UQAC
Université du Québec
à Chicoutimi

Reconnaissance des interactions tactiles autour d'une montre intelligente à partir des capteurs intégrés

par Koffi Deladem Moïse Etou

Mémoire présenté à l'Université du Québec à Chicoutimi en vue de l'obtention du grade de maîtrise ès sciences (M. Sc.) en informatique

Québec, Canada

RÉSUMÉ

Les montres intelligentes connaissent aujourd'hui une adoption croissante en tant qu'objets connectés portables, mêlant esthétique, suivi de santé, notifications et outils de productivité. Toutefois, malgré cette polyvalence, leur petit écran limite considérablement les modalités d'interaction. En particulier, la saisie tactile est entravée par la taille réduite de la surface d'affichage et l'obstruction visuelle causée par les doigts, ce qui nuit à l'efficacité des interactions.

Face à cette contrainte, de nombreux travaux de recherche ont tenté d'étendre les capacités interactives des montres à l'aide de dispositifs externes ou de capteurs supplémentaires. Bien que prometteuses, ces approches impliquent souvent un matériel spécialisé coûteux, encombrant, peu accessible ou encore implique une modification des montres existantes. Cela freine leur adoption en conditions réelles, notamment dans des contextes commerciaux ou auprès du grand public.

Dans ce contexte, notre projet propose une approche alternative, fondée uniquement sur les capteurs déjà embarqués dans les montres intelligentes actuelles (accéléromètre, gyroscope, capteur de pression, capteur de fréquence cardiaque, etc.). L'objectif est de détecter des gestes effectués autour de la montre et sur la peau (toucher, glissement, pression, etc.) sans ajouter de matériel, en utilisant des algorithmes d'apprentissage superviser pour reconnaître les interactions à partir des signaux capteurs. Cette méthode permettrait d'interagir avec la montre sans avoir à toucher son écran, élargissant ainsi l'espace d'interaction au bras et à la main.

Pour ce faire, nous avons développé une application mobile open-source de collecte multi-capteurs (tels que les capteurs PPG, SpO_2, environnementaux, etc.), compatible avec les montres Android Wear OS. Cette application permet de guider le participant à travers une expérimentation, d'enregistrer, d'annoter et d'exporter les données issues des capteurs. Un protocole de collecte rigoureux a été mis en place afin d'enregistrer une diversité de gestes autour de la montre, constituant ainsi un jeu de données structuré pour l'entraînement et l'évaluation de modèles de machine learning.

Bien que les performances de reconnaissance gestuelle obtenues dans cette étude restent limitées en raison d'un volume restreint de données, d'une palette de gestes peu distinctifs, de l'utilisation de modèles relativement simples et de la fréquence d'échantillonnage instable des montres, les retombées s'avèrent particulièrement significatives. Nous proposons un retour d'expérience approfondi sur les contraintes techniques propres aux plateformes Android Wear OS, telles que la variabilité de la fréquence d'échantillonnage, l'absence d'alignement temporel entre les capteurs et les interruptions causées par les mécanismes d'optimisation énergétique. Nous avons également mis en lumière les défis liés à leur exploitation (hétérogénéité des fréquences, désynchronisation, gestion des données manquantes). Ces aspects, encore rarement abordés dans la littérature, sont ici analysés, structurés et documentés, fournissant ainsi un socle pour de futurs travaux exploitant les capteurs embarqués des dispositifs Android.

Par ailleurs, l'application que nous avons développée représente une contribution concrète, réutilisable, libre, modulaire et sans dépendance matérielle externe. Elle facilite la collecte, l'annotation et l'exploitation de données multi-capteurs dans des scénarios expérimentaux reproductibles. Ce travail amorce ainsi une nouvelle direction dans le développement de systèmes de reconnaissance gestuelle, en exploitant la richesse des capteurs disponibles dans les dispositifs grand public.

En somme, notre étude ne se limite pas à la validation de la faisabilité d'un système de prédiction des gestes, ni à la mise en place d'une solution logicielle ouverte et légère pour la collecte de données gestuelles autour de la montre. Elle offre également un état des lieux critique des défis techniques propres aux dispositifs Android Wear OS, et propose un cadre méthodologique

structurant pour faire progresser la reconnaissance gestuelle vers des systèmes véritablement autonomes, fiables en conditions réelles et accessibles au plus grand nombre.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES ABRÉVIATIONS

API : Application Programming Interface (Interface de Programmation d'Application)
EMG : Electromyography (Électromyographie)
FBG : Full Bezel Glide (Glissement Complet sur la Bordure)
KNN : K-Nearest Neighbors (méthode des K Plus Proches Voisins)
KSPC : Keystrokes Per Character (Frappes Par Caractère)
PBG : Partial Bezel Glide (Glissement Partiel sur la Bordure)
PPG : Photoplethysmography (Photopléthysmographie)
SpO_2 : Peripheral capillary oxygen saturation (Saturation Pulsée en Oxygène)
SDK : Software Development Kit (Kit de Développement Logiciel)
SVM : Support Vector Machine (Machine à Vecteurs de Support)
WPM : Words Per Minute (Mots Par Minute)
XGBOOST : Extreme Gradient Boosting (Boosting par Gradient Extrême)

REMERCIEMENTS

Avant tout, Je remercie chaleureusement mon directeur de mémoire, Monsieur Pascal Fortin, pour sa bienveillance, sa patience, sa disponibilité et ses conseils avisés. Son encadrement inspirant a joué un rôle déterminant tout au long de ce travail. Sa confiance, dès notre première rencontre, a été pour moi un véritable moteur.

Je souhaite également exprimer ma reconnaissance à l'ensemble de mes enseignants, pour la qualité de leurs enseignements et leur engagement constant dans notre formation. Ils ont nourri ma réflexion et structuré ma démarche de recherche.

Un immense merci à mes parents Chocho Jibidar et Koffi Etou, pour leur amour inconditionnel et leur soutien sans faille à chaque étape de ma vie. À ma grande sœur Afi Delali Inda Etou, pour sa présence constante et son écoute patiente, même à distance, malgré le décalage horaire.

Je suis profondément reconnaissant envers Axel Levier, mes amis Noureddine Lourimi, Karim Abdel Rehim et Aminata Ouédraogo pour leur soutien, leurs encouragements et leur compréhension face à mes silences et mes absences. Une pensée particulière à Franck Gnaoré, dont la présence récente mais précieuse a été d'un grand réconfort, ainsi qu'à Ida Koffi, Leroy Abiguime et Justine Pakai, trois amis exceptionnels, dont les mots, ont souvent su apaiser mes doutes.

Je n'oublie pas les membres du laboratoire de recherche Lagora et mes camarades de parcours, pour leur solidarité, leurs conseils, et l'esprit d'entraide qui ont marqué ces années d'étude.

Enfin, Je tiens aussi à exprimer ma profonde gratitude à toutes celles et ceux qui ont contribué, de près ou de loin, à la réalisation de ce mémoire.

À toutes et à tous, merci du fond du cœur. Ce mémoire est aussi le vôtre.

INTRODUCTION

Le marché des montres intelligentes connaît une forte croissance [1]. Plus de 148,74 millions de personnes utilisaient des montres intelligentes en 2019. Des études estiment qu'il y aurait 230,85 millions d'utilisateurs en 2028, reflétant ainsi l'intérêt grandissant des consommateurs pour cette innovation. Malgré leur large adoption, ces dispositifs présentent plusieurs limites liées à la taille de leur écran, notamment la navigation dans une liste déroulante ou la saisie de texte [2][3][4].

La quête d'une utilisabilité optimale, d'une discrétion et la nécessité d'une disponibilité constante ont poussé de nombreux chercheurs à se tourner vers des techniques, telles que l'utilisation de périphériques externes, ou l'exploitation de l'espace disponible sur le bras. L'avantage de cette dernière méthode est que le bras est toujours disponible et ne nécessite pas de source d'énergie supplémentaire, contrairement aux accessoires externes qui requièrent leur propre alimentation et qui peuvent être facilement égarés. De plus, de nombreux travaux, tels que AuraSense [5] ou encore LumiWatch [6], qui, en plus d'exploiter l'espace disponible sur le bras, construisent ou améliorent des montres déjà existantes pour améliorer les interactions avec ces dispositifs. Toutefois, cette approche peut s'avérer coûteuse pour l'utilisateur final ou nécessiter l'ajout de composants rendant les dispositifs modifiés moins compacts et encombrants.

Considérant ces contraintes, nous nous intéressons donc dans ces travaux, comme dans la mise en place du Tapskin [7] qui utilise le microphone, à l'exploitation des capteurs déjà intégrés à la montre intelligente (ex. : capteurs de mouvement, pression, orientation, rythme cardiaque) et l'espace disponible sur le bras. Grâce au mode de fonctionnement des capteurs intégrés dans la montre, nous formulons l'hypothèse que la détection de gestes effectués sur la peau pourrait influencer les mesures collectées et permettre ainsi d'identifier le geste en question. Il est en effet raisonnable de penser que les capteurs déjà présents dans les montres intelligentes recèlent un potentiel important pour la reconnaissance d'entrées tactiles cutanées, sans nécessiter de dispositifs supplémentaires.

Notre étude vise donc principalement à étendre les capacités d'interaction des montres intelligentes disposant de capteurs, en exploitant l'espace disponible sur le bras. Les sous-objectifs consistent à explorer et concevoir de nouvelles méthodes d'interaction avec les montres intelligentes pour améliorer l'expérience utilisateur et développer un nouvel outil de collecte de données. Ainsi, pour ce faire, dans le chapitre un (1), nous passerons en revue les différentes méthodes d'interaction existantes, tant sur la peau qu'autour de la montre. Nous analyserons aussi les gestes courants d'interaction, tels que les tapotements, les balayages, les pincements et les rotations.

Les chapitres deux (2) et trois (3) seront consacrés à l'approche proposée dans le cadre de notre travail de recherche, à la méthodologie employée durant le projet, incluant les étapes de la collecte de données, la sélection des capteurs, ainsi que le processus d'analyse des gestes. Nous expliquerons les choix méthodologiques effectués et les défis techniques auxquels nous avons été confrontés. Ensuite, dans le chapitre quatre (4), nous détaillerons le processus de développement de l'application utilisée pour la collecte de données. Nous aborderons les technologies utilisées, telles que le langage de programmation, les cadriciels et l'architecture. Nous mettrons également en lumière les particularités de l'application en matière de fonctionnalité et les optimisations mises en place pour garantir la fiabilité des données. Des pistes d'amélioration pour les versions futures seront également proposées.

Dans le chapitre cinq (5), nous nous focaliserons sur la présentation des caractéristiques des données collectées. Le chapitre six (6) sera dédié à l'exploration des algorithmes d'apprentissage automatique utilisés, ainsi qu'à la présentation des résultats obtenus pour chacun d'eux. Le chapitre sept (7), quant à lui, portera sur l'analyse et la discussion des résultats précédemment présentés. Enfin, le dernier chapitre (Conclusion) proposera une synthèse de notre étude, en soulignant les limites, les contributions majeures tout en ouvrant des perspectives pour de futures recherches dans le domaine de l'interaction personne-machine, en particulier pour les dispositifs portables à écran réduit.

Ce mémoire apporte les contributions suivantes :

- Un nouvel outil de collecte de données portable compatible avec tous les appareils Android (téléphone et montre intelligente Wear OS). Cet outil permet une collecte manuelle et scénarisée limitant les besoins d'annotations post-collecte, contrainte encore présente dans les approches traditionnelles.
- Une exploration du potentiel des capteurs embarqués sur la Google Pixel Watch 3 afin de faire la reconnaissance de gestes en périphérie de la montre.

Ces contributions ouvrent de nouvelles perspectives quant à l'exploitation des capteurs embarqués dans les montres, non seulement en tant que dispositifs de collecte de données, mais aussi en intégrant d'autres types de capteurs que les capteurs inertiels jusqu'ici relativement peu utilisés dans les travaux de recherche portant sur l'interaction avec l'espace autour de la montre et la surface de la peau pour des interactions plus naturelles.

CHAPITRE 1 : REVUE DE LITTÉRATURE

Au fil des années, de nombreuses méthodes d'interaction avec les montres intelligentes ont été explorées, et de nouvelles approches continuent d'émerger. Pour mieux comprendre ces avancées, nous examinerons les recherches qui ont contribué à améliorer l'interaction avec ces dispositifs. Cette revue de littérature a pour but de mettre en évidence les stratégies développées pour pallier les limites des montres intelligentes, notamment celles liées à la taille réduite de leur écran. Elle offrira également un aperçu des différents modes d'interactions gestuelles, des types de gestes, des technologies, des algorithmes utilisés, ainsi que de leurs applications potentielles, tout en analysant les limites de ces techniques.

Pour mener à bien cette analyse, nous nous sommes appuyés sur des documents et articles disponibles dans des bases de données scientifiques fiables, telles que l'ACM et l'IEEE. Les termes de recherche, tels que « *smartwatch* », « *smart watch* », « *gesture recognition* », « *on-body interactions* », « *around-watch* » et « *interaction* », nous ont permis d'accéder à des documents pertinents. Parmi les articles identifiés, une soixantaine a été sélectionnée pour l'analyse. Nous avons décidé de classer ces articles par thématique et avons ainsi regroupé les travaux sur les interactions qui nous intéressent en trois grandes catégories.

La première catégorie concerne les interactions effectuées directement avec la montre (écran ou boîtier). Nous avons nommé cette dernière *« Interactions sur et avec la montre* ». La deuxième catégorie, « *Interactions gestuelles dans l'air et autour de la montre* », se concentre sur la reconnaissance des gestes réalisés dans l'espace proche de la montre, sans contact direct avec celle-ci ou la peau. Enfin, la troisième catégorie, « *Interactions autour de la montre et sur la peau* », regroupe les gestes réalisés sur la peau à proximité et sans contact direct avec la montre, mais permettant d'interagir avec elle. Maintenant que nous connaissons les différentes catégories qui nous intéressent et ce à quoi elles font référence, nous allons étudier les travaux qui ont marqué chacune d'elles.

1.1 INTERACTIONS SUR ET AVEC LA MONTRE

Traditionnellement, l'écran tactile demeure le moyen d'interaction le plus adopté pour les montres intelligentes, souvent utilisé pour la saisie de texte via des claviers virtuels [8]. Cependant, il présente des limites, notamment en matière de précision lors de la saisie de texte ou de la navigation dans des listes déroulantes, ce qui peut engendrer un certain inconfort pour l'utilisateur [2]. Bien que moins précis que d'autres moyens d'interaction, comme le cadran rotatif (Bezel Input), qui consiste à manipuler la bordure de la montre, l'écran tactile demeure, selon plusieurs études expérimentales [9], la méthode d'interaction privilégiée par les utilisateurs. Il s'agit également de l'approche la plus répandue dans les montres intelligentes disponibles sur le marché.

Malgré cette préférence des utilisateurs, l'interaction avec cadran rotatif est également bien reconnue dans le domaine de l'interaction humains-machines, comme en témoigne l'utilisation du concept BezelGlide [10]. Cette technologie vise à réduire l'occlusion de l'écran tout en permettant une interaction fluide avec les graphiques et les applications des montres intelligentes. Dans cet article, les chercheurs ont mené deux études auprès des utilisateurs : la première mesurait le niveau d'occlusion de l'écran lors de l'interaction avec le cadre de la montre, tandis que la seconde portait sur la création de deux systèmes d'interaction basés sur le glissement des doigts le long du cadre qui sont ; le « *Full BezelGlide* » (FBG) et le « *Partial BezelGlide* » (PBG). Lors d'une expérimentation visant à évaluer les performances des différentes techniques d'interaction, telles que le taux d'erreur et le niveau d'occlusion sans implémentation d'algorithmes complexes et en mettant uniquement l'accent sur la conception matérielle et l'expérience utilisateur, il ressort que le PBG, limité à certaines zones du cadre, a démontré une meilleure précision dans les interactions étudiées. Il a même surpassé le Shift [11], une méthode sans occlusion pour les appareils mobiles à écran tactile, ainsi que le FBG qui lui permet une interaction continue sur tout le cadre de la montre. Ainsi, ces résultats suggèrent que l'utilisation partielle du contour peut offrir un équilibre entre facilité d'utilisation et réduction de l'occlusion. Malgré cela, notons que les performances du BezelGlide peuvent être affectées en situation de mouvement. De plus, les recherches se sont concentrées sur des interfaces

simples, comme les graphiques, sans inclure d'autres éléments tels que du texte ou des icônes. Des travaux futurs pourraient explorer l'intégration de ces éléments pour enrichir l'expérience utilisateur.

