10 échelles à manipuler. Ces valeurs sont immédiatement transférées au pc où la classification sera effectuée. Au même temps le NRF52 reçoit via RX UART la donnée que l'utilisateur a écrit dans la caractéristique RX avec le service NUS dans l'application mobile, cette donnée

IMPLEMENTATION ET RÉSULTATS

correspond à l'âge et le sexe de cet utilisateur, le microcontrôleur les envoi directement au pc via TX UART pour classifier la posture ensuite il recoit le résultat de cette classification via RX UART au NRFf52. La dernière étape de cette communication c'est la transmission de ce résultat (l'état de la posture) vers le central BLE qui sera lu par l'utilisateur dans la caractéristique TX.

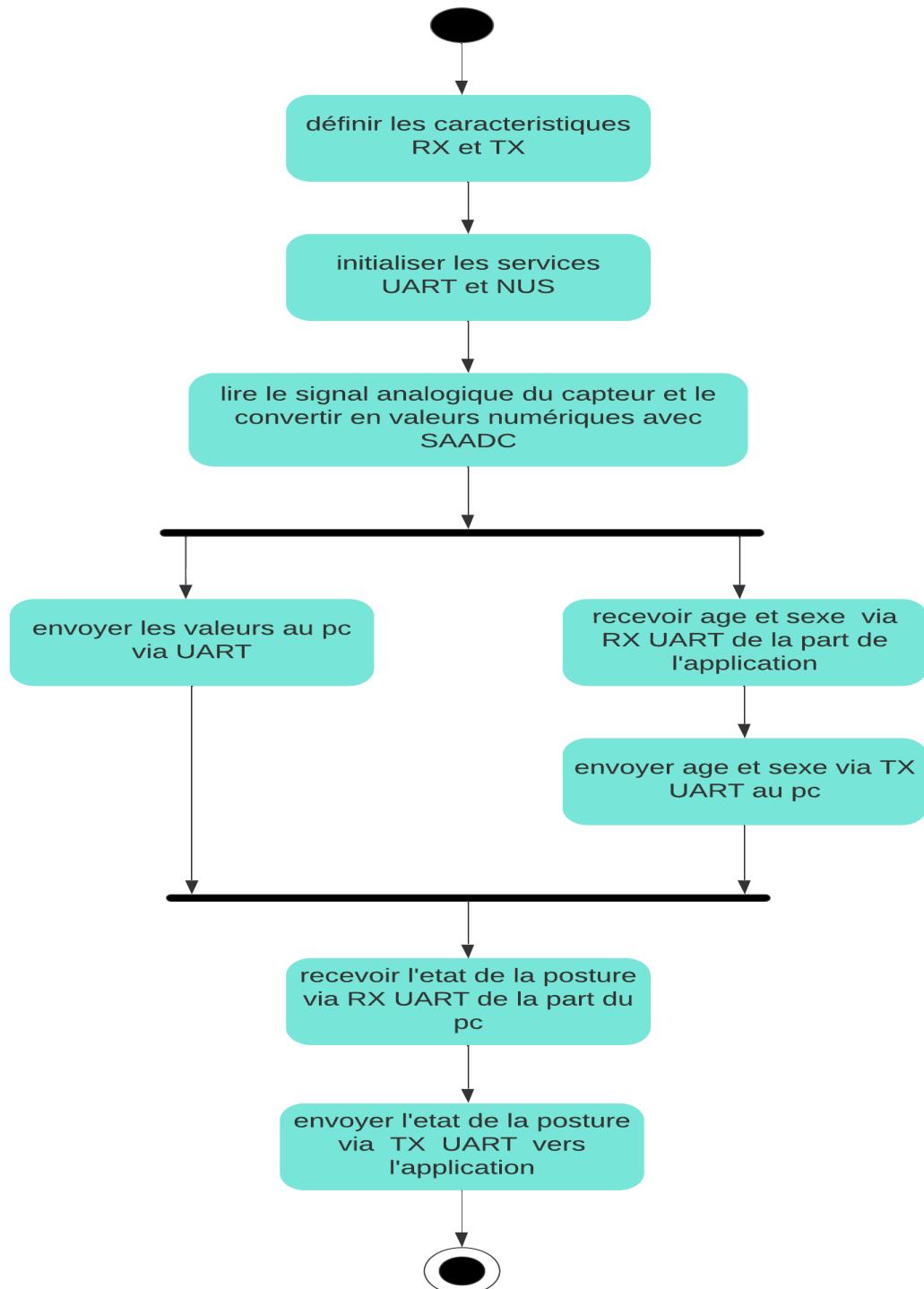


FIGURE 5.6 – Diagramme d'activité : périphérique BLE

5.5 Classification de la posture

5.5.1 Données d'entraînement

Pour mettre en place notre dataset , nous avons enregistré l'activité musculaire de plusieurs individus des deux sexes âgés de 18 à 50 ans dans des différentes positions assises à savoir position correcte , incliné vers l'avant , incliné à droite , incliné vers la gauche .La durée d'enregistrement est de 2 minutes pour chacune de ces dernières .

5.5.2 Prétraitement des données

Il semble intéressant de choisir des caractéristiques bien précises au lieu d'entrainer le modèle d'apprentissage sur la totalité des données brutes.

Pour faire , visualisons d'abord les données récoltées lorsque la personne est dans une bonne et mauvaise position (incliné vers l'avant) pendant 5 minutes.

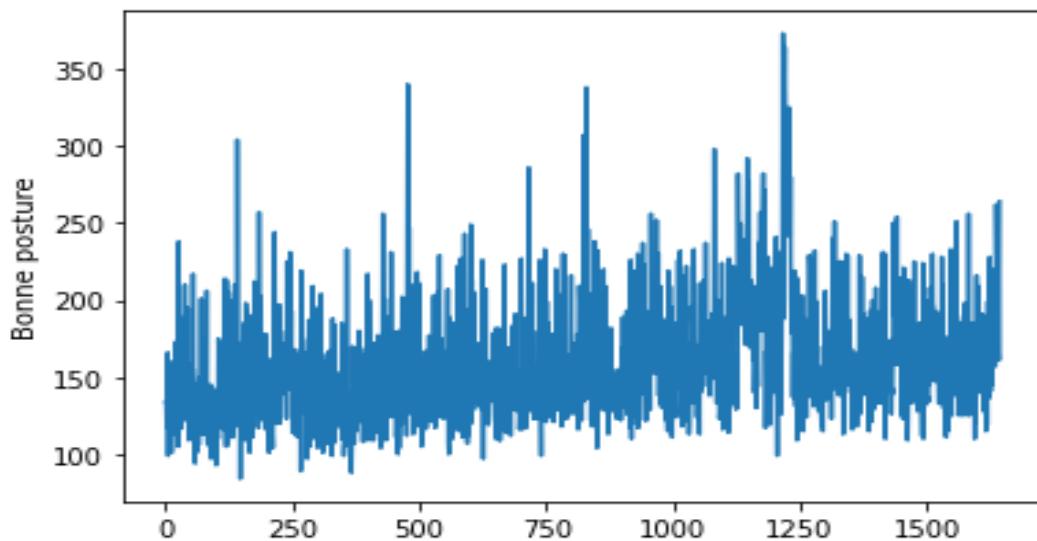


FIGURE 5.7 – graphe : visualisation des données en cas de bonne posture

La figure 5.7 montre que les données présentant la contraction musculaire lorsque la personne est en bonne position sont d'une variance faible c'est-à-dire que les valeurs sont proches les unes des autres (dans ce cas entre 90 et 250) , il atteint rarement de grandes valeurs

IMPLEMENTATION ET RÉSULTATS

(dans ce cas maximum est 350) ce qui exprime que la valeur moyenne sera petite (ici entre 90 et 250) ainsi que l'écart type est réduit .

la cause de cette variation est que lorsque l'utilisateur s'assoit correctement le muscle reste en état de repos , il n'effectue pas de grosses contractions.

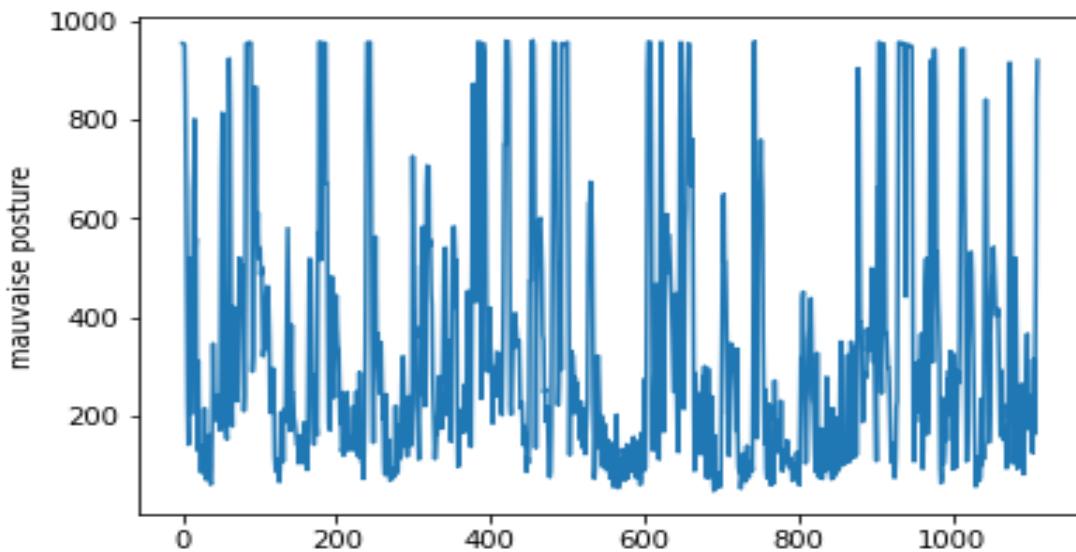


FIGURE 5.8 – graphe :visualisation des données en cas de mauvaise posture

La figure 5.8 montre que lorsque l'individu maintient une mauvaise posture la variance des valeurs de la contraction du muscle trapèze est élevée c'est-à-dire que les nombres sont très distants. (dans ce cas entre 90 et 1000) , il atteint souvent de grandes valeurs (dans ce cas maximum est 1000) ,l'écart type de la série de données ici est grand .La cause de cette variation est que lorsque l'utilisateur s'assoit inconvenablement le muscle effectue des contractions et avec le temps ça devient douloureux ce qui explique les pics.

Sur la base de ce qui est mentionné ci-dessus, les caractéristiques qu'on juge représentatives des séries de données sont le maximum(**max**), minimum(**min**), la moyenne(**mean**), la variance (**var**), l'écart type(**std**) .De plus , puisqu'on entraîne notre modèle sur des personnes du sexe et d'âge différents , on ajoute les caractéristiques **Age** et **Gender** .

IMPLEMENTATION ET RÉSULTATS

Y	max	min	mean	std	var	age	gender
bonne posture	593	60	148.3236364	67.63440756	4574.413086	22	0
bonne posture	964	67	189.1126187	270.3791964	73104.90985	19	0
bonne posture	962	75	217.8783784	161.7952975	26177.71828	23	0
bonne posture	964	64	261.9643268	209.3505237	43827.64179	21	0
bonne posture	423	180	276.9336219	38.2567563	1463.579403	24	0
bonne posture	503	67	102.4797203	52.36687077	2742.289154	20	0
bonne posture	962	38	151.2266974	136.1497563	18536.75615	30	1
bonne posture	906	80	142.7683284	75.64253785	5721.793533	18	0
bonne posture	373	84	157.6131387	37.84454183	1432.209346	23	0
mauvaise posture	963	61	312.6736905	227.451074	51733.99105	22	0
mauvaise posture	959	68	320.6885246	314.7829812	99088.32523	19	0
mauvaise posture	962	60	165.8701299	102.5049577	10507.26635	23	0
mauvaise posture	1023	176	521.4596182	278.7373557	77694.51344	24	0
mauvaise posture	966	96	560.657228	297.9913581	88798.8495	20	0
mauvaise posture	962	99	272.7501505	169.0705444	28584.84898	23	0
mauvaise posture	966	126	435.9261364	282.3364302	79713.85981	21	0
mauvaise posture	961	47	336.5229523	259.0917072	67128.51276	30	1
mauvaise posture	960	129	465.1117735	290.5782568	84435.72331	18	0

FIGURE 5.9 – visualisation du dataset

La figure 5.9 montre une partie de données après extraction des caractéristiques pour chaque série de données enregistrées pendant 2 minutes exprimant une bonne et mauvaise posture.

5.5.3 Outils de programmation

Notre choix de langage de programmation se porte sur **python** à cause de sa facilité d'utilisation et la présence des bibliothèques destinées au domaine de l'analyse de données et d'apprentissage automatique , parmi ces derniers :

-**Scikit-Learn** (sklearn), est une bibliothèque utilisé essentiellement pour l'apprentissage automatique. Elle contient différents algorithmes de classification, de régression et de clustering, ainsi que divers fonctions pour le prétraitement des données.

-**TensorFlow** est un framework open source développé pour exécuter l'apprentissage automatique , l'apprentissage en profondeur et d'autres charges de travail d'analyse statistique et prédictive.