Dans le prolongement des recherches visant à dépasser les contraintes d'occlusion liées à la petite taille des écrans des montres connectées, Gil et al. [12] proposent une approche d'identification des doigts utilisés pour interagir avec la montre, à partir des profils de contact tactile et des angles d'approche. L'idée consiste à associer des fonctions spécifiques à chaque doigt, dans le but d'élargir les possibilités d'interaction sans augmenter la taille de l'écran. Cette démarche s'appuie sur des travaux antérieurs [13], [14] ayant montré, notamment dans le contexte des tablettes ou des claviers physiques, que l'identification des doigts pouvait améliorer l'expérience utilisateur. Cependant, les technologies existantes permettant une telle identification sont encore peu adaptées, voire indisponibles, pour les montres intelligentes. Pour pallier cette limite, les auteurs ont mené deux études expérimentales reposant sur la collecte de données tactiles détaillées à partir d'un écran capacitif standard. Ces données incluent les coordonnées de contact, les formes des ellipses de contact et les angles d'approche des doigts. Trois doigts ont été considérés : le pouce, l'index et le majeur. L'identification a été réalisée à l'aide d'algorithmes d'apprentissage automatique, principalement des arbres de décision (Random Forest, Random Tree). Les résultats montrent une précision de classification élevée, atteignant 98 % dans des conditions où les participants adoptaient des poses de contact exagérées. En revanche, dans des conditions plus naturelles, les performances chutent, avec une précision moyenne autour de 70 à 79 %, variant selon le doigt et le modèle utilisé.

Les résultats indiquent que cette approche est suffisante pour des tâches simples ou peu fréquentes, mais moins adaptée aux interactions répétitives ou prolongées. Par exemple, le pouce et le majeur présentent des performances réduites pour les cibles de petite taille, alors que l'index demeure relativement stable. L'étude souligne également les limites physiques du format smartwatch, notamment la difficulté de capturer correctement les contacts proches des bords de l'écran. Ainsi, les éléments interactifs basés sur cette technologie devraient idéalement être situés loin des bords inférieurs et droits de l'écran. Les auteurs concluent en appelant à des recherches supplémentaires, notamment sur l'évaluation en conditions réelles d'usage, le développement de

capteurs plus réactifs, et l'exploration de gestes combinés. Ils proposent également des exemples d'interfaces exploitant l'identification des doigts, comme des icônes multifonctions (« tricons ») ou des claviers virtuels optimisés selon les doigts utilisés.

L'utilisation du bracelet de la montre pour la saisie de texte est une autre approche notable. Funk et al. [15] ont comparé un clavier linéaire et un clavier multitap, deux configurations d'alignement de claviers. Ils ont développé des prototypes de claviers virtuels positionnés sur le bracelet et ont réalisé des tests utilisateurs pour mesurer la vitesse de frappe (WPM) et le nombre de frappes par caractère (KSPC). L'incapacité de l'utilisateur à toucher tout le pourtour du poignet en regardant la montre a conduit à privilégier le côté du bracelet orienté vers le corps. Les utilisateurs ont tapé plus rapidement et avec moins d'erreurs avec le clavier multitap. Contrairement à d'autres travaux, aucune utilisation spécifique d'algorithmes complexes n'est mentionnée, l'étude s'est concentrée sur la conception d'interfaces et l'évaluation utilisateur. Bien que prometteuse, cette approche nécessite une modification des montres actuelles. L'intégration de capteurs supplémentaires peut augmenter les coûts et la complexité. Dans le futur, des matériaux conducteurs flexibles ou des technologies haptiques pourraient être explorés pour faciliter cette intégration sans compromettre le design ou le confort.

Yang et al. [16] proposent une technique innovante d'interaction à deux mains pour les montres intelligentes en utilisant des capteurs électromyographiques (EMG) pour reconnaître les postures de la main et exécuter divers types de commandes. Ils utilisent un bracelet MYO, captant des postures spécifiques de la main associées à des commandes distinctes. Ces postures sont illustrées à travers des applications de déverrouillage par mot de passe basé sur des motifs de posture et de contrôle d'appareils domestiques. Le bracelet MYO utilisé dans l'expérience ne reconnaît qu'un nombre limité de postures prédéfinies (comme la main ouverte, le poing fermé, ou l'inclinaison de la main à droite ou à gauche), ce qui restreint la variété d'interactions possibles. De plus, la reconnaissance des gestes repose sur la stabilité du capteur EMG et peut être affectée par des mouvements parasites ou des interférences musculaires, ce qui pourrait nuire à la précision dans des contextes d'utilisation quotidienne. Enfin, l'intégration de la technologie EMG pour des applications pratiques reste un défi,

notamment en termes de confort et de discrétion qui sont deux caractéristiques essentielles pour des dispositifs portables. Malgré ces limites, cette technique démontre la flexibilité et le potentiel de l'EMG pour enrichir l'interaction avec les montres intelligentes, et les auteurs envisagent, dans des travaux futurs, de reconnaître des postures plus complexes pour augmenter encore les possibilités d'interaction.

L'interaction tactile sur les montres intelligentes, bien qu'efficace, présente des limites liées à l'occlusion de l'écran et à la précision sur de petites surfaces. Les approches alternatives, telles que l'utilisation du cadran rotatif, l'identification des doigts ou les capteurs EMG, offrent des solutions pour pallier ces contraintes. Toutefois, ces approches nécessitent souvent l'intégration de capteurs ou de systèmes spécialisés qui rendent leur mise en œuvre complexe et plus coûteuse. Aussi, la préférence des utilisateurs pour les interactions directes avec l'écran souligne l'importance de concevoir des interfaces qui équilibrent innovation, discrétion et intuitivité.

1.2 INTERACTIONS GESTUELLES DANS L'AIR ET AUTOUR DE LA MONTRE

L'exploration des interactions gestuelles sans contact direct avec la montre a conduit à des approches innovantes. *Blowatch* **[17]**, par exemple, propose de souffler sur la montre pour effectuer des actions telles que régler le volume ou répondre à un appel. Cette méthode offre une interaction mains libres, évitant les problèmes d'occlusion liés aux petits écrans. Le système utilise des microphones supplémentaires pour détecter le souffle de l'utilisateur. Les variations de pression sonore captées par les microphones sont analysées pour identifier les actions correspondantes. Des algorithmes de traitement du signal audio sont employés pour distinguer le souffle des bruits ambiants. La mise en œuvre nécessite des modifications matérielles, comme l'ajout de microphones supplémentaires. De plus, l'absence d'évaluation de l'exactitude et de l'efficacité de la méthode limite sa validation. Les chercheurs prévoient d'intégrer des capteurs piézoélectriques pour améliorer la fiabilité face aux interférences environnementales, ce qui pourrait impliquer le développement d'algorithmes plus sophistiqués pour le filtrage du bruit.

Serendipity **[18]** utilise les capteurs d'une Samsung Galaxy Gear pour distinguer des mouvements de motricité fine, tels que pincer ou taper et frotter les doigts de la main où la montre est portée. Les auteurs ont collecté des données à partir de l'accéléromètre, du gyroscope et du capteur d'accélération linéaire à une fréquence de 50 Hz. Ils ont extrait des caractéristiques temporelles et fréquentielles des signaux, puis ont utilisé des algorithmes de classification, tels que les Machines à Vecteurs de Support (SVM), le classificateur Naive Bayes, la régression logistique et les K-Plus Proches Voisins (K-NN). Le score F1 moyen obtenu pour les gestes était de 87 %. Le système souffre d'un taux de faux positifs élevé en l'absence de geste d'activation. L'introduction d'un geste d'activation réduit ce taux, mais ajoute une complexité. Les variations de performance entre utilisateurs suggèrent la nécessité d'algorithmes adaptatifs ou d'un apprentissage personnalisé. Des techniques d'apprentissage profond pourraient être explorées pour améliorer la précision et la fiabilité.

Xu et al. [18] ont également utilisé les capteurs intégrés pour reconnaître trente-sept (37) gestes classés en mouvements du bras, de la main et des doigts. Les données des capteurs ont été collectées et des caractéristiques ont été extraites pour chaque geste. Les auteurs ont utilisé des classificateurs, tels que Naive Bayes, la régression logistique et les arbres de décision pour la classification des gestes. La régression logistique a obtenu la meilleure précision globale, atteignant jusqu'à 98 %. Les défis incluent le bruit des mouvements lors de gestes avec un bras libre et les variations individuelles. L'utilisation de techniques d'apprentissage profond, comme les réseaux neuronaux récurrents (RNN) ou les réseaux neuronaux convolutifs (CNN), pourrait améliorer la reconnaissance des gestes en capturant des caractéristiques plus complexes.

Enfin, *BiTipText* **[20]** propose une saisie de texte bimanuelle sur un clavier miniature au bout des doigts, permettant une entrée « eyes-free ». Les participants ont atteint une vitesse moyenne de 23,4 mots par minute avec un taux d'erreur non corrigé de 0,03 %. Dans cet article, Zheer et al. […] ont conçu un clavier virtuel réparti sur les bouts des doigts. Un système de suivi de mouvement a été utilisé pour capturer les tapotements des doigts, et un décodeur statistique basé sur un modèle de langage a été utilisé pour prédire les mots saisis, réduisant l'ambiguïté liée à la petite taille du clavier.

Bien que performant, ce système nécessite une familiarisation avec un dispositif non standard. Des études comparatives avec d'autres méthodes de saisie et des tests en conditions réelles pourraient aider à évaluer son adoption pratique.

Les interactions gestuelles dans l'air et autour de la montre offrent des alternatives prometteuses aux interactions tactiles traditionnelles. Elles étendent les capacités des montres intelligentes tout en corrigeant les limitations liées à la taille de l'écran. Les défis majeurs résident dans la robustesse des systèmes face aux variations individuelles et environnementales, ainsi que dans l'équilibre entre complexité et intuitivité.

1.3 INTERACTIONS AUTOUR DE LA MONTRE ET SUR LA PEAU

Les recherches récentes explorent l'utilisation de la peau comme surface d'interactions pour les montres intelligentes. *SkinTrack* **[21]** propose un suivi tactile continu sur la peau en utilisant un anneau émettant un signal électrique et un bracelet de capteurs. Le système repose sur la transmission d'un signal électrique à haute fréquence à travers la peau. Les capteurs du bracelet mesurent les différences de phase du signal pour déterminer la position du toucher. Un modèle mathématique est utilisé pour convertir ces mesures en coordonnées spatiales avec une erreur moyenne de 7,6 mm. Bien que non invasif et peu coûteux, le système nécessite le port d'un anneau, ce qui peut être contraignant pour certains utilisateurs en situation de handicap. Des recherches pourraient explorer des méthodes pour intégrer l'émetteur directement dans la montre ou utiliser des signaux bioélectriques naturels. L'optimisation des algorithmes de localisation pourrait également améliorer la précision.

SkinWatch **[22],** quant à lui, utilise les déformations de la peau sous la montre pour détecter des gestes, permettant une interaction multi doigts sans occlusion de l'écran. Cette méthode simplifie les interactions sur de petits écrans. Des capteurs de pression ou de déformation sont intégrés sous la montre pour détecter les mouvements de la peau causés par les gestes des doigts. Des algorithmes de reconnaissance de motifs analysent les signaux pour identifier les gestes effectués. Mais, il faut noter que la sensibilité aux mouvements involontaires et aux variations de la peau peut

affecter la précision. Des améliorations dans la détection des gestes et des algorithmes de filtrage pourraient renforcer la fiabilité.

TapSkin [5] est une technique innovante qui permet de reconnaître jusqu'à 11 gestes de tapotement sur la peau autour de la montre intelligente, en utilisant les capteurs inertiels (gyroscope et accéléromètre) et le microphone déjà intégrés dans les montres intelligentes. Tapskin exploite les variations des signaux acoustiques et inertiels générés par les tapotements pour distinguer les gestes. Pendant la phase expérimentale, les algorithmes de classification, basés sur des SVM, sont utilisés pour identifier les gestes avec une précision allant jusqu'à 97,32 %. La dépendance à la synchronisation audio et les interférences dans des environnements bruyants sont des défis majeurs. L'intégration de capteurs supplémentaires ou le développement d'algorithmes de traitement du signal plus fiable pourraient atténuer ces problèmes.

LumiWatch **[6]** est un prototype capable de projeter des graphiques interactifs sur la peau, transformant le bras en surface tactile. Avec une surface interactive de 40 cm², il offre un espace d'interaction largement supérieur à celui des écrans traditionnels. Le dispositif intègre un projecteur laser et un capteur de profondeur tel qu'une caméra infrarouge pour suivre les mouvements des doigts sur la peau. Des algorithmes de calibration géométrique corrigent les distorsions dues à la surface courbe du bras. Le système reconnaît les touches avec une erreur moyenne de positionnement de 7,2 mm. Les défis dans cette étude incluent l'étalonnage sur une surface non plane et la gestion de l'éclairage ambiant. L'optimisation du projecteur et des capteurs de suivi est essentielle pour une adoption pratique.

Skin Buttons **[23]** utilise des projecteurs miniatures pour projeter des icônes tactiles sur la peau, élargissant la zone interactive sans augmenter la taille de l'appareil. Les icônes sont facilement reconnaissables et la détection tactile est précise. Les projecteurs laser projettent des icônes fixes sur la peau, et les capteurs tactiles détectent le contact lorsque l'utilisateur appuie sur ces icônes. La simplicité du système permet une faible consommation d'énergie. Comme avec LumiWatch, les conditions d'éclairage et la complexité du matériel sont des obstacles. Des solutions pour miniaturiser davantage le système et améliorer son efficacité énergétique seraient bénéfiques. La reconnaissance

tactile peut être affectée par les mouvements du poignet. L'utilisation de capteurs plus sensibles ou l'intégration de techniques de suivi pourrait améliorer la précision.

Pour finir, *AuraSense* [5] exploite la détection de champs électriques pour permettre des interactions enrichies autour des montres connectées, telles que la reconnaissance de gestes au-dessus de la montre ou l'activation de boutons virtuels sur la peau. Le dispositif utilise des électrodes capacitives en configuration « shunt-mode », qui mesurent les perturbations du champ électrique causées par la proximité de parties conductrices (ex. doigts). Les signaux sont ensuite interprétés à l'aide de modèles SVM à noyau RBF, permettant une classification et une régression en temps réel avec une faible latence. Néanmoins, des limites subsistent, notamment la sensibilité aux interférences électromagnétiques et à la dérive du signal au fil du temps. Des solutions telles que la normalisation des signaux ou un recalibrage dynamique sont envisagées pour améliorer la stabilité du système.

Les interactions autour de la montre et sur la peau ouvrent de nouvelles possibilités pour dépasser les limitations des écrans tactiles. En exploitant la peau comme surface interactive, ces approches offrent des méthodes innovantes pour enrichir l'expérience utilisateur. Les défis technologiques et ergonomiques restent cependant à surmonter pour une intégration réussie dans des produits commerciaux.

1.4 CONCLUSION

La diversité des méthodes d'interaction explorées sur l'écran, autour de la montre, dans l'air ou sur la peau témoigne du dynamisme de la recherche dans le domaine des montres intelligentes. Chaque approche apporte des solutions pour compenser les contraintes de petite taille des dispositifs, tout en introduisant de nouveaux défis tant technologiques qu'ergonomiques.

Les technologies et algorithmes utilisés varient, allant des méthodes d'apprentissage automatique pour la reconnaissance de gestes aux techniques de traitement du signal pour l'analyse des données sensorielles. Les avancées dans les capteurs intégrés, les matériaux conducteurs et les modèles de machine Learning permettent d'envisager des interactions plus naturelles et intuitives,

même si les interactions tactiles traditionnelles restent prédominantes en raison de leur intuitivité et de l'habitude des utilisateurs.

Pour l'avenir, il est essentiel de poursuivre les recherches en intégrant les retours des utilisateurs, en améliorant les systèmes et en explorant la convergence des différentes méthodes. L'objectif principal sera de développer des montres intelligentes qui soient non seulement technologiquement avancées, mais aussi parfaitement adaptées à l'usage quotidien et aux attentes des utilisateurs, ce qui d'ailleurs nous pousse à apporter notre contribution à travers cette étude.

CHAPITRE 2 : APPROCHE PROPOSÉE

À la suite de l'exploration des travaux antérieurs et des différentes approches méthodologiques mobilisées par les chercheurs dans le domaine de la reconnaissance de gestes, ce chapitre est consacré à la présentation de l'approche que nous proposons dans le cadre de cette recherche.

2.1 PRINCIPES GÉNÉRAUX DE NOTRE APPROCHE ET POSITIONNEMENT PAR RAPPORT AUX TRAVAUX EXISTANTS

Dans le cadre de ce travail, nous proposons une approche expérimentale visant à détecter et reconnaître des gestes d'interaction effectués autour de la montre et sur la peau adjacente, plutôt que directement sur l'écran tactile. Nous avons choisi ce mode d'interaction, désigné par le terme anglophone « *around-device interaction* » ou encore *périmontre*, pour répondre aux limitations ergonomiques des écrans de petite taille, notamment l'occlusion du contenu par les doigts.