-**Keras** est une API de réseau de neurones écrite en python, conçue pour être rapide et simple

IMPLEMENTATION ET RÉSULTATS

à utiliser pour le développement et le teste de réseaux de neurones. Elle permet de créer facilement des couches pour ces réseaux en profondeur et de mettre en place des architectures complexes.

- **NumPy** est une bibliothèque numérique open source, elle prend en charge les tableaux et matrices à grande dimension.

- **Matplotlib** :bibliothèque qui permet de créer et visualiser des graphes sur python.

-**Seaborn** est une bibliothèque Python pour la visualisation de données basée sur Matplotlib. Il fournit une interface de haut niveau pour dessiner des graphiques statistiques attrayants et informatifs.

-**pandas** :une librairie qui permet d'importer des tableaux csv, excel ...etc , dans python afin de charger les données d'entraînement dans sklearn.

Pour l'environnement de développement on a opté pour **jupyter** offert par anaconda car c'est un environnement de développement Python (IDE) pour la science des données. Il possède des fonctionnalités d'édition avancées et offre des tests interactifs à travers son interface.

5.5.4 Classification binaire

Dans ce cas le modèle est entraîné sur deux types de données, celles recueillies dans une position convenable et dans une mauvaise posture. Tout d'abord, on crée plusieurs modèles : un modèle de classification basé sur les réseaux de neurones et on détaille son architecture , quatre modèles d'apprentissage automatique à savoir : support vector machine(SVM) , K-nereast neighbor(KNN), Decision tree(DT),random forest(RF) et on détaille pour chacun ses paramètres.

Architecture du modèle perceptron multicouches :

Après plusieurs essais d'implémenter un réseau de neurones avec plusieurs couches cachées on a déduit que l'architecture la plus adéquate à notre ensemble de données est celle présentée et détaillée par la suite (FIGURE 5.10). En effet on a essayé d'augmenter le nombre de couches

IMPLEMENTATION ET RÉSULTATS

jusqu'à six en variant les fonctions d'activations (Relu, Sigmoid, softmax ...) et le nombre de neurones de chacune mais la justesse n'augmente pas car la taille de notre dataset est petite pour l'utilisation de l'apprentissage en profondeur.

L'architecture que nous avons utilisée est de type perceptron multicouches (PMC). La particularité de ce modèle de réseaux est que l'information y circule dans un seul sens, celui de l'avant.

Ce système est dit non bouclé, il ne comporte donc pas de boucles au niveau des connexions neuronales. Un PMC est organisé en couches. Une couche est un ensemble de neurones qui n'ont pas de connexion entre eux mais plutôt avec des neurones d'autres couches. Chacun de ces neurones suit le fonctionnement d'un neurone formel.

En principe, le perceptron multicouches comporte au moins trois couches successives : une couche d'entrée qui lit les informations entrantes provenant des caractéristiques obtenues durant la phase de prétraitement (un neurone= une entrée), une ou plusieurs couches cachées participant au transfert, et enfin une couche de sortie qui fournit la réponse du système (la classe de l'élément à classifier).

voici à quoi ressemble notre réseau de neurones :

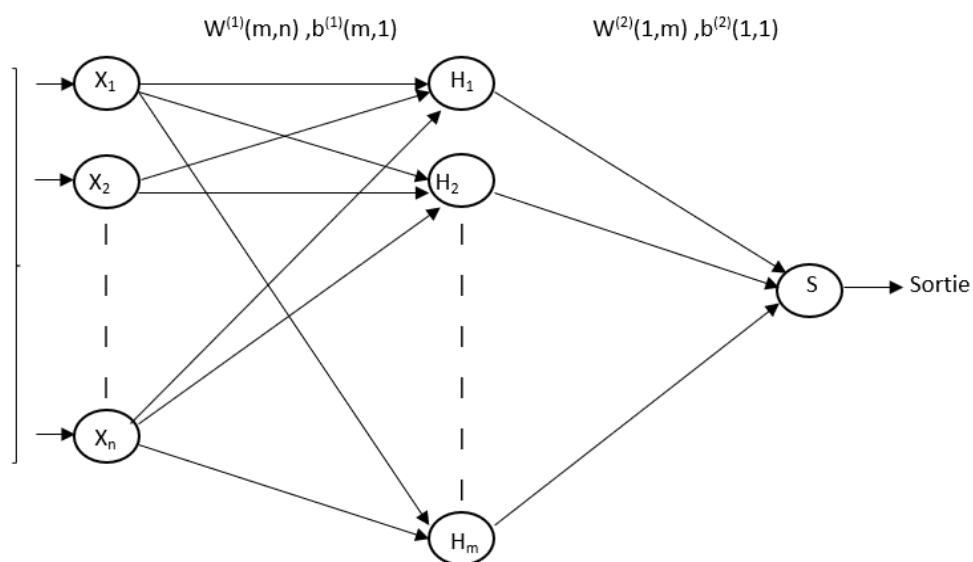


FIGURE 5.10 – Architecture du réseau de neurones

IMPLEMENTATION ET RÉSULTATS

La figure 5.10 représente l'architecture de notre réseau .

le PMC se compose de :

- Une **couche d'entrée** de **i=7** neurones i étant le nombre de caractéristiques (max, min, mean, var, std, age, gender).
- Une **couche cachée** de **m=40** neurones, dont la fonction d'activation **F** est tangente hyperbolique **(tanh)**.
- Une **couche de sortie** de **k=1** neurone. Sa fonction d'activation **O** est la fonction **Sigmoïde** qui est non linéaire également connu sous le nom de fonction logistique. Cette dernière est généralement utilisée dans la couche de sortie de la classification binaire, où le résultat est soit 0, soit 1, car la valeur de la fonction sigmoïde se situe entre 0 et 1 seulement, le résultat peut être facilement prédit comme étant 1 (classe :mauvaise posture) si la valeur est supérieure ou égale à 0,5 et 0 (classe :bonne posture) sinon.

Le résultat correspondant au neurone de sortie est obtenu comme suit :

$$O(S) = \frac{1}{1 + \exp(-S)} \quad (5.1)$$

Ce résultat représente la probabilité d'appartenance de l'entrée à la classe qui correspond au neurone de sortie.

La sortie d'Hm le m ième neurone de la couche cachée, est calculée avec la formule 5.2 :

$H_m = F(m)$ avec

$$m = b_m^{(1)} + \sum_{i=1}^{n=7} W_{mi} * X_i \quad (5.2)$$

La sortie de S, le neurone de la couche de sortie, est calculée à l'aide de la formule 5.3 :

S=O(S) où

$$S = b^{(2)} + \sum_{j=1}^{m=40} W_{mj} * H_j \quad (5.3)$$

Ainsi, la formule 5.4 représente la valeur de la sortie S en fonction des entrées, des poids et des

IMPLEMENTATION ET RÉSULTATS

biais :

$$S = O(b^{(2)} + \sum_{j=1}^{m=40} W_j^{(2)}(F(b_j + \sum_{i=1}^{n=7} W_{ji(1)} * X_i))) \quad (5.4)$$

Une fois que l'architecture du réseau de neurones est fixée, nous passons au processus d'apprentissage. Il s'agit d'un apprentissage supervisé, le but est de faire converger la sortie du réseau vers le résultat désiré.

Pour cela, nous avons séparé l'ensemble de données en un ensemble d'entraînement et un autre de test, ce dernier représente 20% de la totalité.

Le processus d'apprentissage consiste à minimiser la fonction d'erreur (appelée aussi fonction de coût) qui représente l'écart entre le résultat obtenu et le résultat désiré, et cela en modifiant les poids des neurones de manière itérative jusqu'à atteindre un critère d'arrêt.

Pour ce faire, on a adopté le mode batch où les poids sont mis à jour après avoir appliqué toutes les entrées d'entraînement au réseau.

Pour notre problème de classification binaire, la fonction de coût qu'on trouve appropriée est celle de l'entropie binaire croisée .

En effet, combinée à la fonction d'activation Sigmoïde au niveau du neurones de la couche de sortie, elle facilite l'interprétation des résultats sous forme de probabilité d'appartenance des échantillons à une classe.

L'entropie croisée du réseau est calculée comme suit :

$$E = \frac{1}{l} \sum_{j=1}^l T_j * \ln(S_j) \quad (5.5)$$

l : représente le nombre total d'échantillons.

T_j : réponse désirée au niveau de la sortie de groupe d'échantillons j.

Il existe plusieurs algorithmes d'apprentissage permettant d'estimer les poids d'un perceptron multicouches.

Il s'agit d'algorithmes d'optimisation numérique qui utilisent la technique de rétropropagation. Cette technique consiste à propager l'erreur calculée à travers le réseau de la couche de sortie

IMPLEMENTATION ET RÉSULTATS

vers la couche d'entrée.

dans notre cas on a opté pour un algorithme d'optimisation présent sur Tensorflow c'est l'optimisation d'Adam il s'agit d'une méthode de descente de gradient stochastique basée sur l'estimation adaptative des moments du premier et du deuxième ordre.

L'entraînement s'arrête lorsque la fonction de coût atteint son minimum ou cette dernière ne diminue plus.

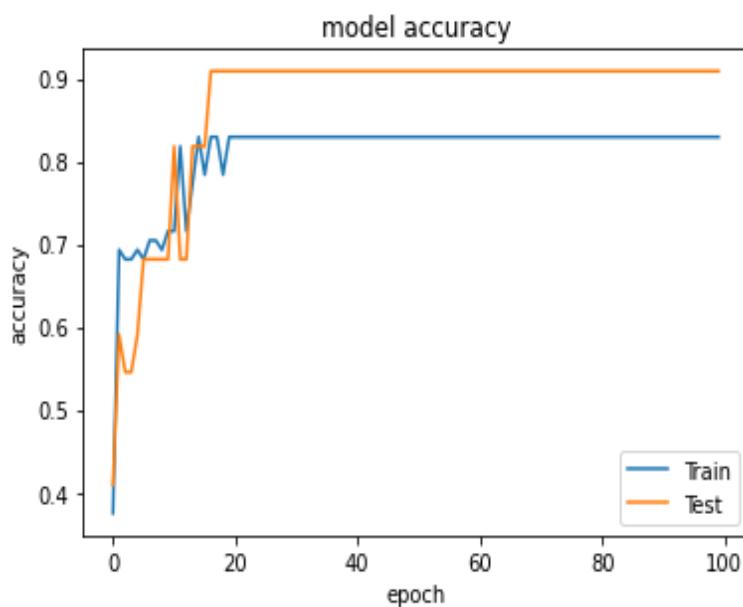


FIGURE 5.11 – Multilayer Perceptron :accuracy

Le graphe de la figure 5.11 montre la variation de la justesse en fonction du nombre d'epochs, on remarque que le modèle a atteint une justesse de 92% pour les données de test et 84% pour les données d'entraînement . Ces valeurs pouvaient être plus élevées si la taille de l'ensemble de données était plus grande.

Le temps d'exécution pour entraîner l'ensemble de données est de 6,3 ms, ceci est très long pour l'implémenter en temps réel.

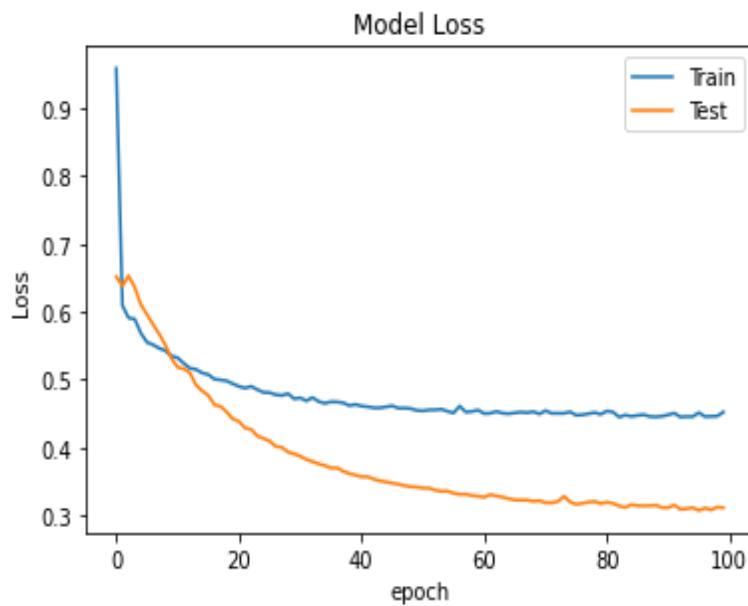


FIGURE 5.12 – Multilayer Perceptron :loss

le graphe de la figure 5.12 montre la variation de la perte d'entropie croisée binaire en fonction du nombre d'epochs. On remarque que la perte diminue avec le temps.

Afin de créer les modèles d'apprentissage automatique (SVM, KNN, DT, RF) nous allons utiliser la bibliothèque sk-learn de python. On varie les paramètres essentiels de chaque modèle Afin de choisir celles qui donnent la meilleure justesse comme montré ci-dessous :

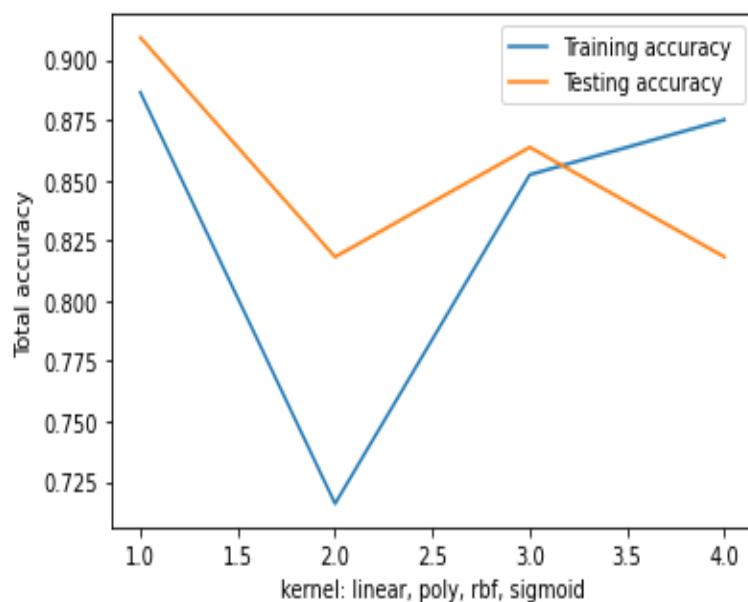


FIGURE 5.13 – Support Vector machine :accuracy

IMPLEMENTATION ET RÉSULTATS

La figure 5.13 affiche les valeurs de la justesse en fonction du noyau du SVM. Le choix du paramètre s'est porté sur le noyau linéaire en effet les valeurs de la justesse pour les données d'entraînement et les données du test sont au maximum et se rapprochent l'un de l'autre (89% et 91%). Le temps d'exécution pour entraîner l'ensemble de données est de 3ms.

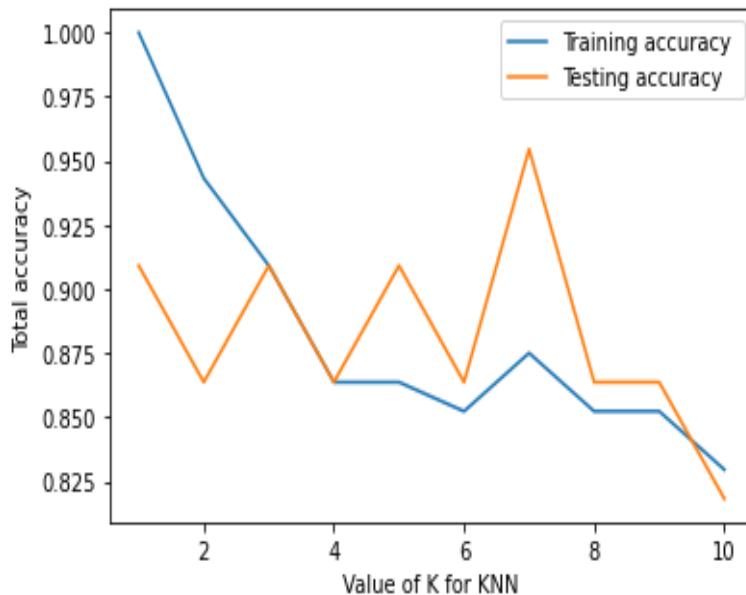


FIGURE 5.14 – k-Nearest Neighbor :accuracy

La figure 5.14 affiche la variation de la justesse en fonction du nombre de voisins les plus proches (k) dans le modèle KNN, lorsque la valeur de k est égale à 3 la justesse pour les données d'entraînement et les données de test sont égaux (91%). Le temps d'exécution pour entraîner l'ensemble de données est de 0,0066 ms.

IMPLEMENTATION ET RÉSULTATS

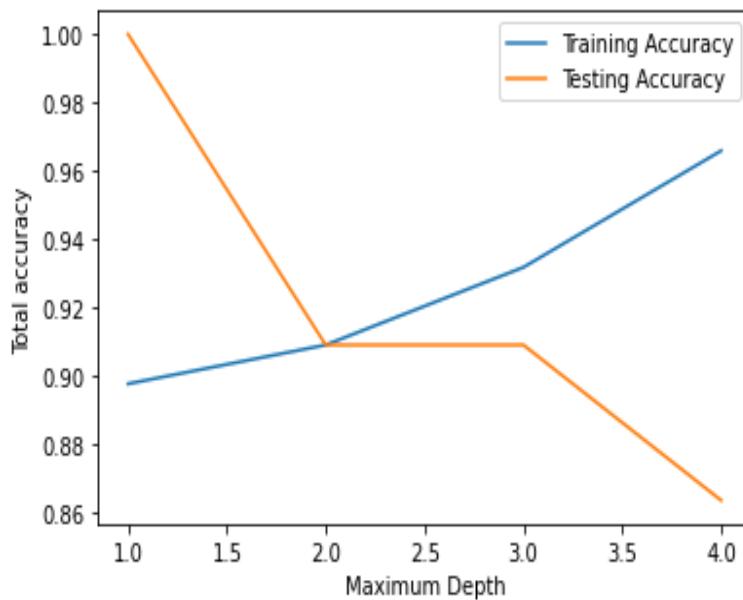


FIGURE 5.15 – Decision Tree :accuracy

La figure 5.15 montre la variation de la justesse en fonction de la profondeur maximale dans le modèle DT, lorsque la valeur de la profondeur est égale à 2 la justesse pour les données d'entraînement et de test sont égaux (91%). Le temps d'exécution pour entraîner l'ensemble de données est de 0,012 ms.

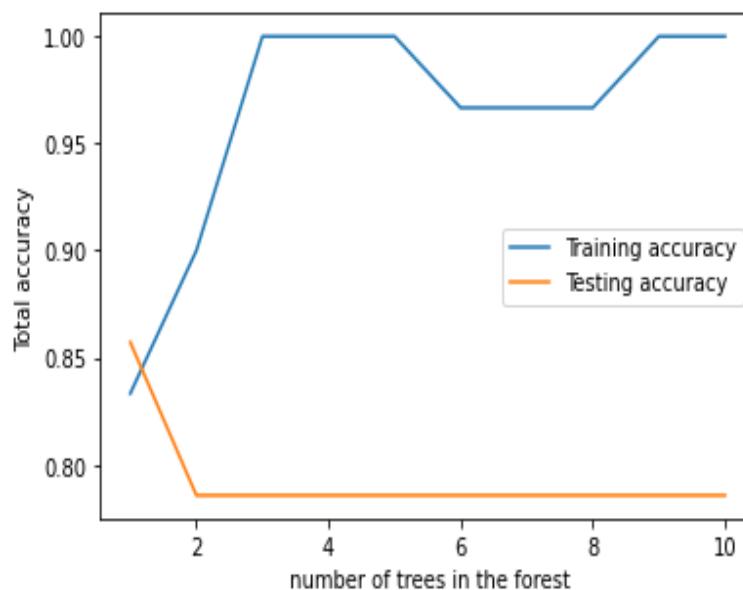


FIGURE 5.16 – Random Forest :accuracy

La figure 5.16 montre la variation de la justesse en fonction du nombre d'arbres dans la forêt dans le modèle RF, lorsque la valeur du nombre d'arbres est égale à 1 la justesse pour

IMPLEMENTATION ET RÉSULTATS

les données d'entraînement et de test sont proches (82% -89%). Le temps d'exécution pour entraîner l'ensemble de données est de 0,0045 ms.

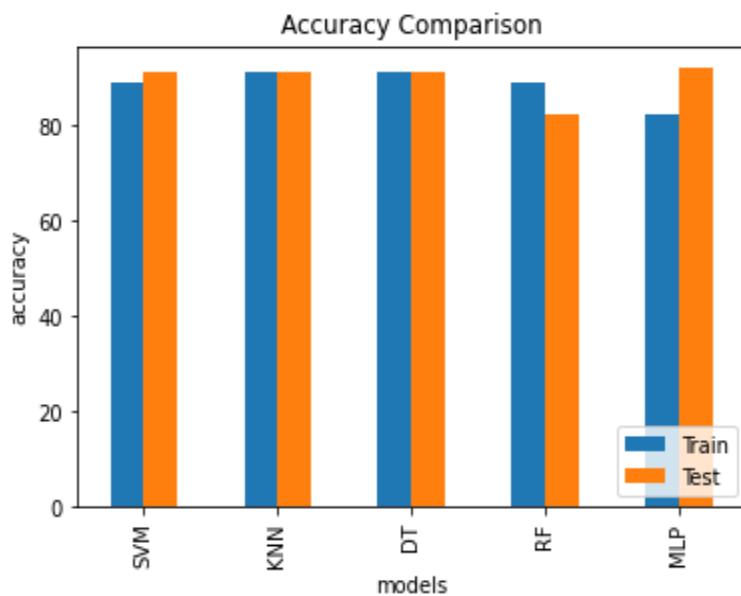


FIGURE 5.17 – accuracy comparison

Le graphe de la figure 5.17 exprime une comparaison entre les modèles utilisés en terme de leurs justesse, on remarque que le modèle KNN et DT ont la justesse la plus élevée(91%) pour les données d'entraînement et de test. Afin de choisir le plus performant des deux on observe le temps d'execution, KNN a le temps d'execution le plus réduit donc c'est le modèle le plus adapté pour le système embarqué en temps réel .

5.5.5 Classification Multiclassee

Dans ce cas le modèle est entraîné sur quatre types de données celle recueillies dans une position convenable et dans trois autres mauvaises postures celle inclinée vers l'avant , inclinée vers la gauche ,inclinée vers la droite . Comme dans le cas de classification binaire nous avons opté pour une comparaison entre MLP,SVM,KNN,DT,RF.

Architecture du MLP

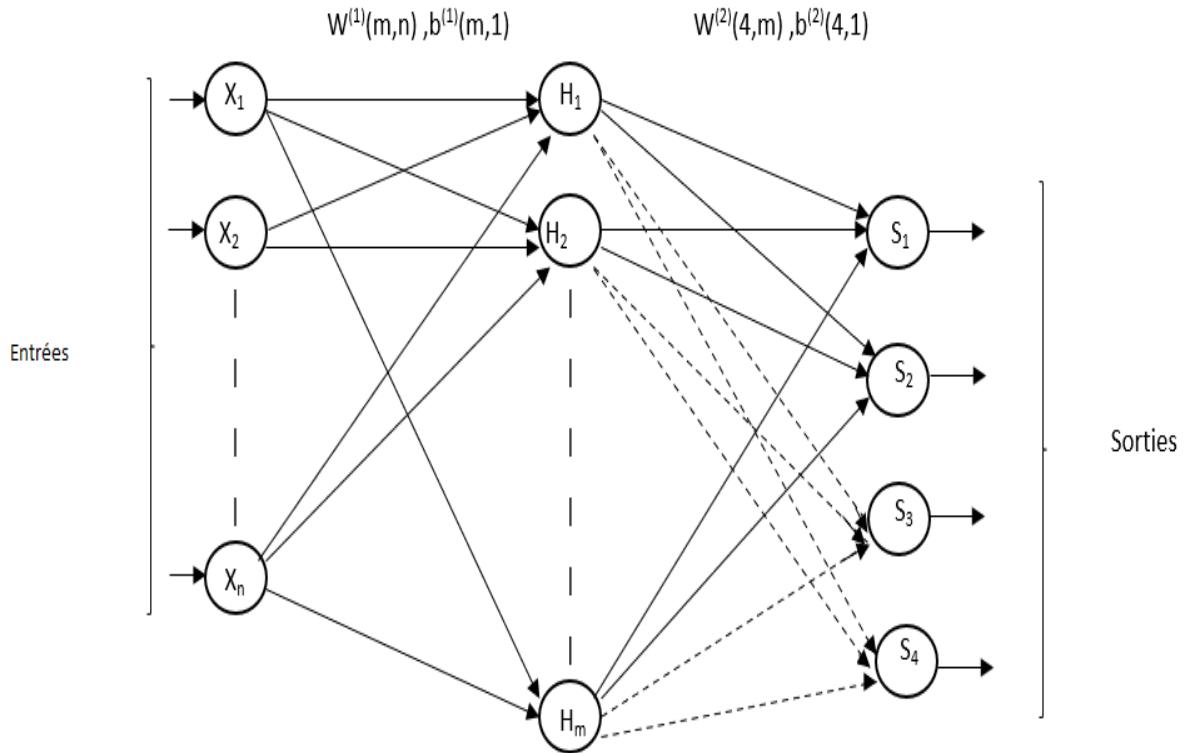


FIGURE 5.18 – Architecture du perceptron multicouche dans le cas d'une classification multiclesse

La figure 5.18 montre l'architecture du MLP. Cette dernière est légèrement différente de celle de classification binaire , en effet le nombre de neurones de la couche de sortie est quatre car on a quatre classes, de plus la fonction d'activation O de la couche de sortie est softmax qui est une fonction utilisée dans des problèmes de classification.