Cette orientation s'inscrit dans une volonté de repenser les modes d'interaction homme-machine en contexte portable, en exploitant le potentiel des capteurs embarqués pour étendre l'espace d'interaction au-delà de la surface de la montre elle-même. En ce sens, notre approche vise à capter et interpréter des gestes effectués dans la proximité immédiate de la montre (ex. : figure 1), que ce soit au-dessus, à côté ou directement sur la peau du poignet afin de déclencher des actions ou des commandes, sans contact avec l'interface visuelle.

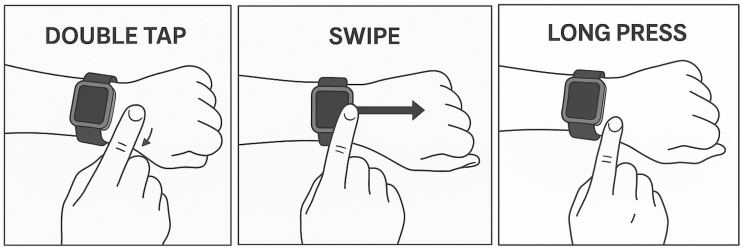

Figure 1 : Exemple de geste étudié dans notre étude

En nous appuyant sur les travaux existants majoritairement fondés sur des dispositifs spécialisés (tels que des unités de mesure inertielles externes ou des montres modifiées), nous reconnaissons l'efficacité de ces solutions pour la reconnaissance gestuelle. Toutefois, ces approches présentent des limites importantes en termes de coût, de généralisabilité dans les environnements de production des montres, et d'intégration dans des contextes d'usage réels. Nous avons donc opté pour une démarche méthodologique s'inspirant de ces travaux, mais fondée sur l'exploitation exclusive des capteurs embarqués dans les montres connectées commerciales, sans recours à des équipements externes. Cette spécificité permet une collecte de données en situation quasi réelle, tout en assurant la reproductibilité et la faisabilité technique du dispositif.

Ce travail s'inscrit ainsi dans une logique d'innovation pragmatique, en cherchant à rendre la reconnaissance de gestes non seulement fonctionnelle, mais également transposable à divers contextes applicatifs, tels que la navigation dans les menus, la saisie de texte, le déplacement d'éléments à l'écran, entre autres interactions. L'objectif est de proposer une solution accessible et adaptable à l'ensemble des montres connectées disponibles dans le commerce.

2.2 HYPOTHÈSE DE TRAVAIL ET JUSTIFICATION DE L'APPROCHE

Nous formulons l'hypothèse qu'il est possible de prédire, avec un niveau de précision satisfaisant, les gestes humains à partir des données issues de différents types de capteurs embarqués dans les montres intelligentes. Il ne s'agit donc pas uniquement de s'appuyer sur les capteurs inertiels classiques (tels que l'accéléromètre ou le gyroscope), mais également sur d'autres capteurs potentiellement présents, comme les capteurs de lumière, de pression ou encore de capteurs inertiels dérivés issus de la fusion des capteurs (tels que le capteur de gravité, l'accélération linéaire, le vecteur de rotation ou le vecteur de rotation pour jeux).

Cette hypothèse repose sur l'idée que, combinées et traitées de manière adéquate, ces données multisources permettent de capter des variations fines et distinctives associées à l'exécution de gestes spécifiques. Toutefois, la validité de cette hypothèse dépend de plusieurs conditions, comme la qualité de la collecte des données, leur alignement temporel, la représentativité des

fenêtres d’analyse, ainsi que le choix judicieux des caractéristiques extraites et des modèles d’apprentissage utilisés.

2.3 RETOMBÉES ANTICIPÉES

Grâce à la réalisation de ce projet, nous pensons être en mesure de mettre en place une technique d'interaction qui permettrait de reconnaître trois types de gestes effectués sur la peau autour d'une montre intelligente : les gestes associés à un pavé numérique, ceux correspondant à un pavé directionnel, ainsi que les gestes de compression.

Les utilisateurs aussi pourront interagir avec leurs montres intelligentes même à travers des obstacles, tels que des vêtements ou un manteau couvrant le bras. Ce qui pourrait être pratique, par exemple l’hiver pour défiler la musique de ses écouteurs sans sortir son téléphone et en pressant juste son bras à travers un manteau. Ou encore, décrocher un appel à partir d’un signe autour de la montre. Ces avancées dans les interactions pourront enrichir non seulement l'accessibilité et la facilité d'utilisation, mais ouvriront également des perspectives pour la mise en place de nouvelles interfaces utilisateur. Aussi, par le développement de l’application de collecte d’autres chercheurs pourront faire plus aisément la collecte de données issues des capteurs de téléphones et de montres intelligentes, offrant ainsi une alternative à la collecte de données par utilisation de capteurs propriétaires externes.

En permettant des commandes plus intuitives, moins restrictives et encombrantes, ce projet promet d'élargir les horizons de l'utilisation des appareils à petit écran, rendant ainsi la technologie encore plus naturelle et intégrée dans la vie quotidienne. Les implications de telles innovations pourraient transformer notre manière de concevoir et d'utiliser la technologie portable, en la rendant plus fluide et adaptée aux contextes et aux environnements variés.

CHAPITRE 3 : MÉTHODOLOGIE DE RECHERCHE

Après avoir présenté, dans le chapitre précédent, une revue des travaux existants ainsi qu'une réflexion approfondie sur les approches méthodologiques adaptées au domaine de l'interaction homme-machine, nous abordons, dans le présent chapitre, la méthodologie utilisée dans le cadre de cette recherche. Pour rappel, notre étude s'inscrit dans une volonté d'explorer de nouvelles formes d'interactions avec les dispositifs à petit écran, en particulier les montres intelligentes, en mobilisant l'espace corporel périphérique, notamment le bras et la peau environnante. L'objectif est de concevoir des gestes d'interaction naturels, intuitifs, et qui ne nécessitent aucun contact direct avec l'écran tactile.

La démarche méthodologique adoptée repose sur une approche expérimentale qui combine la collecte de données, l'analyse technique et la discussion des résultats. L'étude vise à générer des données multimodales riches, nécessaires à l'entraînement de modèles d'apprentissage automatique capables de reconnaître des gestes réalisés à proximité du dispositif. Parallèlement, elle cherche à documenter les aspects techniques du système développé et à évaluer son utilisabilité.

3.1 COLLECTE DE DONNÉES

Dans le cadre de cette recherche, la mise en œuvre d'une collecte de données primaires impliquant des participants humains s'est révélée indispensable. Cette démarche s'est accompagnée d'une demande d'autorisation éthique déposée auprès du Comité d'éthique de la recherche de l'Université du Québec à Chicoutimi (CER-UQAC). L'approbation a été obtenue sous le numéro de dossier 2025-1891 (*conf : CERTIFICATION ÉTHIQUE*), permettant ainsi de garantir que l'ensemble des procédures respectait les normes en vigueur en matière de recherche avec des êtres humains. La validité des résultats repose en grande partie sur la qualité du protocole expérimental. Celui-ci a été élaboré avec rigueur afin de minimiser les biais et d'assurer une collecte de données aussi représentative que possible. Un total de dix-huit participants a été recruté pour l'étude. Leurs profils présentaient une certaine diversité en termes d'âge, de genre et de couleur de peau dans le but

d'introduire une variabilité suffisante dans les gestes enregistrés. Cette diversité est essentielle pour accroître la robustesse des modèles d'apprentissage, notamment dans des conditions d'utilisation réelles où les caractéristiques physiques des utilisateurs peuvent influer sur la performance de reconnaissance gestuelle.

Le protocole expérimental a également intégré une phase de familiarisation permettant aux participants de se former à la réalisation des gestes attendus. Cette étape visait à réduire les écarts liés à une mauvaise compréhension ou à une exécution incorrecte des mouvements. Par ailleurs, toutes les sessions de collecte ont été conduites selon des procédures strictement standardisées. L'objectif était de garantir une uniformité dans les conditions d'enregistrement, tout en limitant les effets d'apprentissage ou de contexte susceptibles d'altérer la fiabilité des données recueillies.

3.2 APPAREILS ET OUTILS

La phase de collecte s'est appuyée sur un ensemble d'outils technologiques spécifiquement sélectionnés et développés pour répondre aux exigences de l'étude. L'appareil central utilisé pour l'enregistrement des gestes était une montre connectée Google Pixel Watch 3 (Figure 2), portée au poignet par chaque participant. Ce modèle a été retenu en raison de sa stabilité, de sa capacité à fournir des données brutes de capteurs variés, équivalents à ceux que l'on retrouve dans les principales montres intelligentes du marché. Sa polyvalence en faisait un choix pertinent pour une étude centrée sur la reconnaissance de gestes complexes.

Figure 2 : Google Pixel Watch 3

Pour garantir des conditions d'enregistrement homogènes entre les participants, un bloc de calibrage de 2 millimètres (Figure 3) a été systématiquement utilisé. Placé entre la peau et la montre

lors du serrage du bracelet, ce dispositif permettait de standardiser l'ajustement du bracelet et d'assurer une surface de contact régulière, réduisant ainsi les variations liées à la position de la montre ou à la morphologie individuelle. Cette standardisation visait à limiter les biais liés à l'emplacement des capteurs, susceptibles d'affecter les caractéristiques des signaux recueillis.

Figure 3 : Bloc de calibrage

La réalisation des gestes a été encadrée par un système de guidage visuel installé sur un téléphone positionné devant chaque participant. Ce système présentait, pour chaque geste à effectuer, une illustration graphique du mouvement attendu, un court texte descriptif, ainsi que des repères temporels indiquant la durée de l'action et le moment précis de son exécution (Figure 6). Cette interface a été conçue pour offrir une expérience intuitive et accessible, facilitant la compréhension des instructions tout en assurant une exécution cohérente et synchronisée des gestes entre les différents participants.

Le développement d'une application mobile dédiée à la collecte de données constitue un autre pilier méthodologique de ce dispositif. L'application permet l'enregistrement synchronisé de flux sensoriels provenant des capteurs de la montre, organisés dans un format structuré, directement exploitable pour l'analyse et l'entraînement des modèles. Elle inclut également une fonctionnalité d'annotation permettant d'associer précisément chaque segment temporel à un geste donné, assurant ainsi la qualité de l'étiquetage des données.

L'ensemble du système expérimental a été conçu dans le respect des standards en interaction homme-machine. Le design de l'expérience, les modalités d'exécution, les outils d'enregistrement et les méthodes de contrôle qualité ont été rigoureusement définis, dans le double objectif de produire des résultats scientifiquement valides et de garantir une expérience utilisateur fluide et accessible.

3.3 PROCÉDURE

Avant le début de l'expérimentation, chaque participant a reçu un formulaire d'information et de consentement précisant les objectifs de l'étude, les modalités de participation, ainsi que les droits et responsabilités liés à leur implication dans la recherche. Un expérimentateur a été chargé de présenter ce document, de s'assurer de sa compréhension et de répondre à toute question éventuelle avant de recueillir le consentement libre, éclairé et signé des participants. Aussi, afin de s'assurer que le modèle de prédiction développé soit inclusif et adapté à tous les utilisateurs, un court questionnaire a été administré (l'échelle de Fitzpatrick) aux participants afin de mieux comprendre les caractéristiques de leur peau (conf : Annexe I).

Afin de garantir une exécution fiable des gestes étudiés, une courte séance de familiarisation a été organisée en amont de la collecte. Cette étape préparatoire permettait aux participants de se familiariser avec les mouvements attendus, d'intégrer les consignes gestuelles, et de se sentir plus à l'aise avec l'interface du dispositif expérimental. Cette phase a contribué à limiter la variabilité liée à l'inexpérience et à homogénéiser la qualité des données.

Lors de l'enregistrement des gestes, les participants suivaient l'information du geste affiché sur l'écran du téléphone. Un intervalle fixe de quinze (15) secondes était respecté entre chaque geste, afin d'éviter toute interférence dans les mesures successives et laisser le temps aux capteurs embarqués de se recalibrer automatiquement.

Dans certains cas, l'expérimentateur pouvait intervenir pour demander la répétition d'un geste jugé imprécis ou incomplet, garantissant ainsi une qualité optimale des données. Des consignes verbales pouvaient également être données pour ajuster l'exécution d'un mouvement spécifique ou tester la robustesse du système face à de légères variations gestuelles.

Les gestes étudiés, sont détaillés dans la section suivante. Ils ont été sélectionnés de manière à couvrir une diversité de formes et d'amplitudes gestuelles, afin de tester la flexibilité du système de reconnaissance dans différents contextes d'usage.

3.4 DESCRIPTION DES GESTES ETUDIÉS

Les gestes sélectionnés dans le cadre de cette étude (Figure 4) couvrent un éventail varié d'interactions sans contact réalisées au-dessus ou autour de la montre intelligente. Chacun a été défini selon des paramètres précis (durée, direction, surface d'interaction) afin de simuler différentes modalités d'usage. Ces gestes visent à évaluer la capacité du système à reconnaître des mouvements distincts et pertinents dans un contexte d'interaction gestuelle naturelle.

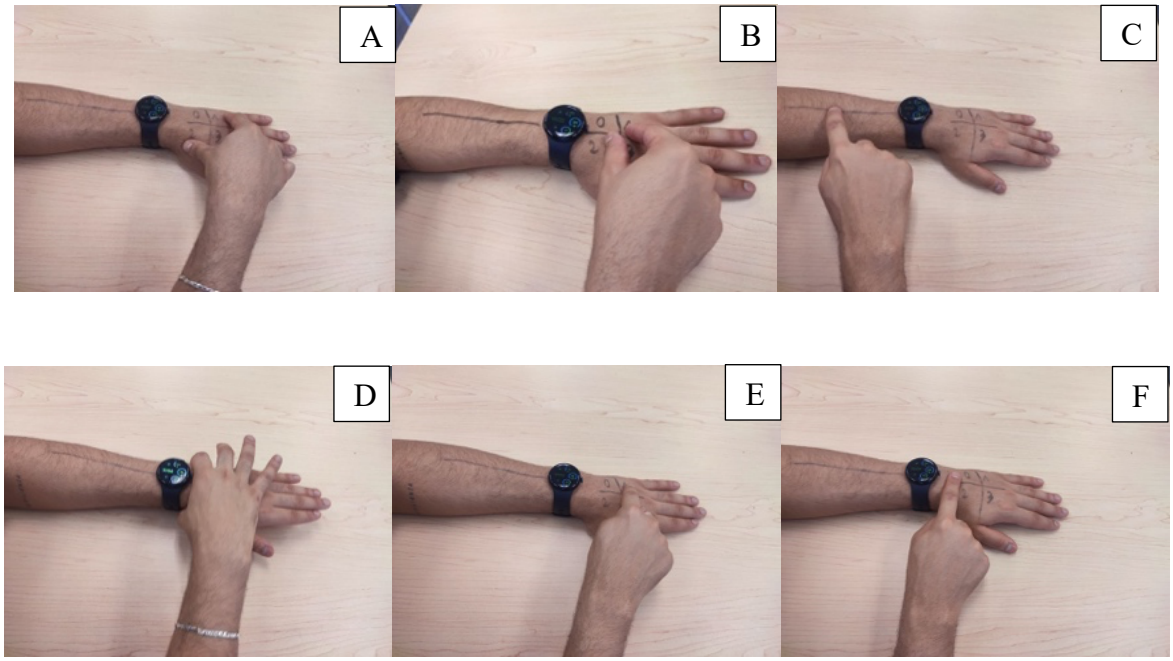

Figure 4 : Geste effectué pendant la collecte de données – Geste de zoom (A) – Geste de dézoom in (B) – Geste de glissement (C) – Geste de rotation (D) – Geste de saisie clavier numérique (E) – Geste d'appui long (F)

Le geste WakeUp, identifié sous le code Test-001-Freq, correspond à une élévation naturelle du bras visant à consulter la montre portée au poignet. Il consiste à quitter une position de repos, bras le long du corps, pour amener progressivement le bras vers une position de consultation, typique de l'action d'activation ou d'interaction avec une montre connectée. Ce geste reproduit un comportement spontané et fréquent dans les usages quotidiens des montres intelligentes,

notamment lorsque l'utilisateur active l'écran, vérifie l'heure, consulte une notification, ou interagit avec une application. Il pourrait être utilisé comme geste d'activation.

Le double tapotement (DT) consiste à effectuer deux tapotements rapides et consécutifs au-dessus de la main autour de la montre avec un ou plusieurs doigts, simulant une interaction courte et discrète. Il vise à tester la capacité du système à détecter des événements gestuels rapides et successifs.

Le geste de balayage (SW) est réalisé sous la forme d'un mouvement linéaire sur une distance de 5 à 10 centimètres, effectué sur la peau. Il peut être orienté horizontalement (de droite à gauche ou de gauche à droite) ou verticalement (du haut vers le bas ou du bas vers le haut). Ce geste permet d'évaluer la sensibilité directionnelle du système.

Le clavier numérique (NP) simule une interaction sur une zone virtuelle divisée en quatre touches et tracée mentalement sur le dessus de la main. Le participant effectue un geste ciblé vers l'une des quatre zones : haut gauche, haut droite, bas gauche ou bas droite. Ce type d'interaction permet d'évaluer la précision du système lorsqu'il s'agit de localiser une action dans une zone restreinte.

Le geste de rotation (RT) consiste à faire pivoter deux doigts sur un angle compris entre 90° et 180°, directement sur ou au-dessus de la peau. Il peut être effectué dans deux directions principales : vers le haut ou vers le bas. Ce mouvement vise à tester la reconnaissance de gestes circulaires et la précision des mouvements rotatifs.

Le glissement (SL) implique le déplacement continu d'un doigt sur une distance de 10 à 15 centimètres, soit de droite à gauche, soit de gauche à droite. Il permet d'évaluer la capacité du système à détecter des gestes prolongés et fluides.

L'appui long (LP) consiste à maintenir une pression prolongée sur une zone spécifique de la peau, sans mouvement. Ce geste est conçu pour tester la reconnaissance de contacts stationnaires de longue durée.

Enfin, le geste de zoom (ZM) est simulé par l'écartement ou le rapprochement de deux doigts, mimant une interaction de zoom avant ou de zoom arrière. Il permet d'évaluer la sensibilité du système à la variation simultanée de deux points de contact.

La liste complète des gestes étudiés, accompagnée de leur description détaillée (nom, consigne, durée, position du bras), est disponible dans le tableau ci-dessous.

Tableau 1 : Tableau de description détaillée des gestes à paramétrer dans l'application de collecte.

ID	Nom du Geste	Description	Durée (sec)	Directions/Actions	Objectif	Nombre Total de Tests
DT	Double Tapotement	Deux tapotements rapides et consécutifs au-dessus de la main.	3	N/A (DT)	Tester la reconnaissance d'interactions rapides et successives.	5
SW	Balayage (Swipe)	Mouvement horizontal ou vertical sur une distance de 5-10 cm au-dessus de la main.	5	SW-HL: Droite → Gauche, SW-HR: Gauche → Droite, SW-VT: Haut → Bas, SW-VB: Bas → Haut	Évaluer la sensibilité aux mouvements directionnels.	20 (4 directions x 5)
NP	Clavier Numérique	Simulation de saisie sur un clavier virtuel tracé sur le dessus de main qui simule 4 quatre zones.	5	NP-TL : Touche Haut Gauche, (0) NP-TR : Touche Haut Droite, (1) NP-DL : Touche Bas Gauche (2) NP-DR : Touche Bas Droite (3)	Tester la précision pour des interactions dans une zone du dessus de la main idéal pour en faire un clavier	20 (4 touches x 5)
RT	Rotation (Rotate)	Faire pivoter deux doigts sur un angle de 90-180° sur la peau.	5	RT-UP: Rotation vers le haut,	Tester la précision des mouvements de rotation.	10

				RT-DN : Rotation vers le bas		(2 directions x 5)
SL	Glissement (Slide)	Glisser un doigt sur 10-15 cm, horizontalement ou verticalement.	5	SL-HL : Droite → Gauche, SL-HR: Gauche → Droite,	Tester la détection des glissements continus.	10 (2 directions x 5)
LP	Appui Long (Long Press)	Maintenir un appui prolongé sur la peau.	5	N/A (LP)	Tester la reconnaissance des pressions prolongées.	5
ZM	Zoom In/Out	Doigts pour simuler un zoom.	5	ZM-IN: Zoom avant, ZM-OUT: Zoom arrière	Évaluer la précision des gestes de zoom.	10 (2 directions x 5)
Test-001-Freq	WakeUp	Quitter une position bras le long du corps pour le porter en position de consultation de la montre.	5	Mouvement naturel de relevé	Simuler un comportement spontané d'activation ou de consultation.	20

3.5 DONNÉES ET ANALYSES

L'approche méthodologique adoptée dans cette étude repose sur l'exploitation conjointe de plusieurs types de capteurs intégrés à la montre connectée, tels que l'accéléromètre, le gyroscope et le capteur de lumière, etc. (Tableau 2). Cette combinaison permet de générer une représentation fine, multidimensionnelle et temporelle des gestes réalisés par les participants. Les données ainsi recueillies serviront à entraîner des modèles d'apprentissage automatique, spécifiquement conçus pour reconnaître les gestes effectués à proximité de la montre. En fonction des performances obtenues, différents algorithmes pourront être explorés afin d'identifier celui offrant le meilleur équilibre entre précision, capacité de généralisation et robustesse.

Tableau 2 : Liste des capteurs intégrés à la Google Pixel 3 retenu pour l'expérimentation

Capteur	Vendeur	Version	Mode Recommandé	Description
Accelerometer-Uncalibrated	TDK	1	Continu	Fournit les accélérations brutes sur les axes X, Y, Z, avec une estimation des biais. Nécessite un traitement manuel.
ECG Sensor	TI	1	Continu	Collecte les signaux ECG (électrocardiogramme) pour analyser l'activité cardiaque en temps réel ou pour des études médicales.
Galvanic Skin Response	TI	1	Continu	Mesure la conductance de la peau pour analyser le stress ou les réponses émotionnelles.
Game Rotation Vector Sensor	Google	1	Continu	Fournit l'orientation en 3D sans dérive magnétique. Utilisé pour la VR/AR ou les jeux interactifs.
GazeSensor	Google	1	Continu	Suit la direction du regard pour des interactions utilisateur ou des études comportementales.
Geomagnetic Rotation Vector Sensor	Google	1	Continu	Fournit l'orientation basée sur les champs magnétiques. Idéal pour la navigation et les applications nécessitant une boussole.
Gravity Sensor	Google	1	Continu	Mesure la force gravitationnelle sur les axes X, Y, Z. Utile pour l'analyse posturale ou les gestes.
Gyroscope-Uncalibrated	TDK	1	Continu	Fournit des vitesses angulaires brutes sur les axes X, Y, Z, avec des biais non corrigés.
Instant Motion Sensor	TDK	1	Continu	Détecte instantanément les mouvements brusques ou soudains. Utile pour déclencher des événements en temps réel.
Linear Acceleration Sensor	Google	1	Continu	Fournit les accélérations sans gravité sur les axes X, Y, Z. Utile pour des analyses de mouvement net.
Low latency off body detect	Google	1	Continu	Détecte si un appareil est en contact avec la peau, optimisé pour économiser de l'énergie dans les wearables.
Magnetometer Sensor-Uncalibrated	STMicro	1	Continu	Mesure les champs magnétiques bruts sur les axes X, Y, Z, avec des biais non corrigés.
Orientation Sensor	Google	1	Continu	Fournit l'orientation en degrés (X, Y, Z). Utile pour les analyses

				simples de mouvement ou de position.
PPG Sensor	TI	1	Continu	Mesure les variations de volume sanguin à l'aide de la photopléthysmographie. Utilisé pour la fréquence cardiaque et le stress.
Pressure Sensor	Goermicro	1	Continu	Mesure la pression atmosphérique. Utilisé pour des applications environnementales ou des mesures d'altitude.
RaiseToTalk	Google	1	Continu	Détecte un mouvement spécifique pour activer un assistant vocal ou un microphone.
Rotation Vector Sensor	Google	1	Continu	Fournit l'orientation en 3D en combinant les données des autres capteurs. Idéal pour la VR/AR et les applications immersives.
Skin temperature sensor	TI	1	Continu	Mesure la température de la surface de la peau. Utilisé pour des applications de santé ou de suivi physiologique.
Stationary Sensor	TDK	1	Continu	Détecte l'absence de mouvement et peut déclencher des actions spécifiques.
Step Detector	Google	1	Continu	Détecte uniquement les événements de pas spécifiques. Réagit rapidement et économise de l'énergie.
TCS3701 light sensor	AMS	1	Continu	Mesure l'intensité et la couleur de la lumière ambiante pour ajuster les écrans ou collecter des données environnementales.
TiltToWake	Google	1	Continu	Détecte une inclinaison pour réveiller l'écran ou activer un appareil. Économise l'énergie.

L'interprétation des résultats se fera à la lumière des objectifs fixés et des scénarios d'usage envisagés. Si les performances des modèles s'avèrent insuffisantes, notamment en raison de variations interindividuelles ou de contextes particuliers, des ajustements méthodologiques seront envisagés. Ceux-ci pourront inclure une collecte complémentaire de données, la révision du protocole expérimental ou l'intégration de nouvelles variables explicatives. L'ensemble de la démarche s'inscrit

dans une logique d’amélioration continue, reposant sur l’analyse des erreurs, les retours d’expérimentation et la confrontation empirique des hypothèses.

CHAPITRE 4 : APPLICATION DE COLLECTE

Contrairement à d'autres études, nous avons choisi d'utiliser exclusivement les capteurs intégrés aux montres intelligentes pour la collecte des données. Cette approche présente plusieurs avantages déterminants. Elle permet une collecte continue et naturelle, tout en évaluant la capacité de la montre à fonctionner comme un dispositif autonome de reconnaissance gestuelle, sans recourir à du matériel et capteurs additionnels. À l'inverse, les méthodes s'appuyant sur des capteurs externes ou des systèmes d'annotation multimodale synchronisés (vidéo, audio, données capteurs) impliquent des contraintes significatives. Ces systèmes nécessitent une synchronisation temporelle précise entre la vidéo du geste, le signal audio et les données des capteurs, afin de permettre l'annotation manuelle. Un tel processus est fastidieux et chronophage. Il augmente aussi la complexité technique en exigeant un calibrage manuel rigoureux, ce qui peut potentiellement être une source d'erreurs humaines. En optant pour une application mobile personnalisée et modulaire, nous levons ces limitations tout en favorisant la reproductibilité de l'étude. En effet, notre protocole peut aisément être reproduit sur d'autres modèles de montres commerciales, ce qui renforce la généralisabilité des résultats.

Pour réaliser cette collecte, nous avons développé une application dédiée et compatible pour Android et Wear OS, conçu pour guider les participants, automatiser la collecte et l'étiquetage des données. Cette application intègre des retours et signaux haptiques, comme la vibration et des bips pouvant aider le participant à se retrouver pendant la phase de collecte. En plus, l'application simplifie et structure le processus, en agrégeant directement les données issues de plusieurs capteurs dans un fichier CSV. Elle remplace les systèmes d'étiquetage manuel et/ou par vidéo, réduisant ainsi le risque d'erreurs humaines et améliorant la rapidité et l'efficacité du flux de travail.

Dans ce chapitre nous verrons en détail le processus de développement de l'application de collecte de données. Ce dernier inclut des descriptions détaillées de l'architecture de l'application, de ses composants modulaires, des choix technologiques, des défis rencontrés, et des solutions

adoptées. Les explications seront étayées par des diagrammes, des exemples et des détails techniques.

4.1 VUE D'ENSEMBLE DE L'APPLICATION.

L'application de collecte développée dans le cadre de cette étude a pour objectif principal de recueillir les données issues des capteurs intégrés aux dispositifs intelligents. Déclinée en plusieurs versions selon le type d'appareil utilisé, elle repose sur une architecture multiplateforme compatible à la fois avec les montres intelligentes fonctionnant sous Wear OS et les téléphones intelligents sous Android. Chaque version de l'application intègre des fonctionnalités spécifiques, adaptées à son environnement matériel et à son rôle dans le processus de collecte.

4.1.1 FONCTIONNALITE DE L'APPLICATION DE LA MONTRE (WEAR OS)

La version Wear OS est principalement dédiée à la captation directe des signaux physiologiques et environnementaux à l'aide des capteurs intégrés à la montre intelligente. Elle fonctionne comme un point de collecte non autonome, dans la mesure où elle dépend des instructions envoyées par l'application companion installée sur le téléphone. Une fois les paramètres définis, cette version permet de lancer les mesures et de transmettre les données collectées à l'application Android. Elle est donc essentielle pour assurer une collecte fine, continue et localisée des signaux, au plus près de la peau et des mouvements de l'utilisateur.

4.1.2 FONCTIONNALITE DE L'APPLICATION TELEPHONE

La version Android pour téléphone intelligent, occupe une position centrale dans l'écosystème applicatif. Elle permet de superviser l'ensemble du processus expérimental en guidant le participant, en configurant les paramètres de mesure (tels que la fréquence d'échantillonnage, la durée ou les seuils de détection), et en assurant la coordination entre les différents dispositifs. Elle est également en mesure de capturer les signaux issus des capteurs internes du téléphone, d'envoyer des commandes vers la montre connectée, de sélectionner les capteurs actifs (sur la montre ou sur le téléphone) dont les données sont à collecter, ainsi que d'afficher les informations (descriptif et

graphique) issues des capteurs. Cette version assure également la sauvegarde des données, que ce soit en local sur l'appareil ou à distance via une infrastructure cloud, notamment en utilisant Firebase. Par ailleurs, cette version prend en charge des fonctionnalités avancées telles que la programmation des gestes à effectuer ou encore la définition de scénarios expérimentaux personnalisés avec le nombre de répétions de gestes et de scénarios.

Il convient de noter que l'application Android est capable de communiquer avec plusieurs montres WearOS simultanément, grâce à l'API MessageClient (section 4.2.2). Cette communication entre plusieurs appareils peut entraîner un léger décalage lors de la synchronisation du démarrage de la collecte, mais celui-ci reste négligeable, de l'ordre de 800 millisecondes pour les besoins de l'étude.

4.1.3 MODE DE FONCTIONNEMENT

Afin de rendre l'application adaptable à divers types de protocoles expérimentaux, deux modes de collecte de données ont été développés :

- Le mode libre (Figure 5) : ce mode permet de démarrer la collecte de manière non scénarisée. Une fois lancée, la collecte s'effectue en continu à partir des capteurs sélectionnés, et se poursuit jusqu'à ce que l'utilisateur l'interrompe manuellement en appuyant sur un bouton "Stop".

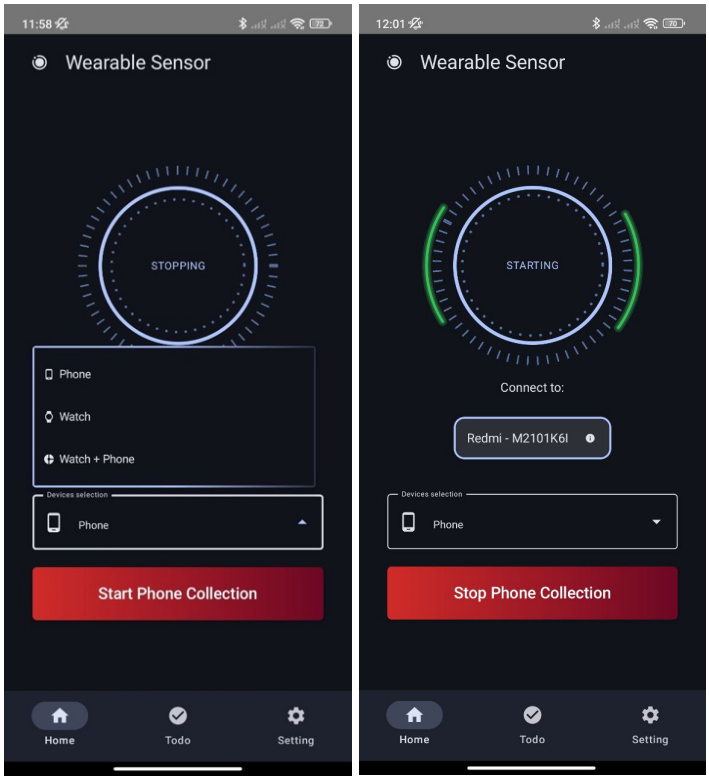

Figure 5 : Capture d'écran de l'interface du mode libre de l'application de collecte

- Le mode scénario (Figure 6) : Il est conçu pour des expérimentations plus structurées, reposant sur la répétition contrôlée de gestes définis à l'avance. Ce mode permet de créer des scénarios personnalisés, composés d'un ou plusieurs gestes prédéfinis. Un même geste peut être intégré plusieurs fois dans un scénario, selon un ordre fixe ou aléatoire, en fonction des objectifs expérimentaux (ex. : Figure 7 et Figure 9). Chaque scénario peut également être configuré pour être répété un nombre déterminé de fois, offrant ainsi une flexibilité dans la conception des sessions de collecte. Lors du démarrage, l'utilisateur a la possibilité de sélectionner et d'enchaîner plusieurs scénarios, ce qui permet de simuler des séquences complexes ou de comparer différents protocoles au sein d'une même session. Ce mode s'avère particulièrement pertinent pour garantir la reproductibilité des données et assurer une comparabilité inter-individuelle des résultats.