Le résultat correspondant au neurone x (x k=4) est obtenu comme suit :

$$O(S_x) = \frac{e^{(S_x)}}{\sum_{n=1}^{k=4} e^{(S_n)}} \quad (5.6)$$

Ce résultat représente la probabilité d'appartenance de l'entrée à la classe qui correspond au neurone x. La fonction softmax affecte un résultat positif à chacun des neurones de sortie, et la somme des sorties est égale à 1.

Une autre différence est que la fonction coût utilisée est l'entropie croisée catégorique .il s'agit

IMPLEMENTATION ET RÉSULTATS

de la valeur par défaut pour les problèmes de classifications multiclasses.

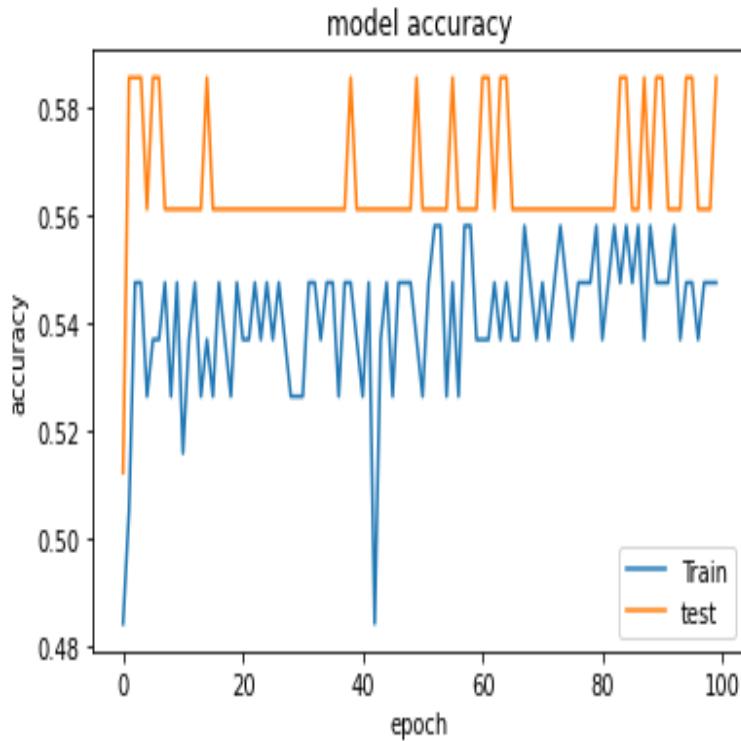


FIGURE 5.19 – Multilayer perceptron model on multiclass classification :accuracy

Le graphe de la figure 5.19 montre la variation de la justesse en fonction du nombre d'epochs, on remarque que le modèle a atteint une justesse de 59% pour les données de test et 56% pour les données d'entraînement .ce qui n'est pas très satisfaisant pour notre système embarqué. Le temps d'exécution pour entraîner l'ensemble de données est de 7 ms ceci est très long pour un système en temps réel.

Une augmentation du nombre d'individus et une variation dans l'âge et le sexe de ces personnes lors de la phase de la collecte de données d'entraînement engendrera une augmentation dans les performances du modèle multilayer perceptron.

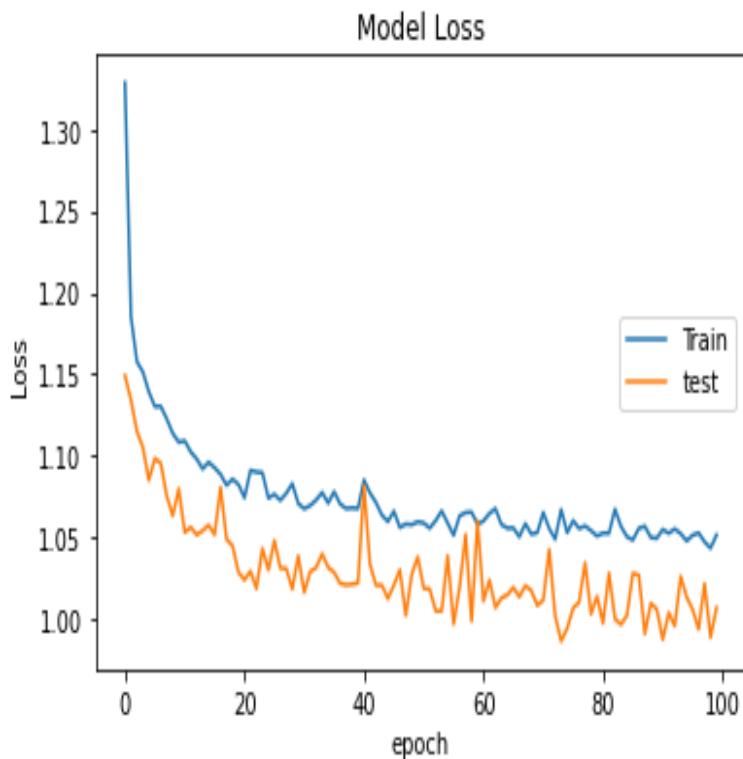


FIGURE 5.20 – Multilayer perceptron model on multiclass classification :accuracy

Le graphe de la figure 5.20 montre la variation de la perte d'entropie croisée catégoricale en fonction du nombre d'epochs. On remarque que la perte diminue avec le temps. Ensuite nous allons créer quatre classificateurs (SVM,KNN,DT,RF). Voici des graphes exprimant le choix paramètres de chaque modèle en fonction de la métrique 'justesse'.

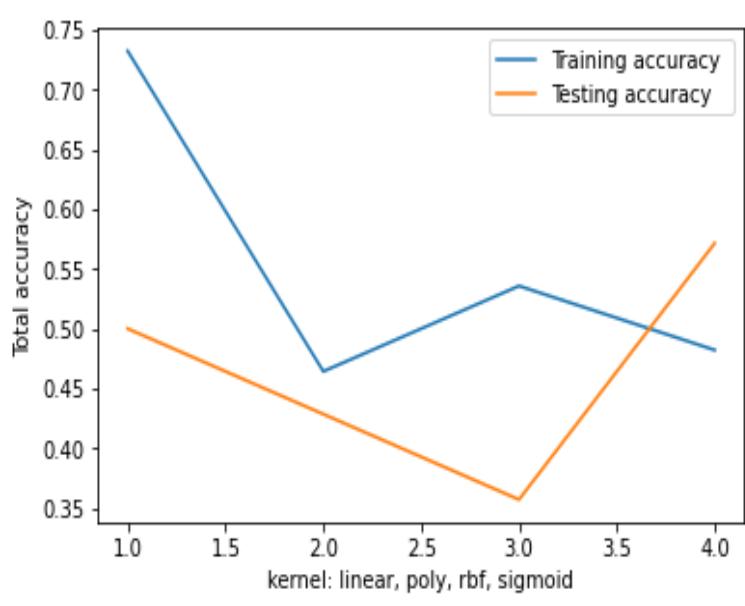


FIGURE 5.21 – Support Vector machine :accuracy

La figure 5.21 affiche les valeurs de la justesse en fonction du noyau du SVM ,lorsque le noyau est sigmoïde la justesse est égale à 59% pour les données d'entraînement et 59% pour les données du test.Le temps d'exécution pour entraîner l'ensemble de données est de 0,2ms.

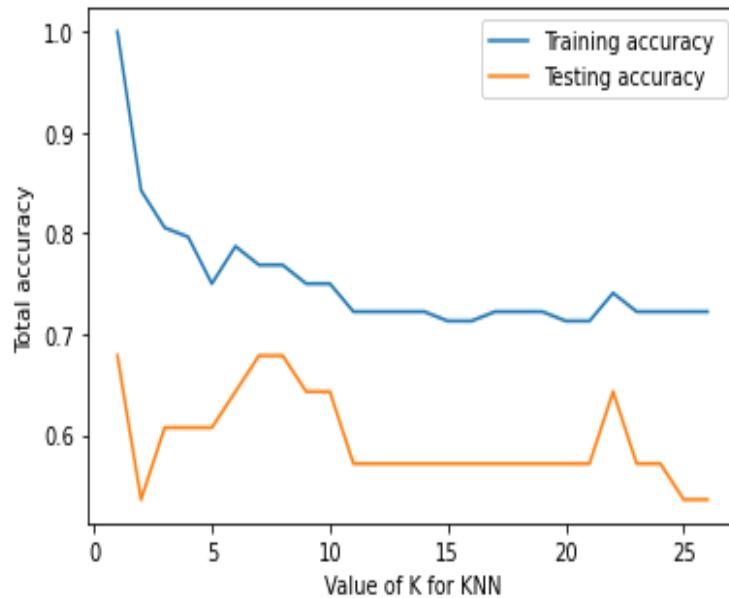


FIGURE 5.22 – k-Nearest Neighbor :accuracy

La figure 5.22 affiche la variation de la justesse en fonction du nombre de voisins les plus proches (k) dans le modèle KNN, lorsque la valeur de k est égale à 8 la justesse pour les données

IMPLEMENTATION ET RÉSULTATS

d'entraînement et les données de test sont les plus proches (train=77%, test==68%). Le temps d'exécution pour entraîner l'ensemble de données est de 0,011ms.

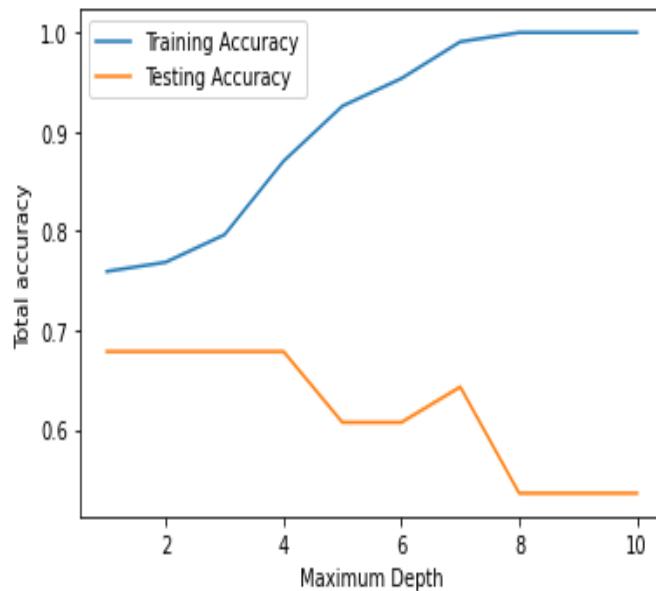


FIGURE 5.23 – Decision Tree :accuracy

La figure 5.23 montre la variation de la justesse en fonction de la profondeur maximale dans le modèle DT, lorsque la valeur de la profondeur est égale à 4 la justesse pour les données d'entraînement et de test sont proches (train=76%, test==68%). Le temps d'exécution pour entraîner l'ensemble de données est de 0,023ms.

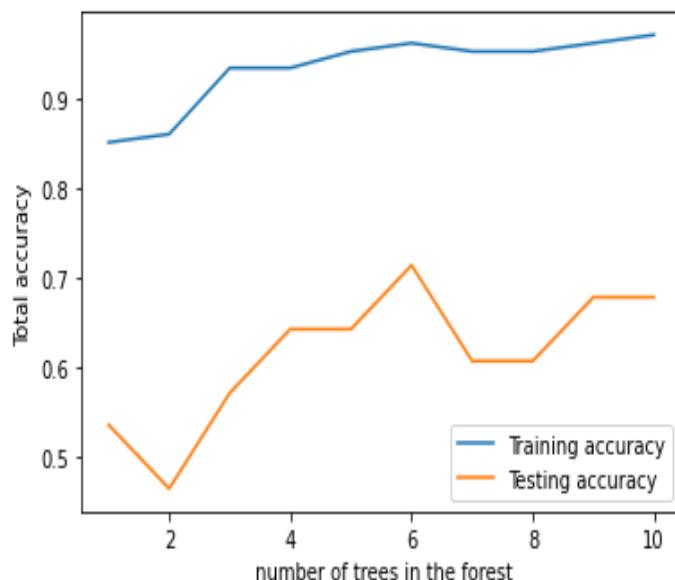


FIGURE 5.24 – Random Forest :accuracy

La figure 5.24 montre la variation de la justesse en fonction du nombre d'arbres dans la forêt dans le modèle RF, lorsque la nombre d'arbres est égale à 6 la justesse pour les données d'entraînement et de test sont les plus proches (train=96%, test=71%). Le temps d'exécution pour entraîner l'ensemble de données est de 0,022ms.