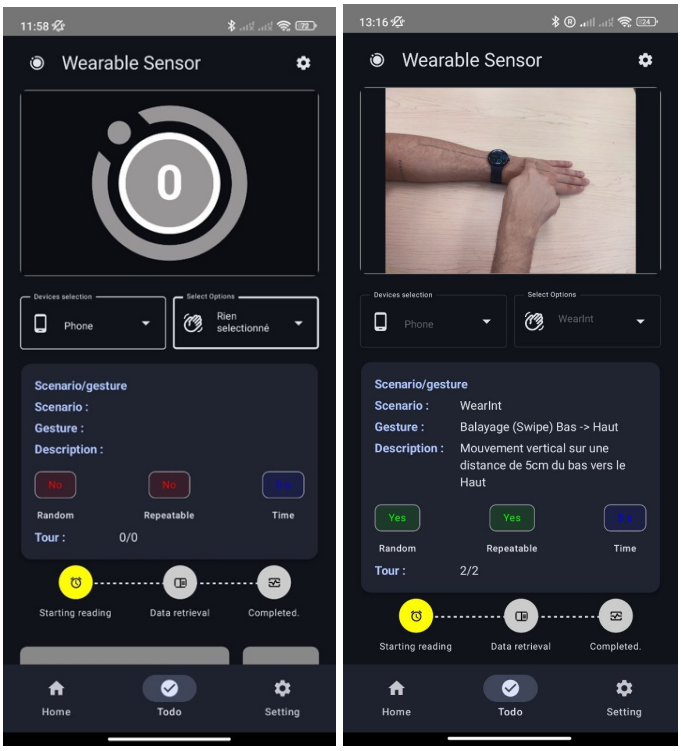

Figure 6 : Capture d'écran de l'interface du mode scénario de l'application de collecte

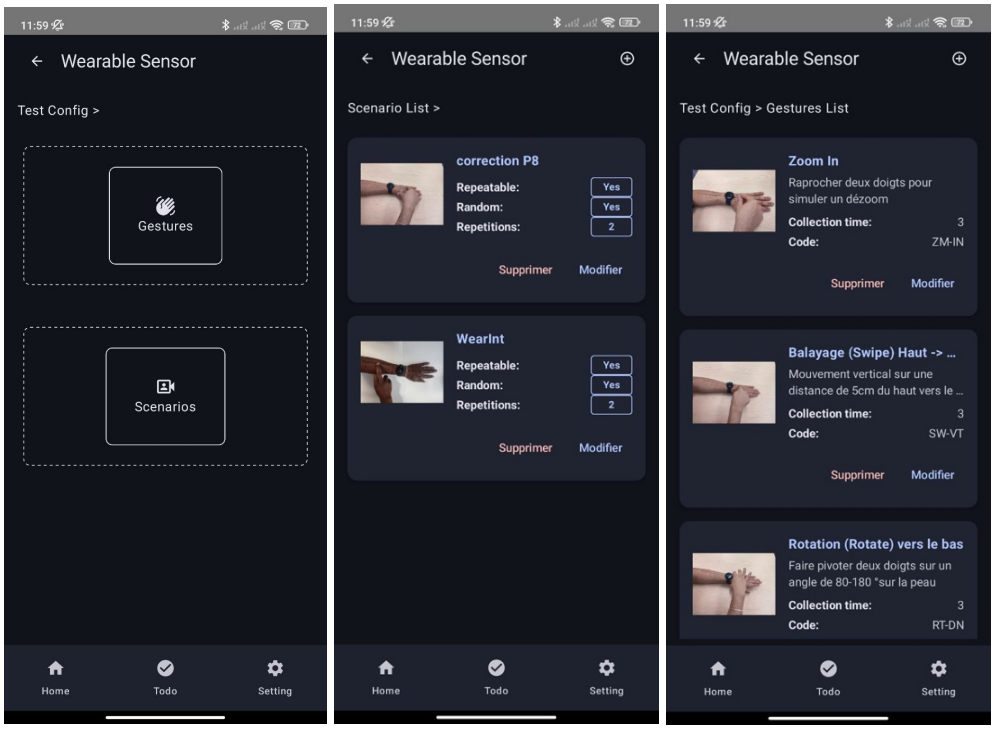

Figure 7 : Capture d'écran de l'interface de configuration des scénarios et des gestes

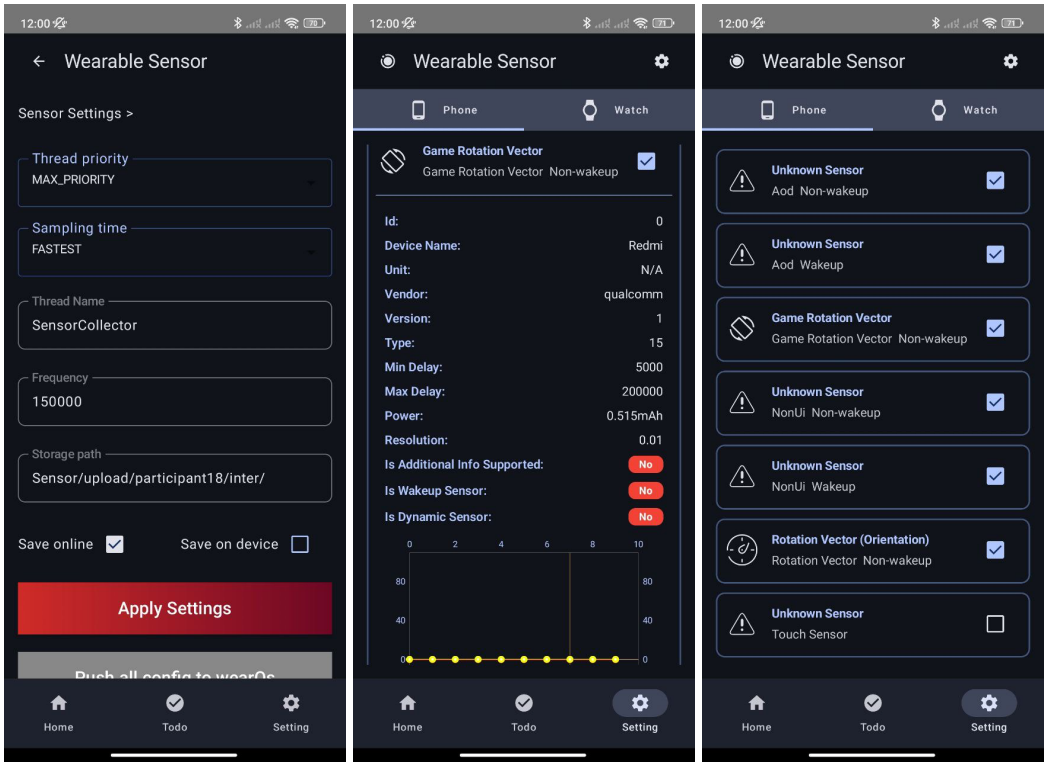

Figure 8 : Capture d'écran de l'interface de configuration de la fréquence d'échantillonnage et de la sélection des capteurs

4.2 CHOIX TECHNOLOGIQUE.

Les technologies utilisées dans le développement de l'application ont été sélectionnées avec soin pour répondre aux exigences du projet et garantir une performance optimale, en particulier dans l'utilisation des capteurs natifs.

Kotlin [24], le langage créé par JetBrains en 2011 et recommandé par Google pour Android depuis la conférence Google I/O 2019, a été choisi pour sa performance, sa lisibilité et sa compatibilité avec les dernières versions du système Android. Ce langage de programmation orienté objet et fonctionnel, avec un typage statique permet de compiler pour la machine virtuelle Java; Il offre un accès direct et natif aux API des capteurs, essentiel pour assurer des performances optimales et une faible latence lors de la collecte des données. Le développement natif permet une meilleure exploitation des capacités des capteurs, une précision accrue des données, et une gestion

optimisée de l'autonomie énergétique. De plus, il renforce la fiabilité en permettant une gestion fine des permissions et des interactions spécifiques avec le matériel. Les versions supportées et les bibliothèques utilisées sont dans le tableau comparatif suivant (Tableau 3).

TABLEAU 3 : Tableau récapitulatif des spécifications et dépendances par modules (Montre, Téléphone, et Module partagé)

Module	Version de compilation	Version cible	Version SDK minimale supportée	Kotlin J V M Target	Bibliothèques
Montre	34 (Android 14)	34 (Android 14)	28 (Android 9 Pie)	1.8	Jetpack compose
Téléphone	35 (Android 14 Preview)	34 (Android 14)	23 (Android 6.0 Marshmallow)	1.8	Firebase, MPAndroidChart, et Accompanist Pager
Shared (Module partagé)	25 (Android 7.1 Nougat)	N/A	23 (Android 6.0 Marshmallow)	1.8	Gson, Coroutine, WorkManager

Pour ce qui est de la création d'interfaces, *Jetpack Compose* [25], un outil moderne (cadriciel d'interface utilisateur) conçu pour la création d'interfaces utilisateur (UI), a été utilisé. Annoncé en 2019 et introduit par Google en 2021, Jetpack Compose offre plusieurs avantages par rapport à l'approche traditionnelle basée sur XML. Cette technologie se distingue par sa concision et sa lisibilité, nécessitant moins de code tout en permettant de créer des interfaces réactives et adaptatives. Grâce à son paradigme déclaratif, elle facilite la création d'interfaces dynamiques. De plus, l'intégration de l'interface et de la logique dans un même fichier Kotlin simplifie la gestion et réduit le couplage entre la vue et le code métier.

Firebase a été intégré à notre projet en raison de sa capacité à fournir un stockage en temps réel, sécurisé et centralisé, parfaitement adapté aux besoins du projet. Le service de stockage de Firebase a été utilisé pour héberger les fichiers CSV générés lors de la collecte des données, garantissant leur accessibilité et leur sécurité. Parallèlement, le service de base de données en temps réel a permis d'enregistrer efficacement les informations issues des capteurs, des gestes et des scénarios dans des collections structurées. Grâce à sa flexibilité et sa synchronisation optimisée, Firebase s'est imposé comme une solution fiable et adaptée à la gestion des données dynamiques et aux exigences de performance sur le marché. Notons aussi que, grâce au SDK et aux bibliothèques clientes présentes sur Kotlin, il s'intègre très facilement à notre projet.

Android Studio, quant à lui, est l'IDE utilisé pour faire développer notre application, car il supporte nativement le développement d'application pour Android et est maintenu directement par Google, garantissant une compatibilité et des mises à jour régulières. Il est livré avec l'Android SDK, facilitant la configuration et l'utilisation des dernières fonctionnalités d'Android. En plus, il inclut un émulateur performant qui permet de tester les applications sur différents appareils, versions d'Android, tailles d'écran et résolutions. Grâce à certaines fonctionnalités, comme la refactorisation, autocomplétion intelligente, le debugging et analyse avancée, puis la prévisualisation. Le développement de notre application est beaucoup plus simple. La version utilisée est Ladybug 2024.2.2.

4.3 CONCEPTION DE L'APPLICATION.

L'application s'appuie sur l'architecture MVVM (Model-View-ViewModel), qui permet de bien séparer les responsabilités entre la couche de présentation, la logique métier et la gestion des données. Ce choix structurel renforce la maintenabilité, facilite l'évolutivité du projet, et améliore l'organisation du code en limitant les interdépendances. En réduisant les risques d'erreurs et en favorisant le test unitaire, cette architecture offre une base solide, parfaitement adaptée aux exigences techniques du projet et à ses évolutions futures.

Par ailleurs, des choix stratégiques ont été effectués en matière d'organisation logicielle, avec l'adoption d'une architecture modulaire pensée pour isoler clairement les responsabilités fonctionnelles. Cette approche vise à simplifier la maintenance, faciliter les tests et accélérer les mises à jour de l'application. Le projet est structuré autour de trois modules principaux : app, wear et shared. Cette séparation modulaire améliore la lisibilité du code, optimise la collaboration entre développeurs, et permet une scalabilité efficace, notamment dans un contexte de développement multiplateforme ou d'évolutions futures de l'application.

4.3.1 BIBLIOTHÈQUE APP

La bibliothèque **app** représente le noyau de l'application mobile et joue un rôle central dans la gestion des expérimentations. Elle est conçue pour guider les participants à travers les différentes étapes de la collecte de données tout en assurant une communication fluide avec les autres composants. Cette bibliothèque gère plusieurs aspects clés.

Premièrement, elle fournit une interface utilisateur intuitive et bien structurée, conçue pour afficher des consignes claires et compréhensibles. Elle permet plusieurs approches pour collecter les données en fonction des besoins spécifiques de l'expérimentation. La première méthode permet une collecte illimitée des données depuis plusieurs montres simultanément grâce à un concept de nœud, ou depuis le téléphone exécutant l'application, ou encore depuis les deux dispositifs. Cette flexibilité est rendue possible grâce à un champ de sélection permettant de définir le dispositif utilisé pour la collecte, ainsi qu'un bouton de démarrage et d'arrêt, simplifiant l'utilisation.

La deuxième méthode repose sur le concept de scénarios de gestes. Dans cette approche, l'expérimentateur commence par enregistrer plusieurs gestes dans l'application, sans se soucier de leur ordre initial. Chaque geste contient des informations essentielles, telles que le titre du geste, une description textuelle, un temps de lecture destiné au participant pour comprendre les consignes, une durée de collecte en secondes, un code d'étiquetage des données collectées, et une photo illustrant visuellement le geste attendu. Une fois les gestes définis, l'expérimentateur configure un scénario, qui est une compilation de plusieurs gestes à exécuter pendant la phase d'expérimentation.

Les scénarios peuvent être configurés pour imposer un ordre strict ou pour permettre une exécution aléatoire. Ils peuvent également préciser le nombre de répétitions pour chaque geste si nécessaire. Un même geste peut être intégré plusieurs fois dans un scénario, selon les besoins spécifiques de l'expérience (Figure9). Pour simplifier la création et l'ajustement des scénarios, un système de glisser-déposer (drag and drop) a été intégré à l'interface, permettant de réorganiser facilement les gestes dans l'ordre souhaité. De plus, il est possible de sélectionner plusieurs scénarios à exécuter consécutivement pendant une même session de collecte, ce qui offre une grande flexibilité.

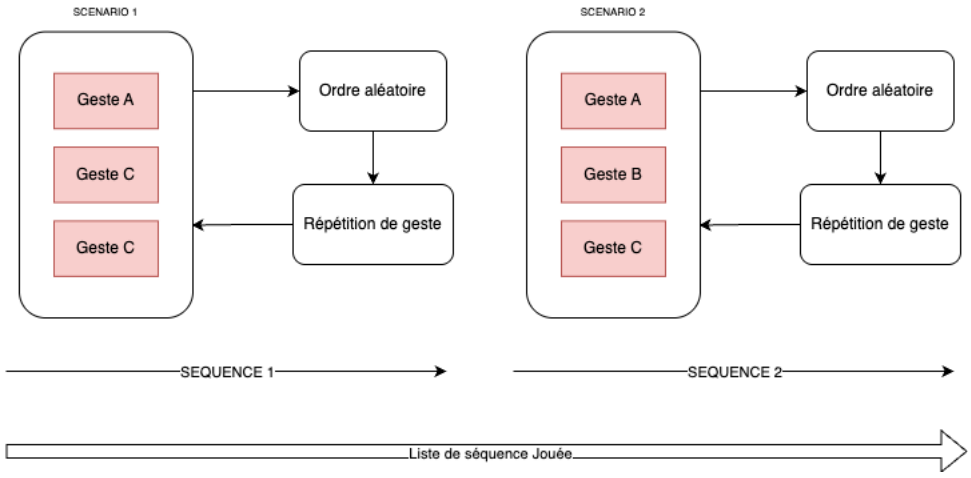

Figure 9 : Schéma explicatif du concept de scénario et de geste dans la programmation des interactions

Les participants reçoivent des instructions précises sur les gestes à réaliser, accompagnés de descriptions textuelles et visuelles. Cette approche garantit une meilleure compréhension des consignes, facilite l'exécution correcte des mouvements et réduit significativement les erreurs d'interprétation. De plus, l'application mobile permet de visualiser graphiquement les données des capteurs du téléphone, les variables disponibles, les informations de capteurs ainsi que de choisir les capteurs à utiliser pour chaque appareil impliqué dans la collecte. La liste des capteurs disponibles est obtenue automatiquement dès que l'appareil est connecté, permettant de ne collecter

que les données nécessaires à l'expérience. Cette fonctionnalité optimise l'utilisation des ressources et simplifie la configuration des sessions expérimentales.

Ensuite, la gestion des sessions constitue une autre fonctionnalité essentielle de cette bibliothèque. Elle permet de suivre l'avancement des expériences et d'enregistrer les résultats de manière structurée en local et/ou sur Firebase. Les données collectées pour chaque participant sont organisées de façon à permettre une analyse simple et rapide ultérieure. Cela permet de centraliser les informations tout en assurant leur intégrité et leur disponibilité d'autant plus que les répertoires d'enregistrement des données peuvent être définis.