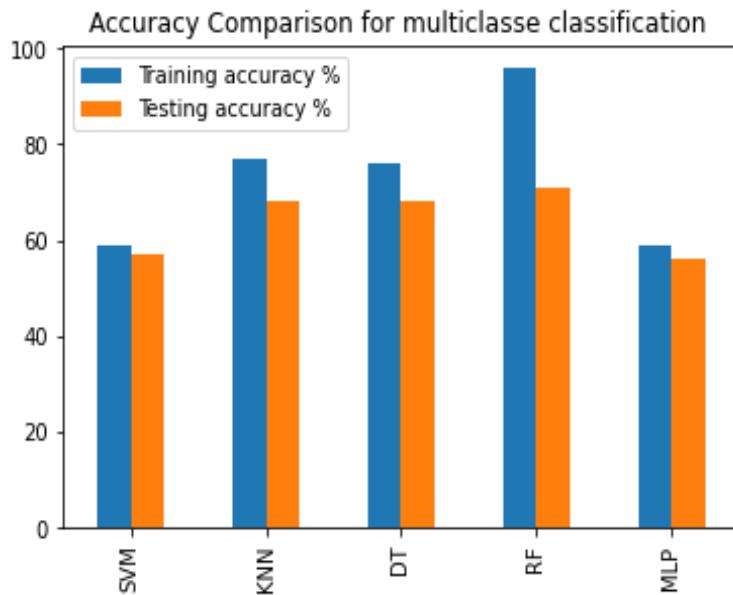


FIGURE 5.25 – accuracy comparison

le graphe de la figure 5.25 exprime une comparaison entre les modèles utilisés en terme de leurs justesse, On remarque une grande variance entre la justesse des données d'entraînement et de test pour le modèle de forêts aléatoires RF (train=96% »test=71%). Ceci est un sous-apprentissage ça sous-entend que le modèle prédictif généré lors de la phase d'apprentissage s'adapte mal au training set par conséquent le taux d'erreur reste grand. Donc on exclut ce modèle de la comparaison. KNN est le plus performant car il a la valeur de justesse plus élevée et de temps d'exécution le plus réduit.

5.6 Application mobile

L'application est développée en utilisant le framework Flutter. Ce dernier permet de déployer des applications sur plusieurs appareils à partir d'une seule base de code tel que les appareils mobiles, Web, de bureau et embarqués.

IMPLEMENTATION ET RÉSULTATS

Dans ce cadre, toutes les structures de page sont implémentées par des composants Widget imbriqués. Les pages sont commutées en changeant les composants du corps principal, et le contenu modifié est redessiné par la méthode setState.

Pour transmettre en temps réel l'état de la posture (résultat du modèle de classification) du PC vers une application mobile on a opté pour une communication Bluetooth low energie-uart. Du coté application mobile on a utilisé la bibliothèque flutter-reactive-ble pour créer une émulation uart sur BLE.

Notre application utilise le service nordic uart avec un UUID spécifique au fournisseur, le service susmentionné utilise deux caractéristiques : une caractéristique RX pour l'écriture et TX pour la lecture.

5.6.1 Implémentation du central BLE

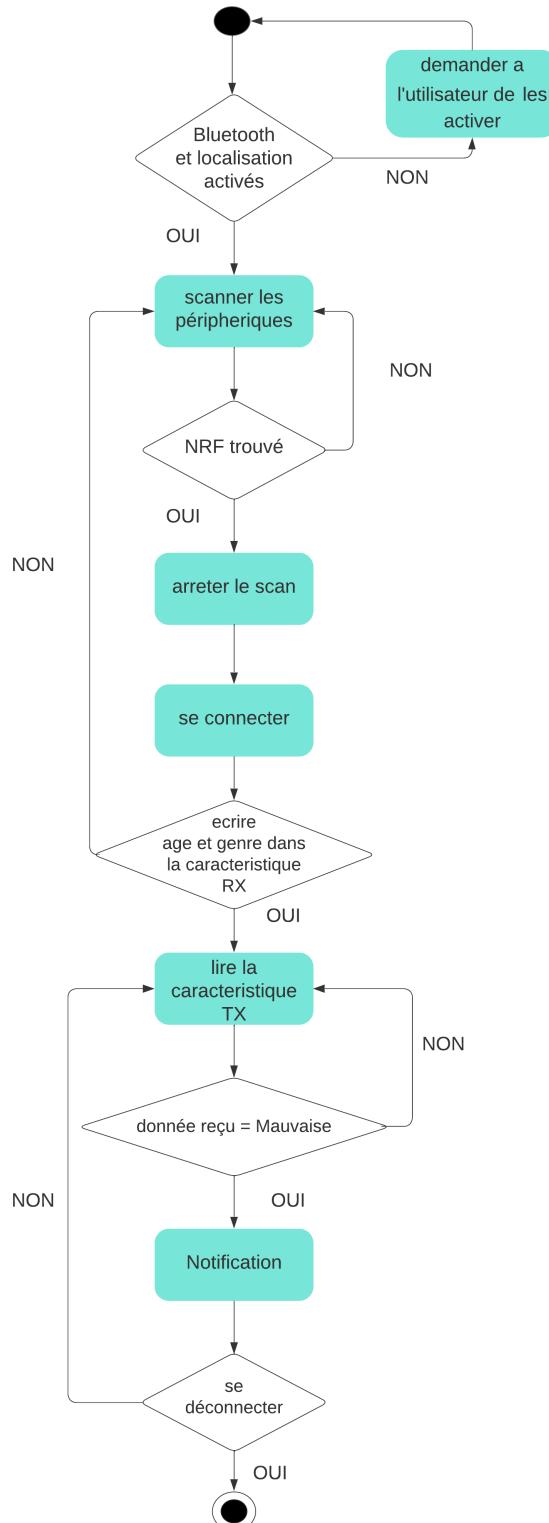


FIGURE 5.26 – Diagramme d’activités du central BLE.

IMPLEMENTATION ET RÉSULTATS

Après avoir implémenté le module BLE dans notre microcontrôleur, il est temps de communiquer avec ce dernier pour envoyer et recevoir les données dans le téléphone. Pour ce faire sur l'application Flutter, nous devons utiliser le package Flutter-reactive BLE qui gère les opérations BLE pour plusieurs périphériques. La bibliothèque BLE réactive prend en charge les éléments suivants :la découverte de périphérique BLE, l'observation de l'état BLE du périphérique hôte, l'établissement d'une connexion BLE, le maintien de l'état de connexion de plusieurs périphériques BLE, la découverte des services implicitement, la lecture et l'écriture des caractéristiques, l'abonnement à une caractéristique ..etc.Un moyen simple de communiquer entre deux appareils BLE consiste à utiliser un « UART » simulé. Un UART fournit un flux d'octets bidirectionnel, de sorte que les deux extrémités d'une connexion peuvent transmettre et recevoir des octets l'une avec l'autre à l'aide des caractéristiques TX et RX. Pour commencer à travailler avec l'application, nous devons ajouter une autorisation d'accès à l'emplacement et demander à l'utilisateur d'activer son bluetooth, puis nous commençons à scanner les appareils en se connectant a notre microcontrôleur après avoir arrêter le scan ,ensuite on écrit les données âge et sexe dans la caractéristique RX qui sera envoyé vers le microcontrôleur, on lit l'état de la posture reçu dans la caractéristique TX.

5.6.2 Interface utilisateur

Pour le prototype de l'application, nous aurons trois pages principales qui apparaîtront dans la barre de navigation à savoir une page d'accueil, une page de connexion UART-BLE et une page qui contient des conseils pour maintenir une bonne posture.

5.6.2.1 La page d'accueil

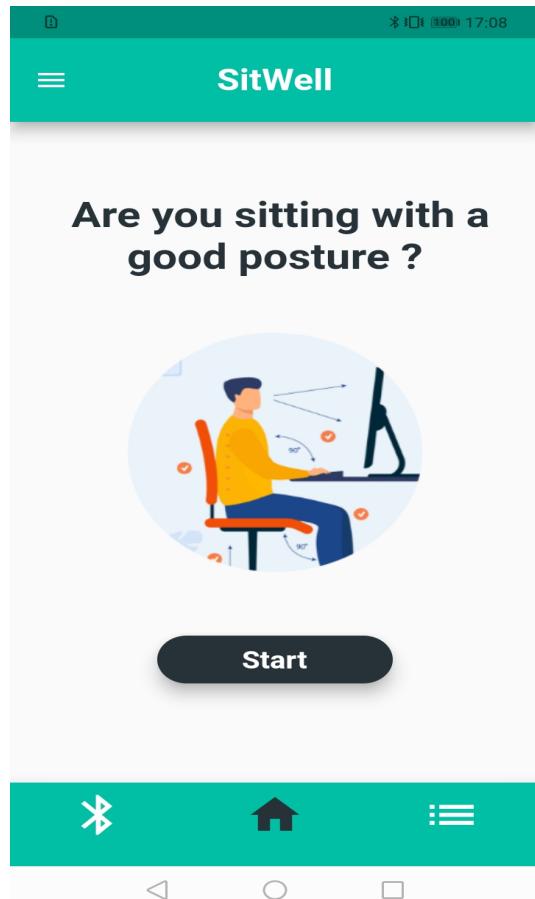


FIGURE 5.27 – Page d'accueil

La figure 5.27 montre la page qui apparait lors de l'ouverture de l'application celle-ci contient le logo et le nom de l'application (sitwell), barre de navigation inférieure contient un bouton Bluetooth qui dirige vers la page de communication BLE-UART, et un bouton liste qui mène vers la page des conseils, le bouton du milieu c'est pour la page Home.

IMPLEMENTATION ET RÉSULTATS

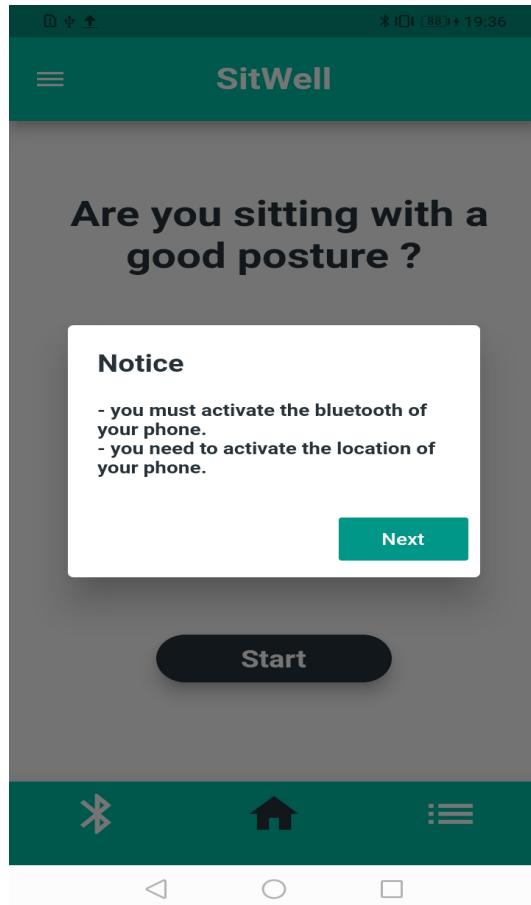


FIGURE 5.28 – Page d'accueil (bouton Start)

La figure 5.28 montre que lorsque l'utilisateur clique sur le bouton start une notice apparaît. La fenêtre notice informe l'utilisateur des étapes qu'il doit suivre afin d'utiliser l'application d'une manière correcte à savoir se connecter via Bluetooth et activer la localisation, le bouton Next dérige l'utilisateur vers la page connexion BLE-UART

5.6.2.2 Page de connexion BLE-UART

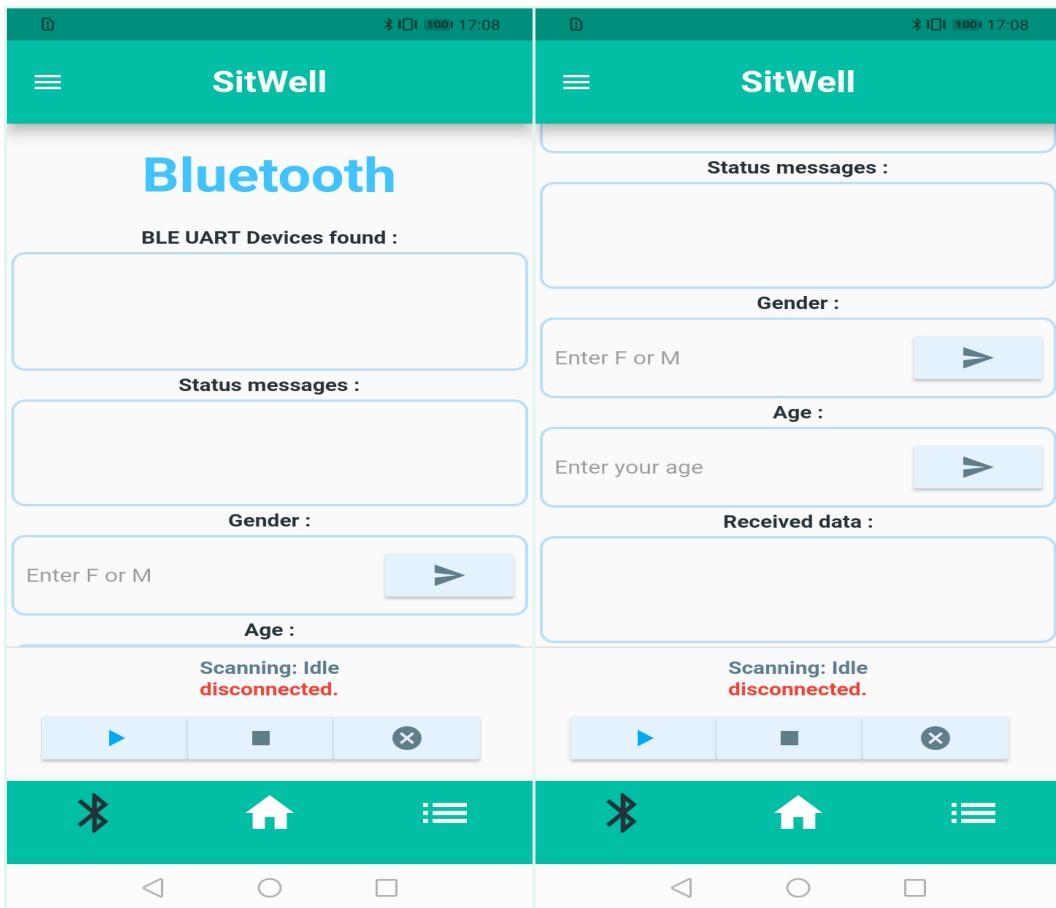


FIGURE 5.29 – Page de connexion BLE-UART

Les figure 5.29 illustre la page de connexion via BLE-UART, qui permet de réaliser la communication entre l'application et le microcontrôleur NRF.