Enfin, cette bibliothèque offre la possibilité de configurer les paramètres de fonctionnement des capteurs, tant sur la montre que sur le téléphone. Elle permet également d'ajuster des paramètres essentiels tels que la fréquence de collecte des données. Cette flexibilité garantit une adaptation optimale aux besoins spécifiques de chaque expérimentation. Par exemple, pour une phase nécessitant une grande précision temporelle, il est possible d'augmenter la fréquence de capture ; tandis que, pour des expériences de longue durée, la fréquence peut être réduite afin de préserver l'autonomie des dispositifs. Il est aussi réglé d'autres caractéristiques essentielles, telles que la sensibilité des capteurs, la plage de détection, ou encore les modes de fonctionnement spécifiques à certains capteurs (par exemple, un mode haute précision ou un mode écoénergie). Grâce à cette flexibilité, l'application offre un contrôle total sur le fonctionnement des capteurs, facilitant la réalisation d'expériences diversifiées, qu'il s'agisse de tests courts et intensifs ou d'expérimentations prolongées nécessitant une gestion stricte des ressources. Cette fonctionnalité contribue également à améliorer la fiabilité et la qualité des données collectées en assurant une configuration adaptée à chaque contexte.

4.3.2 BIBLIOTHÈQUE WEAR

La bibliothèque *wear*, complémentaire à la bibliothèque A*PP*, est conçue pour fonctionner sur les montres intelligentes grâce à un déclenchement initié depuis l'application mobile. Elle assure la collecte des signaux physiologiques associés aux gestes des participants en utilisant les capteurs

intégrés de la montre, selon les configurations établies via la partie mobile. En outre, la bibliothèque Watch gère la transmission des données vers l'application mobile. En cas de perte de connexion, les données collectées sont temporairement stockées localement sur la montre et synchronisées dès que la connexion est rétablie, minimisant ainsi les risques de perte de données. La bibliothèque permet également d'afficher l'étape en cours sur l'écran de la montre, une fonctionnalité étroitement intégrée à l'application mobile. La montre dépend du smartphone pour recevoir les consignes et synchroniser les données, garantissant une coordination optimale entre les deux dispositifs. Cela offre une expérience fluide aux participants tout en permettant à l'expérimentateur de conserver un contrôle centralisé du processus, renforçant la cohérence des données collectées et facilitant le suivi en temps réel des gestes effectués.

4.3.3 BIBLIOTHÈQUE SHARED

La bibliothèque **Shared** est conçue pour être utilisée de manière centralisée, soit par héritage, soit par appel direct, tant par l'application mobile que par la montre connectée. Elle a pour but de standardiser et d'optimiser la gestion des capteurs, des services, ainsi que la communication entre les dispositifs. C'est aussi elle qui assure une uniformité dans le traitement des données et prend en charge l'initialisation et la configuration des capteurs selon les besoins spécifiques de l'expérimentation, garantissant une collecte précise et fiable. Les données brutes capturées sont ensuite formatées de manière cohérente pour faciliter leur analyse et leur étiquetage.

La bibliothèque garantit également la standardisation des données, quel que soit l'appareil utilisé (montre ou mobile), afin d'assurer la cohérence des résultats et une intégration fluide dans les modèles d'apprentissage automatique.

En somme, chaque bibliothèque du projet a une fonction spécifique, ce qui contribue à clarifier la structure de notre application. Grâce à ce projet, nous sommes désormais en mesure de compiler à la fois du code pour la montre et du code pour le téléphone. En cas de problème sur l'un de ces appareils, nous pouvons identifier la bibliothèque à cibler pour le résoudre, ce qui facilite grandement la résolution des problèmes futurs et permet une amélioration du projet.

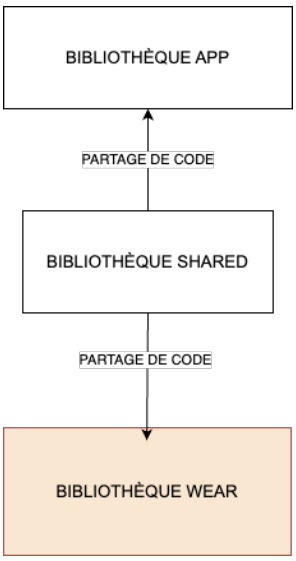

Figure 10 : Flux de communication entre les différentes bibliothèques de notre application de collecte

4.4 PATRON DE CONCEPTION ARCHITECTURALE (DESIGN PATTERN) ET DESCRIPTIONS DES PRINCIPAUX PACKAGES

La structuration de notre projet en plusieurs bibliothèques n'est pas la seule décision majeure prise pour en optimiser l'organisation. L'organisation architecturale globale, tout aussi importante, nous a amenés à adopter l'architecture *MVVM (Model-View-ViewModel)* pour organiser les composants de l'application (Figure 11). Ce choix nous a permis de simplifier les tests unitaires, faciliter la réutilisabilité du code, garantir une navigation intuitive et garantir la maintenabilité et l'évolutivité du projet.

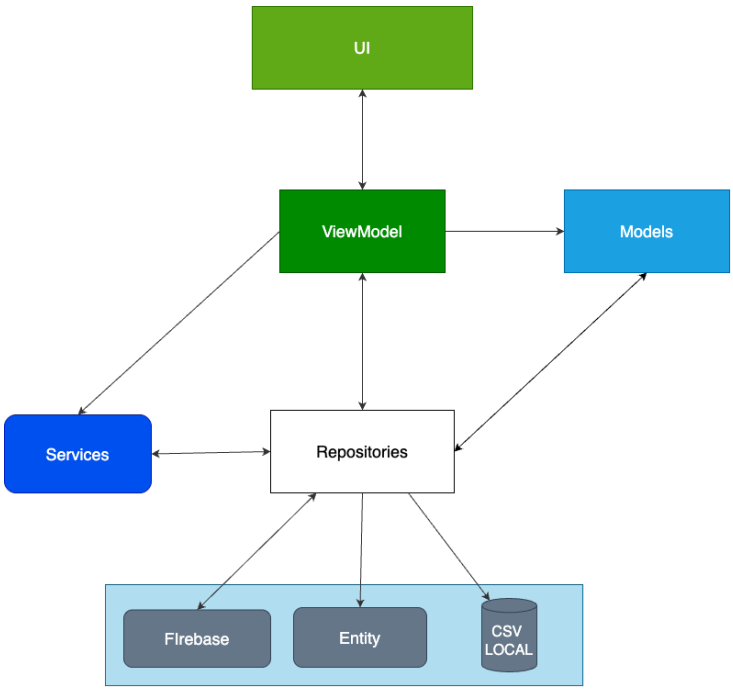

Figure 11 : Architecture de l'application - Modèle MVVM

L'organisation en fichier de notre projet est faite sous forme de package regroupant les classes de fichier de notre application par fonctions.

Le package *Activities* correspond à une unité d'écran ou à une interaction utilisateur spécifique, servant de point d'entrée pour une action précise. Il se limite à gérer la logique de navigation et à initialiser les ViewModels correspondants, garantissant ainsi une séparation nette entre la logique de présentation et la logique métier.

Présent uniquement dans les bibliothèques dédiées à la montre connectée et au téléphone, le package *UI (interface utilisateur)* contient les éléments visuels de l'application, comme les boutons, les étiquettes et les animations. Il constitue la couche directement exposée aux utilisateurs, leur permettant d'interagir avec l'application.

Le package *Entities* regroupe les principales structures de données utilisées dans l'application, telles que les modèles représentant les mouvements (ex. : ScenarioEntity, GestureEntity). Ces

entités facilitent l'organisation, la manipulation et le transfert des données dans les différentes couches de l'application.

Le package *Models* constitue le cœur de la logique métier. Il englobe les directives et algorithmes nécessaires à la gestion et au traitement des données. Par exemple, il définit les correspondances entre les codes de gestes et les actions associées, garantissant ainsi une exécution cohérente de la logique métier.

Repository est le package qui contient les fichiers qui constituent un point central d'accès aux données, qu'elles soient stockées localement (dans l'espace de stockage interne ou dans une base de données) ou hébergées à distance (comme Firebase). Il simplifie et centralise les opérations de récupération et de mise à jour des données, facilitant l'accès aux autres composants.

Le *Service* gère les capteurs et des données brutes recueillies par la montre intelligente ou par le téléphone. Il permet la collecte de données provenant de capteurs, ainsi que de la transformation des données brutes en valeurs exploitables, prêtes à être utilisées par d'autres modules.

Le Package *BroadcastReceiver* contient toutes les classes qui écoutent et réagissent aux événements systèmes ou applicatifs. Il gère les tâches liées aux événements spécifiques, comme les changements de connectivité, les alertes systèmes ou tout autre stimulus externe, garantissant une gestion fiable des signaux externes.

Le *ViewModel* joue le rôle d'intermédiaire entre la logique métier (Models, Repository) et l'interface utilisateur (UI). Il contient les données nécessaires à l'affichage et observe les changements afin de mettre à jour l'interface de manière dynamique.

Enfin, le module *Utils* regroupe des fonctions et des classes utilitaires réutilisables, telles que la gestion du formatage des données ou les conversions, comme la transformation des données des capteurs en valeurs lisibles.

En résumé, l'architecture modulaire de l'application, fondée sur le modèle MVVM, garantit une gestion claire des responsabilités, facilitant l'évolutivité et la réutilisation du code. Chaque module,

qu'il s'agisse de l'interface utilisateur, de la gestion des capteurs ou de la logique métier, contribue à une expérience fluide et cohérente et à la maintenabilité facile de notre projet d'application. La structure permet ainsi une maintenance simplifiée et une interaction optimale entre les différentes parties de l'application. Ce cadre préparera l'application à un fonctionnement efficace, en amenant naturellement l'analyse du flux de fonctionnement dans la section suivante.

4.5 API DE COMMUNICATION DU GOOGLE PLAY SERVICES WEARABLE

Google fournit dans sa documentation des API de couche de données Wear OS, composées de plusieurs types de clients adaptés à divers types de données et contextes d'utilisation [26]. Ces clients facilitent la communication entre la montre connectée et le téléphone, en répondant aux différentes situations et conditions d'exploitation. C'est bien sur ces dernières que nous nous sommes basés pendant la phase de conception de notre application. Ces clients sont : le *DataClient*, le *MessageClient* et le *ChannelClient*.

4.5.1 DATACLIENT

Le client *DataClient* permet de lire et d'écrire des *DataItems* ainsi que des Assets. Les *DataItems* sont des unités d'information synchronisées automatiquement sur tous les appareils associés appartenant au même utilisateur. Ils sont stockés de manière persistante, garantissant un accès continu jusqu'à leur suppression explicite.

Les Assets, quant à eux, sont spécialement conçus pour gérer des données volumineuses, telles que des images ou des fichiers multimédias. Ils complètent les *DataItems* en offrant une solution efficace pour le stockage et le transfert de grandes quantités de données, sans risque de surcharge.

Cependant, l'utilisation de *DataClient* comporte certaines limitations. La synchronisation dépend de la connectivité réseau, ce qui peut entraîner des retards en cas de déconnexion des appareils. De plus, une utilisation intensive des Assets peut impacter la consommation d'énergie et

les performances réseau, notamment sur les appareils portables. Enfin, une gestion manuelle des *DataItems* est nécessaire pour éviter l'accumulation de données inutiles.

Malgré ces inconvénients, *DataClient* reste un outil puissant pour les applications nécessitant une synchronisation fiable, ainsi qu'une gestion avancée et un partage efficace des données entre appareils.

4.5.2 MESSAGECLIENT

Adaptés aux procédures à distance (RPC), les messages sont particulièrement efficaces pour des requêtes unidirectionnelles ou un modèle de communication de type requête-réponse. Contrairement à la synchronisation de données persistantes, les clients de messagerie nécessitent que les nœuds soient connectés au réseau au moment de l'envoi des messages.

Bien que ce client permette une livraison rapide vers le nœud distant, il présente certaines limitations. Notamment, il ne dispose pas d'un mécanisme intégré de nouvelle tentative en cas d'échec de transmission, et il ne prend pas en charge l'envoi de données de plus de 100 Ko.

Un point important souligné dans la documentation est la nécessité de limiter l'envoi de messages aux appareils proches, afin de préserver l'autonomie de la batterie. Cette précaution est particulièrement importante dans des contextes où les connexions réseau peuvent être instables ou lorsque les appareils fonctionnent sur une batterie limitée.

4.5.3 CHANNELCLIENT

ChannelClient permet une communication bidirectionnelle orientée flux entre deux appareils, offrant un tuyau de transmission idéal pour des cas spécifiques. Il est particulièrement utile pour transférer des fichiers lorsque l'accès à Internet est indisponible, envoyer des fichiers volumineux qui dépassent les limites de *MessageClient*, ou transmettre des données en continu, comme des flux audios.

Contrairement à *DataClient, ChannelClient* ne stocke pas les données localement avant la transmission, ce qui économise de l'espace disque. De plus, il transmet les données sous forme d'un flux continu d'octets plutôt qu'en unités distinctes.

Cependant, *ChannelClient* ne gère pas automatiquement la synchronisation ou la cohérence des données. Nous sommes donc nous même responsables de la gestion des données tout au long du transfert.

Pour notre travail, nous avons choisi d'utiliser *DataClient* avec des Assets pour gérer les données collectées. Cette décision repose principalement sur deux raisons. Tout d'abord, le volume important des données collectées nécessitait une solution capable de gérer des charges utiles volumineuses, ce que les Assets permettent de manière efficace. Ensuite, l'intégrité et la fiabilité des données étaient des critères essentiels, et *DataClient* garantit une synchronisation persistante des données entre appareils, même en cas de déconnexion temporaire. Cette approche assure une gestion de livraison des informations collectées, répondant aux exigences de fiabilité du projet.

Par ailleurs, les commandes de démarrage, d'arrêt et d'échange d'informations ont été confiées à *MessageClient*. Cette API est idéale pour les communications légères et rapides, où la transmission en temps réel est primordiale. Contrairement à *DataClient*, *MessageClient* permet d'envoyer des messages simples et non persistants avec une faible latence, ce qui en fait le choix parfait pour transmettre des commandes nécessitant une réception immédiate. Ainsi, l'utilisation combinée de *DataClient* et *MessageClient* répond efficacement aux besoins de gestion des données et de transmission des commandes dans notre projet.

4.6 FONCTIONNEMENT DE L'APPLICATION DE COLLECTE.

Le flux de fonctionnement de l'application a été conçu pour assurer une collecte de données à la fois efficace, fiable et adaptable à divers scénarios d'utilisation. Il repose sur une communication directe Bluetooth pour l'envoi par nœud et sur une possibilité de synchronisation cloud des données avec le client de données (*DataClient*) entre les appareils, éliminant ainsi une forte dépendance au cloud pour la synchronisation des données entre la montre et le téléphone. Ce choix, motivé par des

considérations liées à la gestion des appareils connectés ou déconnectés, ainsi qu'aux exigences de sécurité et de flexibilité, permet de prendre en charge simultanément plusieurs montres connectées. Il offre également la possibilité de configurer des options spécifiques, comme l'utilisation exclusive de la montre, du téléphone ou d'une combinaison des deux dispositifs.

Au démarrage de l'application mobile, celle-ci identifie et enregistre automatiquement l'ensemble des capteurs de l'appareil mobile, tout en collectant des informations telles que la marque, le modèle et un identifiant unique de l'appareil. Ces données sont stockées dans une base de données *Firestore* pour une consultation ultérieure. Si une montre est connectée, l'application transmet une requête pour obtenir la liste des capteurs disponibles sur la montre. La montre répond en transmettant ses informations, ainsi que des données d'identification telles que le modèle et la version du système. Ces informations sont ensuite enregistrées dans *Firestore* pour une utilisation ultérieure.

Le processus de préparation débute par la personnalisation des équipements et des réglages de collecte. À partir de l'application mobile, l'expérimentateur a la possibilité de sélectionner les capteurs à activer, de régler leur fréquence d'échantillonnage et de décider quels appareils seront utilisés pendant la séance. Les options de choix d'appareil offertes sont l'utilisation unique de la montre, du téléphone ou une collecte mixte (montre et téléphone). L'application mobile agit comme le point de contrôle principal, envoyant directement des commandes à la montre pour démarrer, arrêter ou ajuster les services de collecte des données. Cette approche garantit une synchronisation précise et immédiate entre les deux dispositifs, sans intervention d'un serveur externe ou d'un réseau distant.

Pendant la phase d'expérimentation, les consignes concernant les gestes à réaliser sont affichées sur le téléphone, tandis que seules les informations relatives au service de collecte et à l'état de la connexion avec le téléphone apparaissent sur l'écran de la montre (Figure 6).

Pour démarrer la phase de collecte, un message contenant les paramètres de configuration des capteurs et une liste de capteurs à sélectionner, tous fractionnés en petits paquets, sont envoyés

au nœud via un chemin associé au démarrage du service de collecte et à l'utilisation de cette configuration, de même que pour la mise en pause et l'arrêt. Cette segmentation permet de transmettre progressivement les données au téléphone par Bluetooth, en évitant toute surcharge du réseau et en assurant une communication fluide, même pour de lourds volumes de données. Le fractionnement intervient uniquement si la taille totale des données dépasse un seuil défini, et chaque fraction ne dépasse pas ce seuil, qui est de 100 Ko dans notre cas.

Au démarrage du service, les capteurs sélectionnés sur la montre commencent à capturer les données en temps réel. Un bip suivi d'une vibration se déclenche pour avertir les participants que la collecte a commencé. Étant donné le volume potentiellement important de données brutes générées par l'utilisation simultanée de plusieurs capteurs, un système robuste et asynchrone a été mis en place afin de gérer efficacement la mémoire, garantir la persistance des données, et éviter toute perte, même en conditions extrêmes.

Les données collectées sont d'abord stockées dans une zone tampon en mémoire, représentée par une file d'attente principale (writeQueue) pouvant contenir jusqu'à 50 000 éléments. Chaque donnée est préalablement filtrée pour s'assurer qu'elle est valide (absence de clés vides ou de structures incorrectes). Si cette file est temporairement fermée (notamment lors d'un flush bloquant), les données sont redirigées vers un tampon secondaire en mémoire (newBuffer). En cas de saturation de la file, un mécanisme de secours s'enclenche automatiquement : chaque élément excédentaire est immédiatement sauvegardé ligne par ligne dans un fichier temporaire au format JSON (temp_backup_data.json). Cette approche multi-niveaux garantit la continuité de la collecte, même lorsque la capacité de traitement est dépassée.

Un processus asynchrone est lancé en tâche de fond : une coroutine dédiée vérifie toutes les 1000 millisecondes l'état de la file d'attente. Lorsqu'elle contient des données, un flush est déclenché : les éléments sont extraits par lot (jusqu'à 500), puis convertis en *JSON* et compressés via *GZIP*. Le résultat est ensuite sauvegardé dans un fichier compressé unique (nommé temp_sensor_data_#.gz) situé dans un répertoire temporaire sécurisé. Cette méthode réduit significativement l'usage du stockage et simplifie les opérations futures (transfert, suppression, archivage).

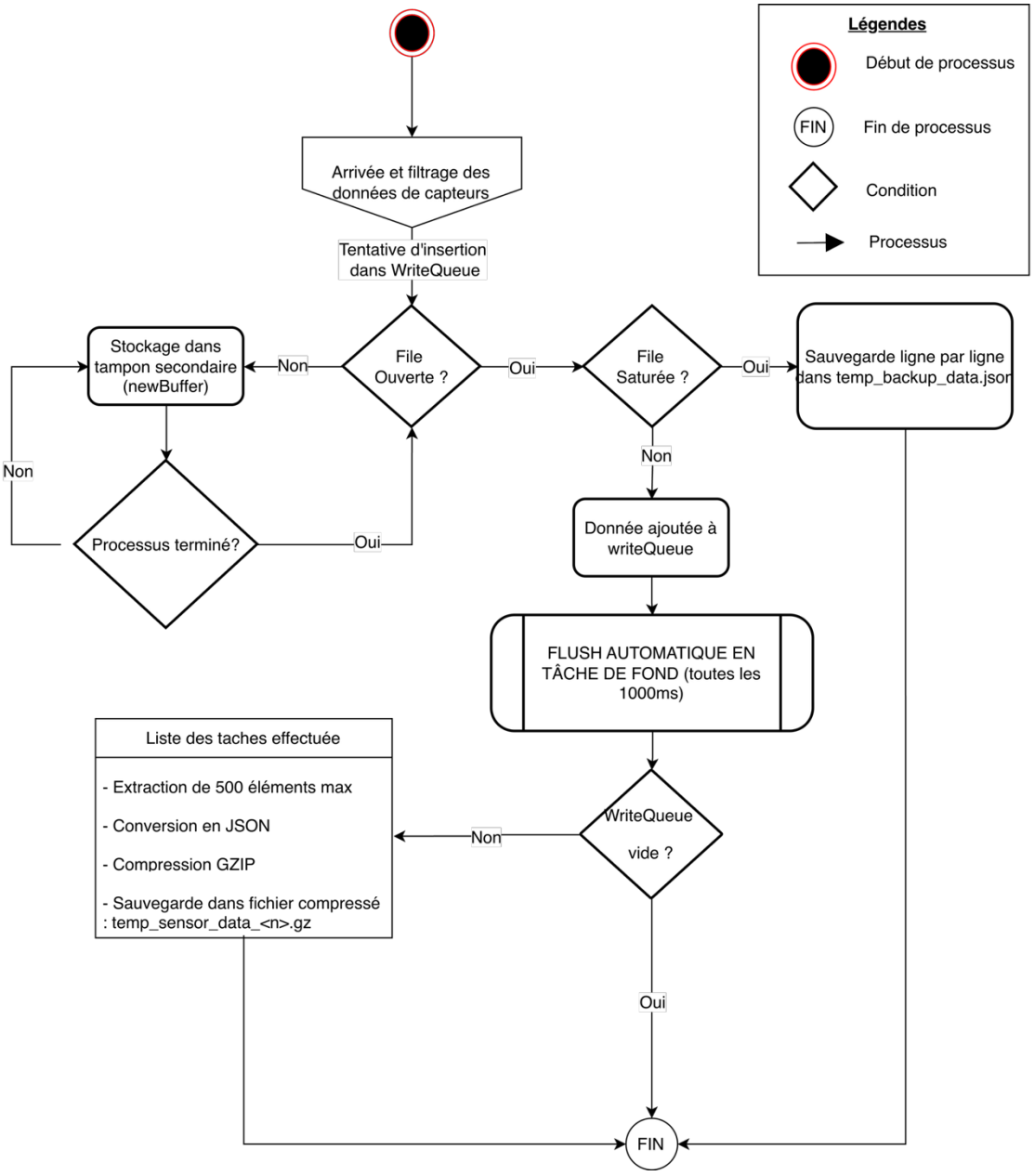

Figure 12 : Processus d'enregistrement des données issues des capteurs

Lorsqu'une récupération complète des données est nécessaire, que ce soit pour un envoi réseau, une synchronisation ou l'arrêt du service, une méthode spécifique est appelée (*getDataToSend*). Elle commence par verrouiller les accès concurrents grâce à un *ReentrantLock*, puis ferme temporairement la file d'attente pour empêcher l'ajout de nouvelles données pendant l'opération. Le contenu du tampon secondaire est vidé dans la file principale, et un flush bloquant est

réalisé pour garantir que toutes les données restantes sont bien enregistrées sur le stockage interne de la montre. Ensuite, tous les fichiers compressés présents dans le cache sont lus, décompressés, et analysés pour reconstruire les objets d'origine *(type Map<String, Any>)*.

Les lignes du fichier temporaire *JSON* sont également relues ligne par ligne, même en cas de corruption partielle, afin d'extraire un maximum de données valides. Les fichiers illisibles ou partiellement défectueux sont renommés avec un préfixe *corrupted_* pour analyse future, sans interrompre le processus.

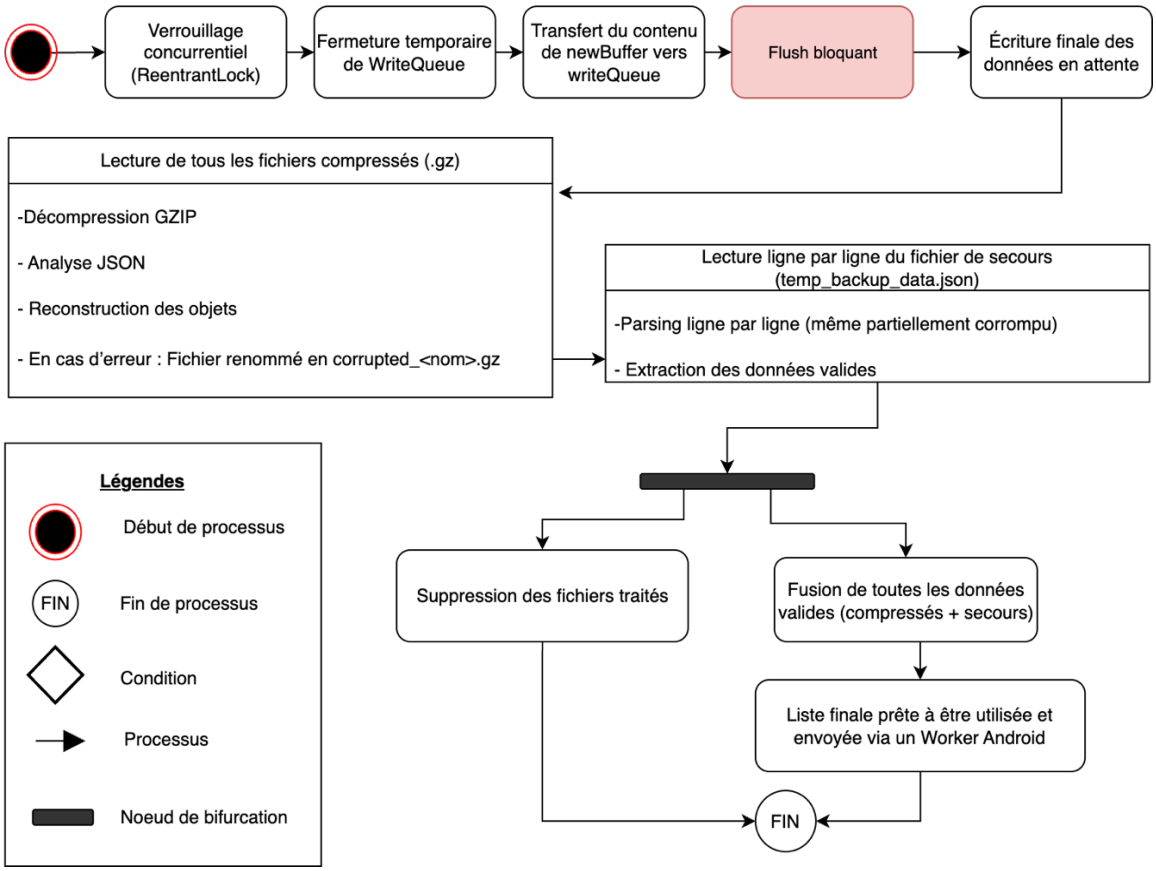

Figure 13 : Processus de récupération des données issues des capteurs

Une fois l'ensemble des données fusionnées, la liste résultante est retournée à l'application, prête à être exploitée. Ce traitement garantit que la collecte reste fiable même en cas d'interruption du service. Un fil de travail Android est utilisé pour effectuer cette récupération de manière sûre et

persistante, assurant que toutes les données sont envoyées à l'activité parente avant la terminaison effective du service, sans bloquer l'arrêt du processus. Enfin, un bip de fin signale au participant que la session de collecte est terminée.

Pendant le démarrage des services, il existe plusieurs manières de gérer les fils d'exécution de gestion des capteurs (Figure 14). On peut soit attribuer un fil d'exécution dédié à chaque capteur individuel, soit utiliser un seul fils d'exécution pour tous les capteurs, ou encore adopter une approche intermédiaire. La première méthode, bien que permettant une isolation complète des capteurs, présente l'inconvénient d'une consommation élevée en ressources système, notamment en termes de mémoire et de puissance CPU, ce qui peut ralentir significativement le démarrage des services, surtout sur des appareils à ressources limitées, comme les montres Wear OS. La seconde méthode, en regroupant tous les capteurs sur un unique fil d'exécution, a pour avantage de réduire la surcharge liée à la création et à la gestion de multiple fils d'exécution, ce qui améliore les performances et accélère le démarrage des services. Cependant, cette approche peut entraîner des goulots d'étranglement si de nombreux capteurs génèrent des événements simultanément, affectant ainsi la réactivité globale du système. Afin de concilier ces deux extrêmes, nous avons donc opté pour une gestion automatisée des fils d'exécution par type de capteur. Cette approche intermédiaire permet de regrouper les capteurs similaires sur des fils d'exécution dédiés à leur catégorie, assurant ainsi une isolation partielle tout en limitant le nombre total de fils d'exécution créés. Ainsi, nous bénéficions d'une meilleure utilisation des ressources et d'une réactivité accrue, tout en maintenant une certaine flexibilité et facilité de maintenance dans la gestion des capteurs.

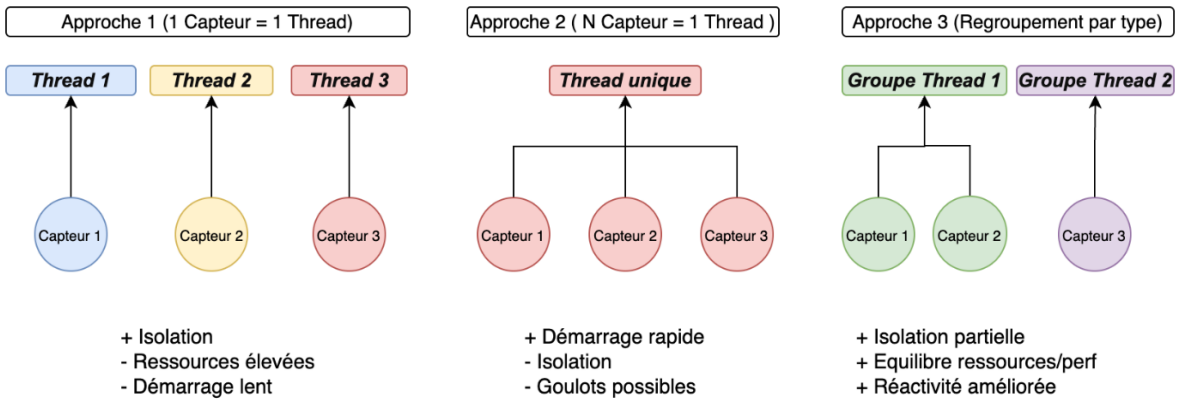

Figure 14 : Comparaison des stratégies de gestion des fils d'exécution de capteurs

Une fois le temps de geste du scénario de collecte écoulé, le service s'arrête, le participant entend un bip et une vibration. Les données collectées sont alors envoyées et disponibles dans les secondes qui suivent sur l'appareil mobile pour consultation par l'expérimentateur, ou peuvent être directement déposées dans le Cloud Storage de *Firebase* afin d'être exploitées ultérieurement ou intégrées à des logiciels tiers.

4.7 DÉFIS RENCONTRÉS ET SOLUTIONS

Dans le cadre du développement de l'application de collecte, plusieurs défis techniques ont dû être relevés. Beaucoup de ces problèmes ont été résolus, ou largement atténués, grâce à des solutions adaptées. L'un des principaux défis concernait l'envoi, la réception et la synchronisation des données entre la montre et le téléphone. Un problème essentiellement lié à la taille importante des données à transférer. Par exemple, lors de l'envoi de la commande de démarrage des capteurs qui inclut la configuration de la fréquence de démarrage et la liste des capteurs, il était nécessaire d'effectuer un transfert immédiat avec confirmation de la présence de l'appareil (nœud). Pour cela, nous avons opté pour l'envoi via un *MessageClient*. Toutefois, comme les capteurs des périphériques mobiles ne possèdent pas d'identifiant unique et fixe, il a fallu combiner plusieurs informations pour les identifier de manière unique. De plus, l'envoi des paramètres de configuration, notamment les informations de fréquence, rendait les données trop volumineuses, dépassant la limite autorisée par

le système de *MessagesClient*. Ces contraintes ont directement influencé notre méthode d'échange des données entre la montre et le téléphone. Nous avons donc scindé les informations de manière que la taille maximale de chaque message ne dépasse pas 100 Ko, restriction mise en place pour optimiser les performances et éviter que le transfert de données volumineuses ne surcharge les appareils. Pour les données plus importantes, nous avons privilégié leur transmission sous forme de fichiers ou d'assets, conformément aux recommandations de la documentation [27].

L'utilisation d'énergie et la surchauffe de l'appareil étaient aussi un problème majeur, car les capteurs qui fonctionnent en continu affectent considérablement l'autonomie de la montre. Pour minimiser cette contrainte, des optimisations ont été intégrées, comme l'ajustement des fréquences de collecte, une meilleure gestion des fils d'exécution et la mise en veille des capteurs inutilisés. Cependant, compte tenu des exigences liées à la collecte permanente, les marges d'amélioration restent limitées.

Une autre des principales contraintes rencontrées lors de l'utilisation des montres connectées réside dans la gestion de leurs ressources limitées, notamment leur faible capacité en mémoire RAM et en stockage interne. Lors d'une collecte, la mémoire se remplit rapidement en raison du volume important de données générées par nos capteurs, ce qui peut entraîner des pertes d'informations. Pour y remédier, nous avons mis en place un système complet qui compresse les données en temps réel avant leur écriture dans la mémoire de stockage de l'appareil et qui gère de manière optimisée les tampons en découpant les informations en lots pour un transfert efficace vers le stockage. Ce dispositif intègre également un mécanisme de sauvegarde temporaire permettant de récupérer les enregistrements en cas de saturation, tout en s'appuyant sur un traitement asynchrone en arrière-plan et l'utilisation du fil de travail pour exécuter les tâches de façon différée, assurant ainsi une collecte fluide et fiable, même dans des environnements aux ressources très restreintes.