IMPLEMENTATION ET RÉSULTATS

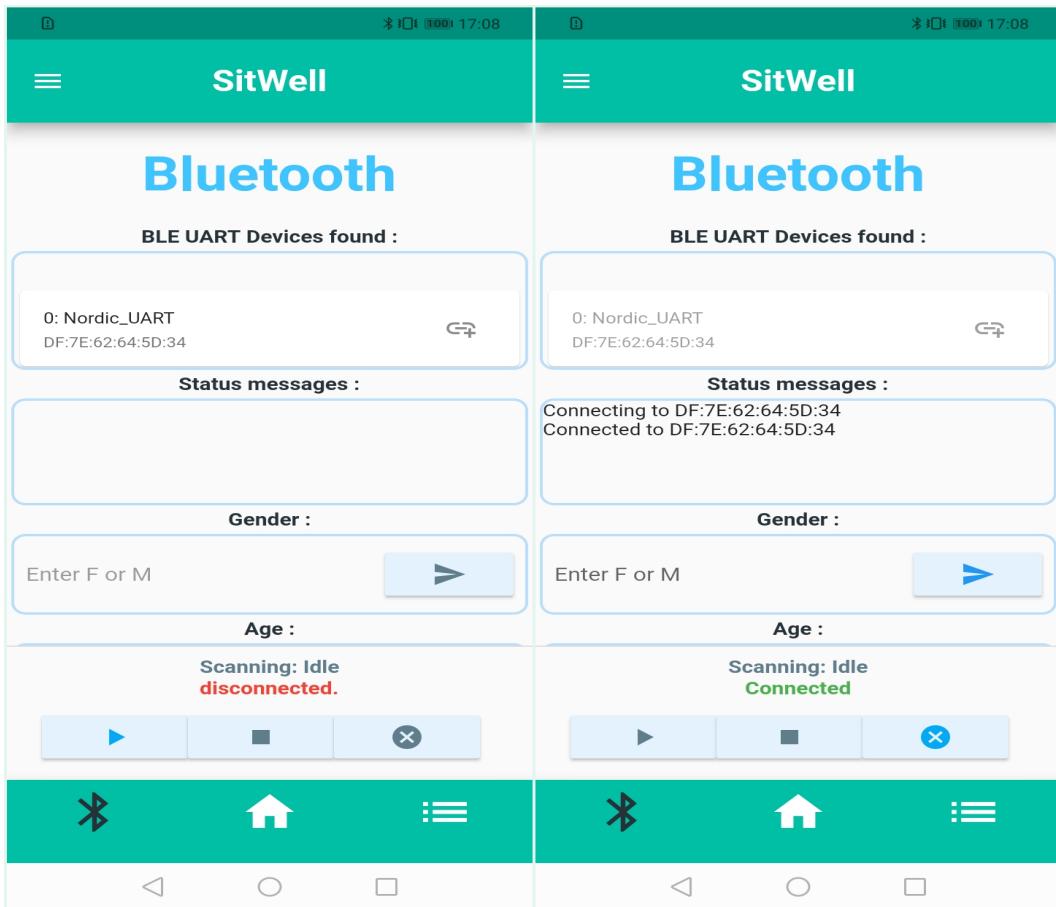


FIGURE 5.30 – Page de connexion BLE-UART(scan et connexion avec des périphériques)

la figure 5.30 illustre la fonctionnalité de recherche des périphériques qui implémentent le service UART-BLE, ceci en effectuant un simple clique sur le bouton de lecture en bas à gauche.le bouton d'arrêt pour arrêter une analyse en cours. Lorsqu'un service UART BLE est trouvé, il s'affiche en haut. Lorsque l'utilisateur clique sur le périphérique (le microcontrôleur) trouvé à droite, une connexion est en cours d'établissement comme on peut le voir dans le volet Messages d'état.Une connexion existante peut être fermée à l'aide du bouton annuler.

IMPLEMENTATION ET RÉSULTATS

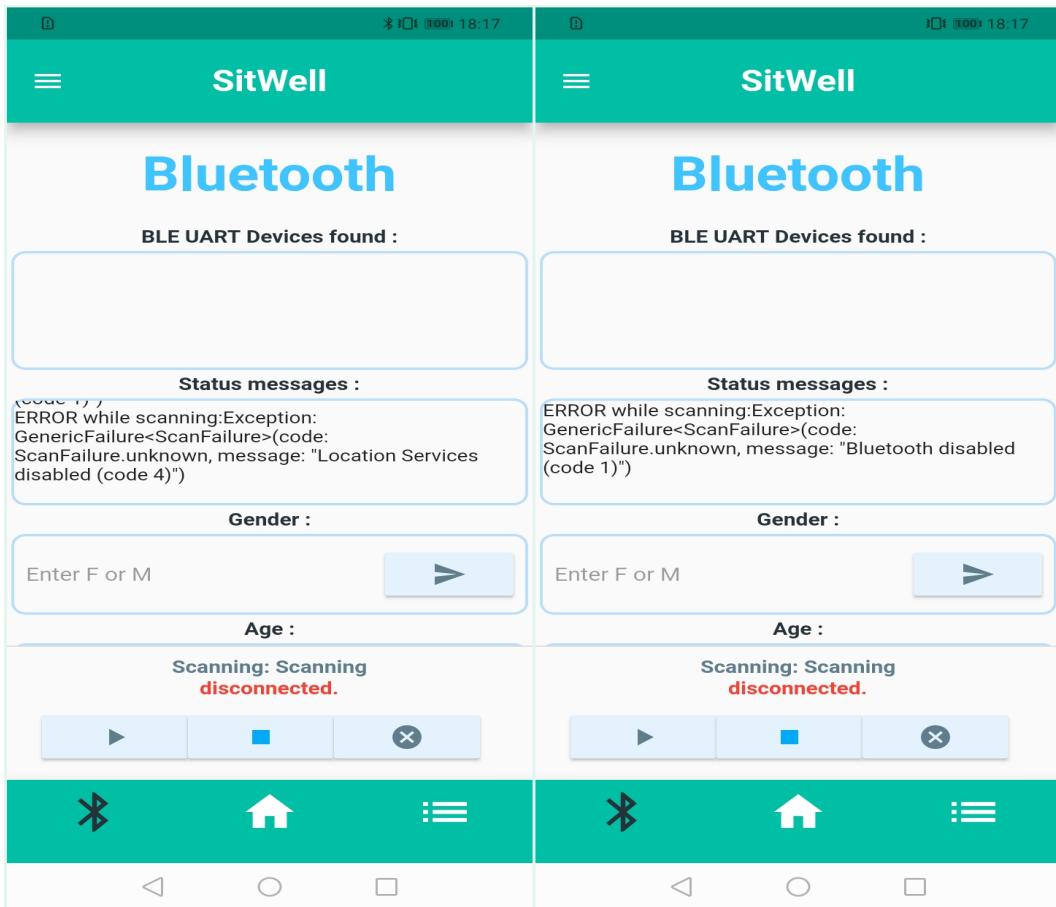


FIGURE 5.31 – Page de connexion BLE-UART (cas d'erreur)

La figure 5.31 montre que lorsque le Bluetooth ou le service de localisation sont désactivés un message d'erreur s'affiche dans l'onglet messages reçus.

IMPLEMENTATION ET RÉSULTATS

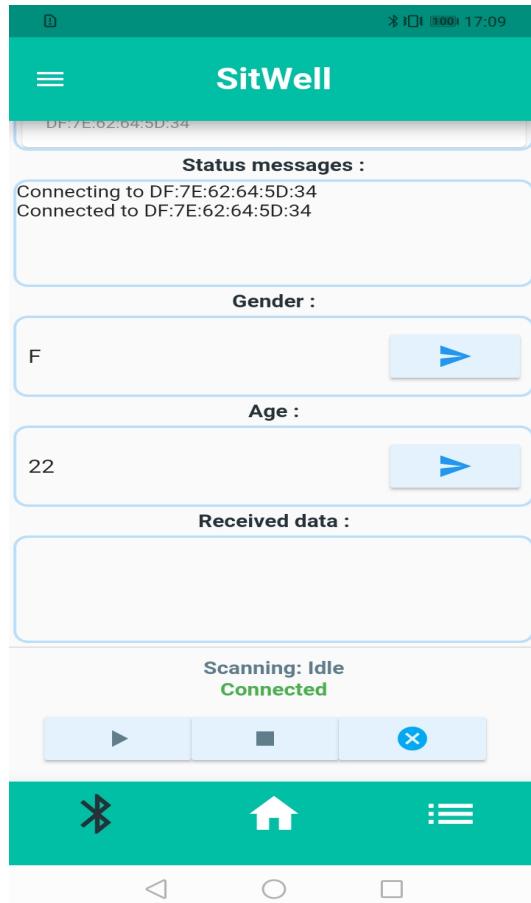


FIGURE 5.32 – Page de connexion BLE-UART(insertion de l'age et le genre)

La figure 5.32 présente l'action de l'insertion du sexe et de l'âge de l'utilisateur, ces paramètres vont être envoyées au périphérique BLE pour qu'ils soient traités par le modèle d'apprentissage automatique dans le pc.

IMPLEMENTATION ET RÉSULTATS

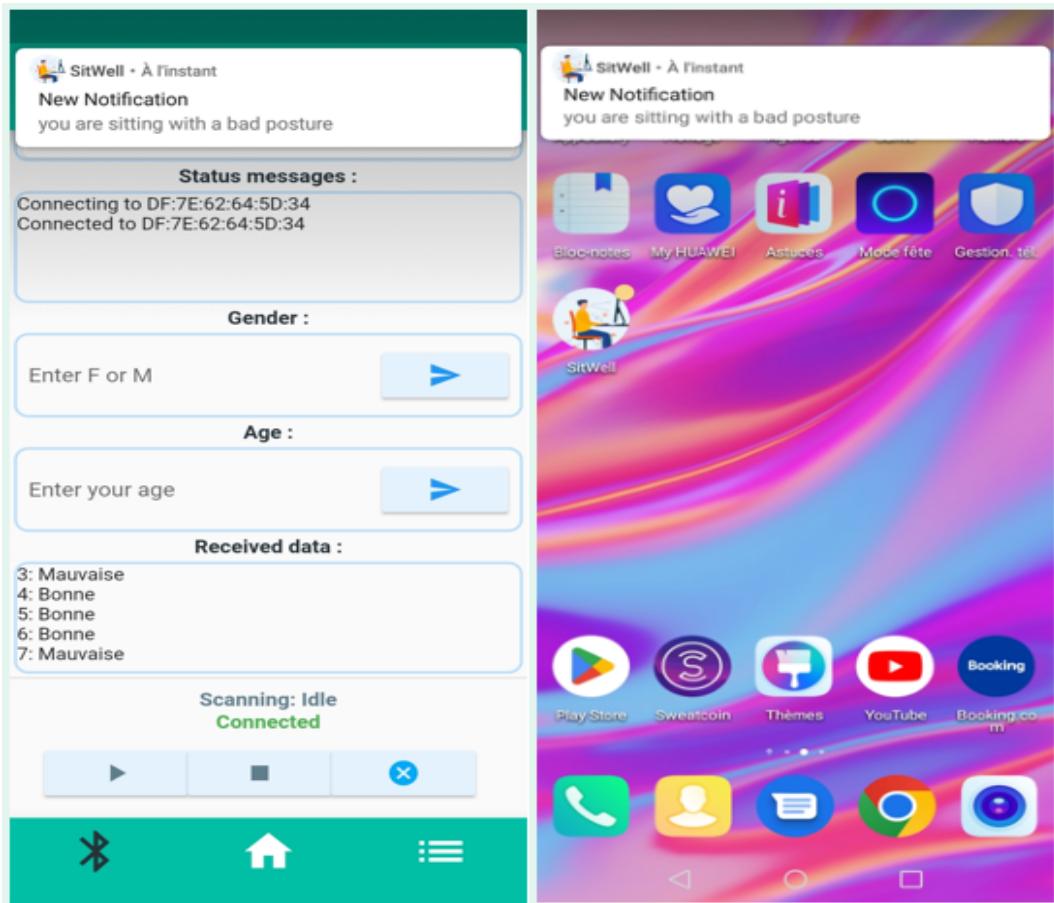


FIGURE 5.33 – Page de connexion BLE-UART(Notifications)

La figure 5.33 illustre l'affichage d'une notification en haut de l'écran en cas où la données reçus en temps réel dans l'onglet 'Received data' indique que l'utilisateur est dans une mauvaise position, ceci est valable même l'utilisateur est en dehors de l'application et même lorsque le téléphone soit verrouillé.

5.6.2.3 Page de conseils et d'exercices

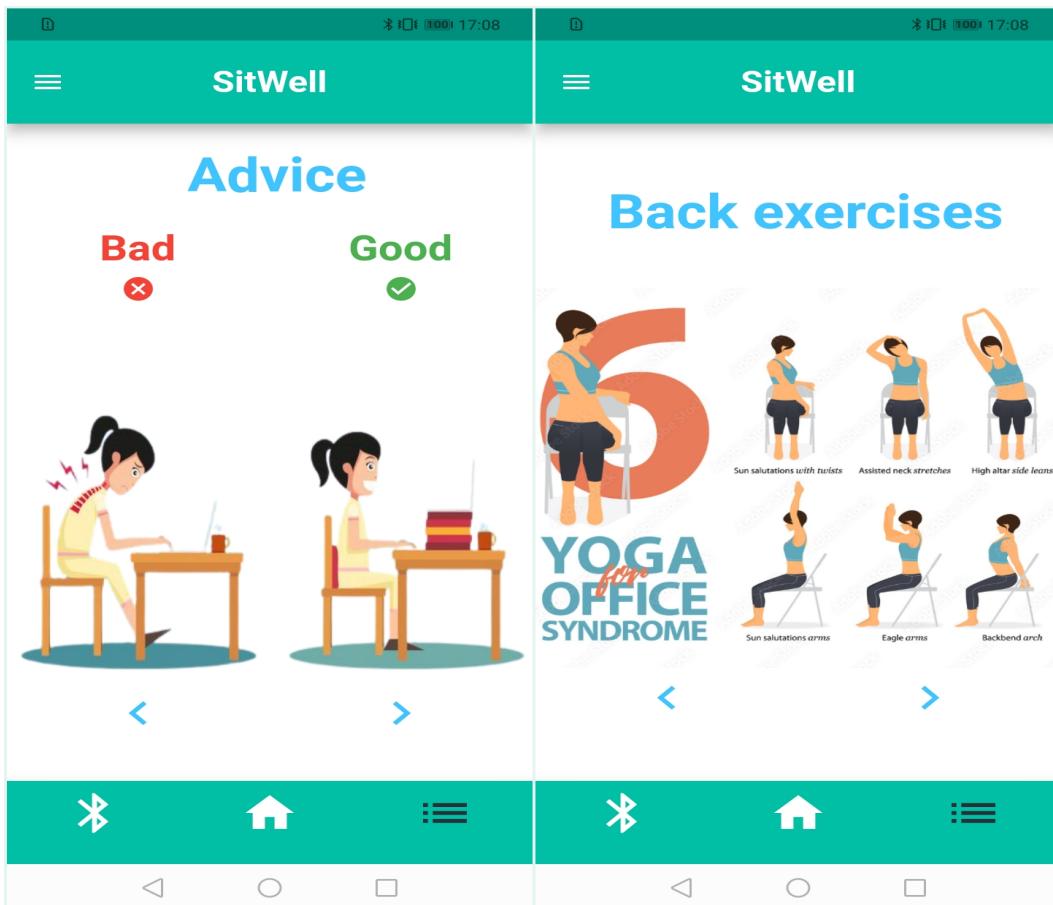


FIGURE 5.34 – Page de conseils et d'exercices

La figure 5.34 présente le contenu de la page conseils et exercices en effet cette dernière contient des menus déroulant illustrant les différentes postures inadéquates et leur correcte forme ainsi que de exercices physiques à suivre pour soulager le mal du dos.

5.7 Analyse des résultats

Lors de la mise en œuvre de notre prototype, plusieurs tests ont été effectués pour chaque sous-système afin de voir les résultats de toutes les étapes, comme indiqué dans ce chapitre. Ces tests nous ont montré des résultats satisfaisants pour chaque étape, ce qui nous a permis de valider chacun des sous-systèmes et de commencer l'assemblage et les tests du système final. Pour nos tests du système final, nous avons implémenté l'algorithme des K plus proches

IMPLEMENTATION ET RÉSULTATS

voisins (KNN) pour une classification binaire (bonne posture, mauvaise posture) car est il est facile, léger, pas compliqué, pas gourmant en terme de mémoire et de ressources, adapté aux exigences du système en terme de temps et de justesse.

Nous avons effectué plusieurs mesures de la contraction musculaire pour des différentes personnes travaillant dans un bureau et en faisant plusieurs positions. Les résultats que nous avons obtenus lors de ces mesures, comme nous l'avons vu sur les différents graphiques, étaient pertinents puisqu'ils montraient que le système pouvait mesurer la contractions musculaires et définir la forme de la colonne vertébrale.

D'autre part, l'application mobile a montré son efficacité pour afficher l'état de la posture et alerter l'utilisateur en temps réel et faire des mises à jour en temps réel avec de bonnes performances et une latence réduite.

Enfin, après tous ces processus de test et d'évaluation, nous sommes arrivés à des résultats prometteurs qui peuvent être validés, donc notre système peut également être validé.

5.8 CONCLUSION

A travers ce dernier chapitre, nous avons d'abord présenté le fonctionnement du processus de collecte de données en effectuant d'abord des tests du capteur EMG et de conversion du signal représentant la contraction musculaire, en exposant ensuite l'assemblage du circuit. Après avoir enregistrer l'activité musculaire de plusieurs individus nous avons présenté la phase de prétraitement des données récoltées, la phase de création de plusieurs modèles de classifications à savoir MLP, SVM, KNN, DT, RF et de comparaison de ces derniers et le choix de celui à utiliser pour l'implémentation en temps réel. De plus, nous avons détaillé les communications entre tous les acteurs du système. Enfin, nous avons décrit précisément les fonctionnalités de l'application mobile et son interface graphique . Tous ces résultats nous ont amenés à conclure que ce prototype pourrait être déployé comme un système de solution, et pourrait être vraiment efficace lorsqu'il est utilisé à des fins médicales. Nous sommes maintenant en mesure de le valider.

CONCLUSION GÉNÉRALE

Dans le cadre de notre PFE, nous avons été invités à proposer une solution permettant la conception et la réalisation d'un système de détection de mauvaise posture assise à l'aide des capteurs d'électromyographie en se basant sur l'intelligence artificielle.

Pour cadrer le contexte de notre étude et ses frontières, nous avons proposé une solution qui consiste en les phases suivantes :

La première phase consiste en la collecte de données en utilisant des capteurs EMG placés sur le muscle trapèze des individus. Afin de mettre en place notre ensemble de données d'entraînement nous avons enregistré l'activité musculaire de plusieurs personnes des deux sexes âgés de 18 à 45 ans dans des différentes positions assises à savoir position correcte , incliné vers l'avant , incliné à droite, incliné vers la gauche .La durée d'enregistrement est d'une minute pour chacune de ces dernières. Les données récoltées vont être numérisées au niveau d'un microcontrôleur et envoyées vers un pc via une communication UART . Ensuite un prétraitement est effectué pour extraire les caractéristiques à utiliser comme données d'entraînement dans plusieurs modèles de classification à savoir un modèle d'apprentissage en profondeur (perceptron multicouche) et quatre modèles d'apprentissage automatique (SVM, KNN, DT,RF) .

Les modèles sont évalués principalement sur deux métriques à savoir la justesse et la temps d'execution en tenant en compte que 20% de l'ensemble de données sont pour le test. Ces modèles sont utilisés pour une classification binaire et multiclasse.

Après comparaisons entre ces modèles on a conclu que le meilleur est KNN avec une justesse de 91% avec un temps d'exécution de 0.0066 ms. Ceci sera implémenté en temps

CONCLUSION GÉNÉRALE

réel car il répond au besoin de notre système embarqué en termes de limitation de ressources temporelle, spatiale et énergétique.

Lorsque la classe de la posture est bien définie, le pc envoie ce résultat à une application mobile via un service personnalisé uart de Bluetooth à énergie réduite .

L'application sert d'un système de rétroaction qui envoie des notifications en temps réel lorsque l'utilisateur est assis dans une position inadéquate ,l'alerte apparaît même dans le cas où le téléphone est en mode verrouillé ou lorsque le client est en dehors de l'application.

Pour finir, plusieurs perspectives sont envisageables, nous citons parmi elles :

- Effectuer des évaluations plus approfondies de notre système en considérant un ensemble de données plus volumineux c'est-à-dire augmenter le nombre de personnes et varier dans l'âge et le sexe.
- Enrichir notre système en utilisant plus de capteurs EMG et utiliser la propriété sans fil de ces derniers.
- Enregistrer l'historique de la position de la personne pendant la journée dans le cloud et afficher des statistiques.
- Utiliser un microcontrôleur Raspberry et implémenter les modèles d'apprentissage dessus afin d'éviter l'utilisation du PC.

BIBLIOGRAPHIE

- [1] F. Tlili, R. Haddad, Y. Ouakrim, R. Bouallegue, and N. Mezghani, “A review on posture monitoring systems,” in 2018 International Conference on Smart Communications and Networking (SmartNets), pp. 1–6, 2018.
- [2] L. Simpson, M. M. Maharaj, and R. J. Mobbs, “The role of wearables in spinal posture analysis : a systematic review,” BMC musculoskeletal disorders, vol. 20, no. 1, pp. 1–14, 2019.
- [3] Définition d'apprentissage supervisé <https://www.journaldunet.fr/web-tech/guide-de-l'intelligence-artificielle/1501311-apprentissage-supervise-definition-et-exemples-d-algorithme/> consulté le 23/10/2021
- [4] Type de données <https://www.techopedia.com/definition/33279/test-set>, consulté le 23/10/2021
- [5] Etapes de machine learning <https://machinelearnia.com/apprentissage-supervise-4-etapes/> consulté le 23/10/2021
- [6] Regression vs Classification <https://sanalyticsinsights.io/5-apprentissage-supervise/> consulté le 26/10/2021
- [7] Fix, E. and Hodges, J.L. (1951) Discriminatory Analysis, Nonparametric Discrimination : Consistency Properties. Technical Report 4, USAF School of Aviation Medicine, Randolph
- [8] Knn Agorithm <https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning/> consulté le 26/10/2021
- [9] What are BLE Central and Peripheral ?, <https://www.feasycom.com/blog-What-are-BLE-Central-and-Peripheral.html>.

BIBLIOGRAPHIE

- [10] Chervonenkis, A. Y. (2013) ‘Early history of support vector machines’, in Empirical Inference : Festschrift in Honor of Vladimir N. Vapnik. doi : 10.1007/978-3-642-41136-6-3.
- [11] P. Du Jardin, "Prévision de la défaillance et réseaux de neurones : L'apport des méthodes numériques de sélection," PhD, Université de Nice, Sophia-Antipolis, 2007.
- [12] M. T. Hagan, H. B. Demuth, M. H. Beale, and O. De Jesús, Neural Network Design, 2nd ed., 1996.
- [13] yahya tawil, Bluetooth Low Energy (BLE), <https://atadiat.com/en/e-bluetooth-low-energy-ble-101-tutorial-intensive-introduction/>, 2018.
- [14] Suresh P, Daniel J V, Aswathy R H 2014 A state of the art review on the Internet of Things (IoT) History, Technology and fields of deployment
- [15] Lampkin V et al. 2012 Building smarter planet solutions with MQTT and IBM WebSphere MQ telemetry IBM, ITSO
- [16] Colitti W, Steenhaut K, De Caro N 2011 Integrating Wireless Sensor Networks with the Web Proc. IP+SN Chicago, USA
- [17] Vergara E J, Prihodko M, Nadjm-Tehrani S 2013 Mobile Location Sharing : An Energy Consumption Study e-Energy pp 289-290
- [18] De Caro N, Colitti W, Steenhaut K, Mangino G, Reali G 2013 Comparison of Two Lightweight Protocols for Smartphone-based Sensing IEEE 20th Symposium on Communications and Vehicular Technology in the Benelux (SCVT) Namur pp. 1-6
- [19] R A Atmoko et al 2017 , Acquisition de données IoT en temps réel à l'aide du protocole MQTT J. Phys. : Conf. Ser. 853 012003
- [20] Dong-Jun Shin, Min-Sang Kim, Wook Song, Se Dong Min, Min Hong, “Implementation of Sitting Posture Monitoring System with Kinect,” International Conference on Multimedia and Ubiquitous Engineering, pp. 144-150, May 2017
- [21] Roh J, Park HJ, Lee KJ, Hyeong J, Kim S, Lee B., “Sitting Posture Monitoring System Based on a Low-Cost Load Cell Using Machine Learning,” Sensors (Basel). 2018.
- [22] Jullia Birsan, Diana Stavarache, Maria-Iuliana Dascalu, Alin Moldoveanu, “SpiMO - Sitting Posture Monitoring System,” RiCHI 2017 Proceedings, 2017.

BIBLIOGRAPHIE

- [23] Jingyuan Cheng, Bo Zhou, Mathias Sundholm, Paul Lukowicz, “Smart chair : What can simple pressure sensors under the chairs legs tell us about user activity ?,” IARIA XPS Press, 2013.
- [24] Wai Yin Wong, Man Sang Wong, “Trunk posture monitoring with inertial sensors,” Eur Spine J, 2008.
- [25] Q. Wang, W. Chen, A.A.A. Timmermans, C. Karachristos, J.B. Martens, P. Markopoulos, “Smart Rehabilitation Garment for posture monitoring,” 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2015.
- [26] Maheswaran Shanmugam, Sunilvignesh Nehru, Sathesh Shanmugam, “A wearable embedded device for chronic low back patients to track lumbar spine position,” Biomed Res 2018 Special Issue, 2018.
- [27] Manju Gopinath, Angeline Kirubha, “Real Time Monitoring of Posture to Improve Ergonomics,” Journal of Biomedical Engineering and Medical Imaging, 2015.
- [28] Sardini, auro Serpelloni and Viviane Pasqui, “Wireless-wearable T-shirt for posture monitoring during rehabilitation exercises,” IEEE Transactions on Instrumentation and Measurement, pp. 439-448, 2015.
- [29] L.E. Dunne, P. Walsh, B. Smyth, and B. Caulfield, “A System for Wearable Monitoring of Seated Posture in Computer Users,” 4th International Workshop on Wearable and Implantable Body Sensor Networks (BSN 2007), pp 203-207, 2007.
- [30] Anindya Nag, Subhas Chandra Mukhopadhyay, Jrgen Kosel, “Wearable Flexible Sensors : A Review,” IEEE Sensors Journal, pp. 3949 - 3960, May 2017.
- [31] Giovanni Saggioab, Giancarlo Orengo, “Flex sensor characterization against shape and curvature changes,” Sensors and Actuators A Physical, pp. 221-231, April 2018.
- [32] Mayo Foundation for Medical Education and Research (MFMER) : Electromyography (EMG), <https://www.mayoclinic.org/tests-procedures/emg/about/pac-20393913>, 2019, consulté le 25/02/2022

BIBLIOGRAPHIE

- [33] Slavomir Matuska, Martin Paralic, Robert Hudec, « A Smart System for Sitting Posture Detection Based on Force Sensors and Mobile Application », Mobile Information Systems, vol. 2020, Id d'article 6625797, 13 pages, 2020. <https://doi.org/10.1155/2020/6625797>
- [34] Roland Zemp, Matteo Tanadini, Stefan Pluss, Karin Schnüriger, Navrag B Singh, William R Taylor, and Silvio Lorenzetti. Application of machine learning approaches for classifying sitting posture based on force and acceleration sensors. BioMed research international, 2016, 2016.
- [35] Ma et al., 2017, Congcong Ma, Wenfeng Li, Raffaele Gravina, and Giancarlo Fortino. Posture detection based on smart cushion for wheelchair users. Sensors, 17(4), 2017.
- [36] Jongryun Roh, Hyeong-jun Park, Kwang Jin Lee, Joonho Hyeong, Sayup Kim, and Boreom Lee. Sitting posture monitoring system based on a low-cost load cell using machine learning. Sensors, 18(1), 2018.
- [37] Bourahmoune, K., Amagasa, T. (2019, août). Formation de posture alimentée par l'IA : Application de l'apprentissage automatique dans la reconnaissance de posture assise à l'aide du coussin intelligent LifeChair. Dans IJCAI (pp. 5808-5814).
- [38] Paul Rosero-Montalvo, Daniel Jaramillo, Stefany Flores, Diego Peluffo, Vanessa Alvear , Milton Lopez, Human Sit Down Position Detection Using Data Classification and Dimensionality Reduction ,Advances in Science, Technology and Engineering Systems Journal Vol. 2, No. 3, 749-754 (2017) <https://dx.doi.org/10.25046/aj020395>
- [39] Kim, Yong M., Youngdoo Son, Wonjoon Kim, Byungki Jin et Myung H. Yun. 2018. « Classification of Children's Sitting Postures Using Machine Learning Algorithms » Applied Sciences 8, no. 8 : 1280. <https://doi.org/10.3390/app8081280>
- [40] Cai W, Zhao D, Zhang M, Xu Y, Li Z. Amélioration de l'algorithme d'apprentissage non supervisé basé sur une carte auto-organisée pour le système de reconnaissance de posture assise. Capteurs. 2021 ; 21(18) :6246. <https://doi.org/10.3390/s21186246>
- [41] Ahmad, J.; Sidén, J.; Andersson, H. A Proposal of Implementation of Sitting Posture Monitoring System for Wheelchair Utilizing Machine Learning Methods. Sensors 2021, 21, 6349. <https://doi.org/10.3390/s21196349>

BIBLIOGRAPHIE

- [42] Sinha, V.K.; Patro, K.K.; Pławiak, P.; Prakash, A.J. Smartphone-Based Human Sitting Behaviors Recognition Using Inertial Sensor. Sensors 2021, 21, 6652. <https://doi.org/10.3390/s21196652>
- [43] Kulikajevas A, Maskeliunas R, Damaševičius R. 2021. Détection de la posture assise à l'aide de la composition d'images hiérarchiques et de l'apprentissage profond. PeerJ Informatique 7 :e442 <https://doi.org/10.7717/peerj-cs.442>
- [44] H. K. Jang, H. Han et S. W. Yoon, « Comprehensive Monitoring of Bad Head and Shoulder Postures by Wearable Magnetic Sensors and Deep Learning », dans IEEE Sensors Journal, vol. 20, no. 22, pp. 13768-13775, 15 Nov.15, 2020, doi : 10.1109/JSEN.2020.3004562.
- [45] Byeon Y-H, Lee J-Y, Kim D-H, Kwak K-C. Reconnaissance de posture à l'aide de modèles d'ensemble profonds dans divers environnements domestiques. Sciences appliquées. 2020 ; 10(4) :1287. <https://doi.org/10.3390/app10041287>
- [46] Alexandra Kanaris, Katherine Lee et Zoë Roecker . One Thing Straight : A Posture Monitoring and Alert System.2017 <https://sites.duke.edu/atdesign/2017/10/17/one-thing-straight-a-posture-monitoring-and-alert-system/> consulté le 01/02/2022
- [47] What is EMG sensor, Myoware and How to use with Arduino, <https://www.seeedstudio.com/blog/2019/12/27/what-is-emg-sensor-myoware-and-how-to-use-with-arduino>, 2019
- [48] Park, S.-y. et Yoo, W.-g. (2012), Effect of EMG-based Feedback on Posture Correction during Computer Operation. Journal of Occupational Health, 54 : 271-277. <https://doi.org/10.1539/joh.12-0052-OA>
- [49] "Lost Worktime Injuries and Illnesses Characteristics," US Bureau of Labor Statistics, 2005.
- [50] Jorge E. Caviedes «Wearable Sensor Array Design for Spine Posture Monitoring During Exercise Incorporating Biofeedback »
- [51] Hung, Kevin Lee, Tsz-Chung Leung, Ho-Wai. (2016). Investigation of sEMG patterns for detection of poor trunk posture. 1-4. 10.1109/ICCE-China.2016.7849774.
- [52] [https://www.electronicaembajadores.com/Datos/pdf1/ss/ssbi/ssbimu4.pdf?](https://www.electronicaembajadores.com/Datos/pdf1/ss/ssbi/ssbimu4.pdf)

BIBLIOGRAPHIE

- [53] Analog EMG Sensor by OYMotion from DFRobot Store,
https://wiki.dfrobot.com/Analog_EMG_Sensor_by_OYMotion_SKU_SEN0240.
- [54] Grove - EMG Detector by The Linux kernel is the main component of a Linux,
[https://wiki.seeedstudio.com/Grove-EMG_Detector.Surface_EMG_Sensor_by_Biometrics_ltd.html](https://wiki.seeedstudio.com/Grove-EMG_Detector_Surface_EMG_Sensor_by_Biometrics_ltd.html)
- [55] Arduino For Beginners, <https://www.makerspaces.com/arduino-uno-tutorial-beginners/>.
- [56] STM32-base, <https://stm32-base.org/guides/getting-started.html>.
- [57] nordicsemi, nRF52 DK, <https://www.nordicsemi.com/Products/Development-hardware/nrf52-dk>.
- [58] Successive approximation analog-to-digital converter by nordic semiconductor,
<https://infocenter.nordicsemi.com>.
- [59] arduino, <http://www.mon-club-elec.fr/pmwiki-reference-arduino/pmwiki.phpn=main.MaterielUno>, 2013.
- [60] <https://www.mouser.fr/ProductDetail/STMicroelectronics/STM32F303VCT6>.
- [61] <https://www.mouser.fr/ProductDetail/Nordic-Semiconductor/NRF52-DK> ?
- [62] Image classment des séries STM . disponible sur <http://www.xuanzhi-sh.com/page87> visité le 01.05.2022
- [63] Kit de développement de microcontrôleur MDK disponible sur
<https://www2.keil.com/mdk5/> visité le 01.05.2022
- [64] <https://www.passeportsante.net/fr/Maux/examens-medicaux-operations/Fiche.aspx?doc=electromyogramme> visité le 01.05.2022
- [65] Introduction à Bluetooth Low Energy GATT. <https://learn.adafruit.com/introduction-to-bluetooth-low-energy/gatt> visité le 25/08/2022
- [66] Nordic UART Service (NUS)
<https://developer.nordicsemi.com/nRF-Connect-SDK/doc/1.6.1/nrf/include/bluetooth/services/>

BIBLIOGRAPHIE

nus.html ?fbclid=IwAR0sKN7Ntj0ERqUgnqnGPN8744EFXgvfd3ovd-41FxApn2mI-lNy7

595IJc Consulté le 05/07/2022

- [68] Apprentissage non-supervisé : définition et algorithmes populaires
<https://www.journaldunet.fr/web-tech/guide-de-l-intelligence-artificielle/1501309-apprentissage-non-supervise/> ?fbclid=IwAR1GLQbhDuTLVziv1nMSSBCDZ0Dg3enFKFlwxoeDg8wcJay1Ci1BhPzug4M
- Consulté le 25/09/2022

- [69] Reinforcement Learning : Définition et application
<https://datascientest.com/reinforcement-learning> ?fbclid=IwAR1Za6taaLkSTxIxNwhtrLcWIY2hzrFG-YG2CcI-L5Rcljauogo0j829ftE

Consulté le 25/09/2022

- [70] Machine à vecteurs de support (SVM) : définition et cas d'usage
<https://www.journaldunet.fr/web-tech/guide-de-l-intelligence-artificielle/1501879-machine-a-vecteurs-de-support-svm-definition-et-cas-d-usage/>
- Consulté le 25/09/2022

- [71] Barkallah Eya. (2016). Conception et évaluation d'un instrument de mesure pour la détection des postures et des mouvements contraignants menant à l'apparition des troubles musculosquelettiques en milieu de travail à l'aide de réseaux de neurones. Mémoire de maîtrise, Université du Québec à Chicoutimi.

- [72] Apprentissage automatique <https://www.ibm.com/fr-fr/cloud/learn/machine-learning>
- Consulté le 25/09/2022

Conception d'un système de détection et d'alerte de mauvaise posture assise à l'aide des capteurs d'Electromyographie basé sur l'intelligence artificielle

OUANDJLI Lyna
KESSAISSIA Meriem

Résumé :

Les mauvaises postures sont associées au développement et à l'aggravation de troubles musculosquelettiques dont la nécessité de surveillance et de correction posturale en temps réel. Notre système de détection et d'alerte de mauvaise posture est composé de trois éléments principaux. Le premier composant comprend le capteur EMG qui collecte les informations de la contraction musculaire . Le second analyse les données collectées et décide du type de la posture, en effet après une comparaison entre cinq modèles de classification, les classificateurs spécifiés à être implémenté en temps réel est K-plus proches voisins car il s'est montré performant en terme de justesse et de temps d'exécution. Le troisième élément est un système de rétroaction à savoir une application mobile qui communique avec les autres composants du système via bluetooth à énergie réduite, le rôle de ce système est d'alerter l'utilisateur en cas de mauvaise posture.

Mots clés :

Troubles musculosquelettiques, capteurs EMG, microcontrôleur, collecte de données, analyse les données, apprentissage automatique, apprentissage en profondeur, classification de la posture, Bluetooth à énergie réduite, application mobile.

Abstract :

Poor postures are associated with the development and aggravation of musculoskeletal disorders including the need for real-time postural monitoring and correction.

Our system is made up of three main elements. The first component includes the EMG sensor which collects information of muscle contraction. The second analyzes the data collected and decides on the type of the posture, indeed after a comparison between five classification models, the classifiers specified to be implemented in real time is K-nearest neighbors because it has proven to be efficient in terms of accuracy and execution time. The third element is a feedback system, namely a mobile application the role of this system is to alert the user in the event of poor posture.

Key-words :

Musculoskeletal Disorders, EMG Sensors, Microcontroller, Data Collection, Data Analysis, Machine Learning, Deep Learning, Posture Classification, Low Energy Bluetooth, Mobile App.

BIBLIOGRAPHIE