4.8 AMÉLIORATION FUTURE

Bien que l'application réponde aux exigences actuelles du projet, plusieurs pistes d'amélioration peuvent être envisagées pour optimiser ses performances, enrichir ses fonctionnalités

et renforcer son adaptabilité à des scénarios plus complexes. Ces améliorations se concentrent sur le temps de latence entre le téléphone et la montre, la gestion des capteurs, l'efficacité énergétique, la collaboration entre utilisateurs et l'intégration de fonctions avancées, afin de répondre aux besoins croissants des futurs expérimentateurs.

Une première amélioration pourrait être liée à une optimisation réduisant le temps latent entre la communication montre-téléphone et téléphone-montre. Une autre pourrait être l'intégration de capteurs tels que le GPS, le microphone et la caméra pour enrichir les données collectées. Le GPS fournirait des informations précises sur la localisation géographique, utiles pour les recherches en extérieur ou dans des environnements spécifiques. Le microphone permettrait d'enregistrer des sons ou interactions vocales, tandis que la caméra capturerait des vidéos ou des photos des gestes pour valider et compléter les données des capteurs. Ces fonctionnalités, activées selon les besoins, offriraient une contextualisation plus riche et de nouvelles perspectives pour les analyses.

Une autre fonctionnalité essentielle à développer serait la prise en charge des séances multitâches. Cette option permettrait de planifier et d'exécuter plusieurs expériences simultanément au sein d'une même session. De plus, la possibilité de stocker, gérer et partager facilement ces expériences avec d'autres utilisateurs renforcerait la collaboration. Un système en temps réel offrirait une flexibilité accrue, permettant à plusieurs chercheurs de travailler simultanément, d'accéder aux données recueillies, de personnaliser les protocoles expérimentaux, ou encore d'annoter les résultats pour un traitement ultérieur. Ce type de collaboration active ouvrirait la voie à des projets d'équipe plus efficaces et coordonnés.

Pour améliorer l'expérience des utilisateurs, l'interface de l'application pourrait intégrer des visualisations en temps réel des données collectées. Cela permettrait aux expérimentateurs de suivre le déroulement des expériences en direct et de détecter rapidement toute anomalie. Une autre amélioration utile serait la possibilité de renommer les capteurs, ce qui rendrait leur identification plus intuitive en fonction des gestes ou des expériences spécifiques. Une gestion claire et organisée des capteurs contribuerait à rendre l'analyse des données plus fluide et plus efficace.

Afin de répondre à des besoins futurs et de s'adapter à une base de produits plus large, l'application pourrait être portée sur d'autres plateformes, comme iOS. Cette extension augmenterait sa polyvalence et la rendrait accessible à un éventail plus large de dispositifs. Par ailleurs, le développement d'outils d'analyse de données spécifiques aux données collectées faciliterait grandement le traitement des données après les expériences. Une automatisation accrue de ces analyses augmenterait considérablement la productivité des chercheurs, leur permettant de se concentrer davantage sur l'interprétation des résultats.

En combinant ces améliorations, l'application pourrait devenir un outil encore plus puissant et flexible, parfaitement adapté aux défis et aux exigences des expérimentations modernes. Ces évolutions permettraient non seulement de répondre aux attentes actuelles, mais également d'anticiper les besoins futurs des chercheurs, tout en maximisant l'impact et l'efficacité de leurs travaux.

4.9 CONCLUSION

Le processus de développement de cette application a permis de créer un outil solide, modulaire et adapté aux besoins de la recherche en reconnaissance gestuelle. En s'appuyant sur une architecture bien conçue et des choix technologiques pertinents, l'application assure une collecte fiable et efficace des données tout en offrant une expérience utilisateur fluide. Les défis rencontrés ont été surmontés, rendant cet outil prêt pour des études futures et des applications élargies ; et les pistes d'amélioration ont été explorées.

CHAPITRE 5 : PIPELINE DE TRAITEMENT

Ce chapitre a pour but d'explorer les différents aspects des données afin de mettre en lumière leur diversité, leur qualité et leur pertinence, tout en identifiant les éventuelles limites ou variations prises en compte dans l'étude. Nous y présenterons également l'ensemble de la chaîne de traitement des données, ainsi que les algorithmes de machine learning utilisés.

5.1 DESCRIPTION, TYPES ET STRUCTURE DES DONNÉES COLLECTÉES

Le jeu de données collectées se compose de signaux associés à des codes de gestes correspondant aux actions effectuées par les participants. Ces données ont été enregistrées à l'aide des capteurs embarqués dans la Google Pixel Watch 3. L'organisation du jeu de données repose sur une structure hiérarchique où chaque participant dispose d'un dossier dédié contenant plusieurs fichiers au format CSV. Chacun représentant un type de geste réalisé. Nous avons recueilli 17 gestes différents de chaque participant, qui ont été exécutés cinq fois sauf celui du Wakeup qui lui a été exécuté 10 fois. Au total, nous avons collecté les données de 18 personnes, mais nous avons exclu l'une d'entre elles car ses données ont été utilisées uniquement pour tester et déterminer la meilleure fréquence suggérée aux capteurs pour la collecte.

Notre jeu de données se compose de 7 994 340 lignes réparties sur 28 colonnes, correspondant à 1 521 observations distinctes. Chaque ligne représente une mesure individuelle effectuée par un capteur à un instant T, dans le cadre d'un geste spécifique. Une observation regroupe ainsi l'ensemble des lignes mesurées lors d'une prise de données complète, incluant plusieurs capteurs.

Parmi les capteurs figurent notamment un accéléromètre, un gyroscope, un capteur de conductance cutanée, un moniteur de fréquence cardiaque, ainsi que d'autres (Tableau 2) permettant une analyse fine et multimodale des mouvements. L'ensemble des données a été collecté et étiqueté à l'aide de notre application mobile dédiée, spécialement développée pour capturer les gestes.

Chacune des 28 variables enregistrées joue un rôle dans l'identification, la classification et l'analyse des gestes. Par exemple, l'attribut deviceId est l'identifiant unique de chaque dispositif utilisé pour la collecte, garantissant la distinction entre les données provenant de différents appareils. Dans notre cas, nous n'avons qu'un seul dispositif utilisé (le même Google Pixel Watch 3).

L'attribut device_brand spécifie la marque du dispositif (par exemple, Google Pixel Watch 3). Cette information permet de retracer l'origine des données, notamment en cas de variabilité des performances entre les différents dispositifs. De plus, le champ device_types indique le type de dispositif utilisé, qu'il s'agisse d'une montre connectée ou d'un téléphone.

Pour ce qui est des gestes, le champ gestureCode attribue un code unique à chaque type de geste collecté. Cela facilite la catégorisation des données et leur association aux gestes spécifiques réalisés par les participants, garantissant ainsi un étiquetage précis pour les analyses. Les capteurs utilisés sont également identifiés grâce à des champs spécifiques. sensor_name fournit le nom du capteur (comme ECG Sensor (wake-up) ou Gravity Sensor), tandis que sensor_type est un code numérique identifiant le type de capteur, utile pour déterminer la catégorie et la fonction du capteur. En complément, sensor_type_name offre une description textuelle du type de capteur (par exemple, "Accelerometer"), et sensor_vendor mentionne le fabricant du capteur.

L'attribut take_id associe un identifiant unique à chaque session de collecte permettant de regrouper les données en fonction des gestes ou des sessions spécifiques. Les valeurs mesurées par les capteurs sont enregistrées dans des champs tels que valeur-x avec x appartenant à un entier naturel $\mathbb{N}$ et qui représente les données collectées sur plusieurs axes donnés (par exemple, l'axe X, Y, Z, etc.). Enfin, le champ z_timestamp fournit l'horodatage précis de chaque enregistrement. Cette information est cruciale pour synchroniser les données, notamment dans les scénarios où plusieurs capteurs ou dispositifs sont utilisés simultanément. En plus, elles nous permettent de faire une analyse de série temporelle.

5.1.1 DISTRIBUTION DES DONNÉES

Dans une optique d'évaluation de nos futurs modèles de classification, nous avons choisi de scinder notre jeu de données en deux sous-ensembles distincts. Plus précisément, trois dossiers correspondant à trois participants ont été mis de côté et ont servi exclusivement à la phase de test. Cette répartition représente environ 18 % du volume total de données, les 82 % restants étant utilisés pour l'entraînement des modèles. La nouvelle répartition des données d'entraînement est de 8 573 500 instances pour 1262 observations et 28 colonnes pour les données d'entraînement et 2 112 000 lignes pour 259 observations et 28 colonnes pour les données de test.

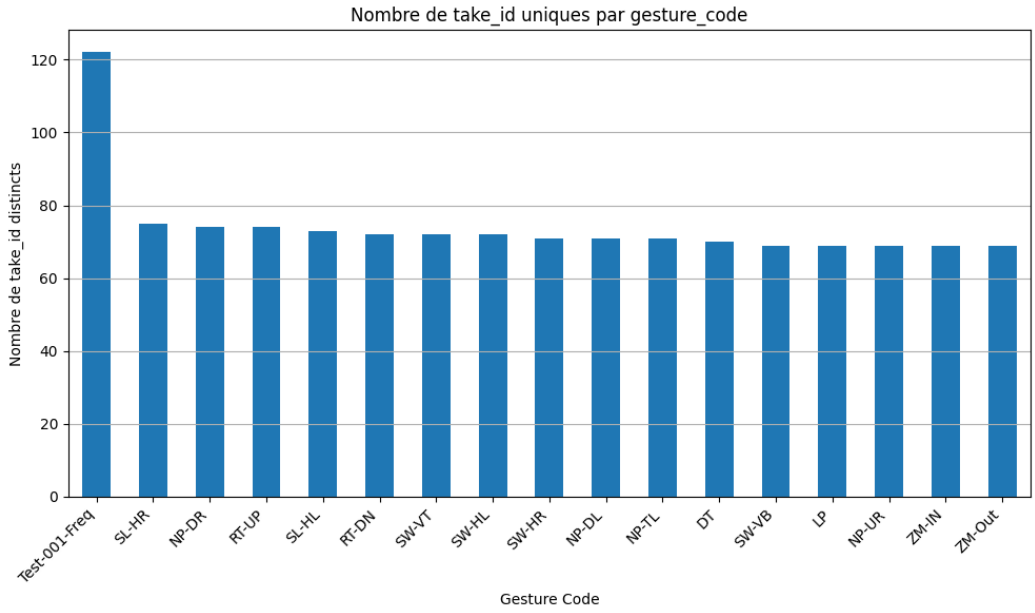

Figure 15 : Répartition des prises de données par type de geste dans l'ensemble d'entrainement

En ce qui concerne la distribution des données d'entraînement selon les gestes, on observe une distribution relativement uniforme du nombre d'instances par geste, à l'exception du geste Test-001-Freq, qui est environ deux fois plus grand que les autres (Figure 15). Cela s'explique par le fait que ce geste d'activation a été effectué deux fois lors de chaque session de collecte.

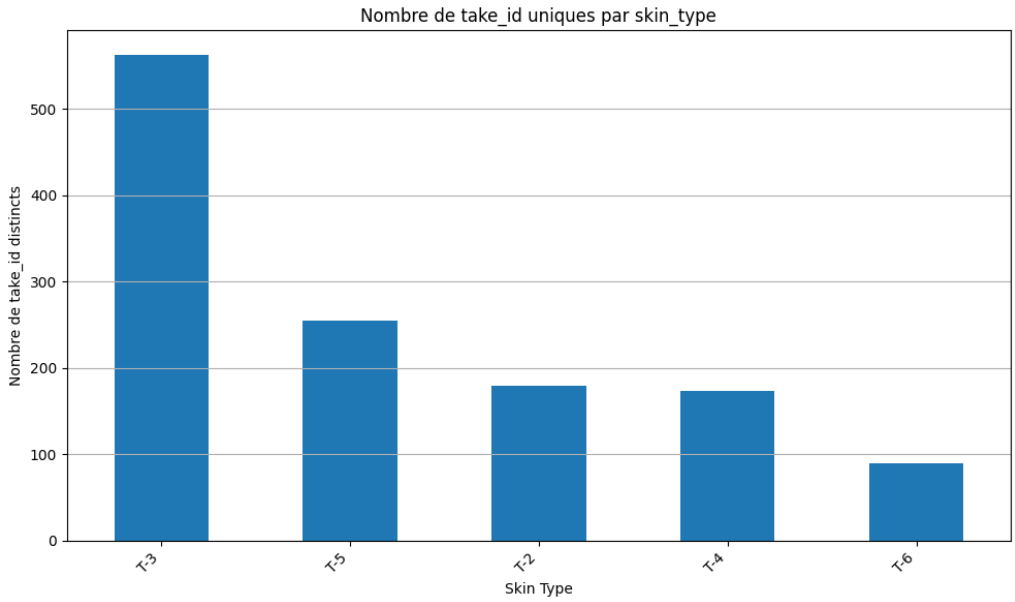

Figure 16 : Répartition des prises de données par type de peau dans l'ensemble d'entrainement

Du point de vue des caractéristiques liées au type de peau des participants, identifiées par un code T suivi du numéro correspondant (par exemple, T-3 pour le type de peau 3), on observe une surreprésentation des individus ayant un type de peau T-3, conformément à la classification de *Fitzpatrick* (Figure 16). Ce système, largement utilisé en dermatologie et en recherche biomédicale, classe la peau humaine en six types (de T-1 à T-6) selon la couleur de la peau et sa réaction à l'exposition solaire (capacité à bronzer ou tendance à brûler). Il est notamment utilisé pour anticiper certaines réponses cutanées à des traitements ou à des dispositifs technologiques (comme les capteurs portés sur la peau), mais aussi dans les études portant sur l'analyse des différences interindividuelles.

Dans notre étude, cette mesure nous permet d'évaluer si certaines caractéristiques physiologiques liées à la peau pourraient influencer la qualité des signaux captés (par exemple, en lien avec la conductivité ou l'adhérence des dispositifs). Le déséquilibre observé en faveur du type T-3 s'explique par la composition naturelle de notre échantillon : la majorité des participants ayant déclaré ce type de peau dans le questionnaire préliminaire d'inclusion.

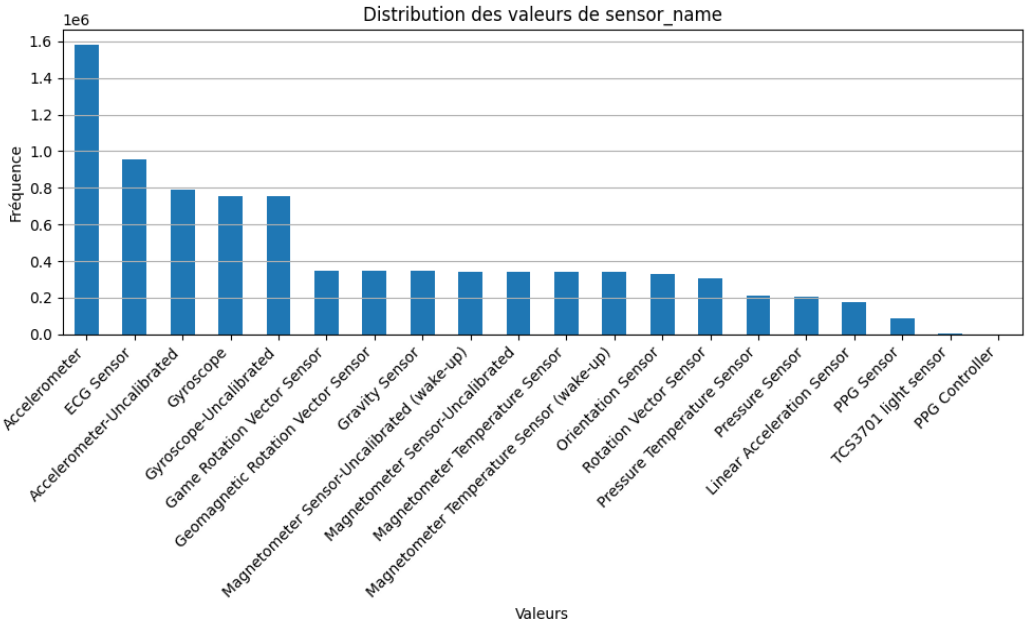

Figure 17 : Distribution des lignes par type de capteur

Comme l'indique la Figure 17, la distribution des données présente une hétérogénéité marquée au niveau de la distribution issue des capteurs. Cette variabilité s'explique par un ensemble de facteurs techniques inhérents au fonctionnement des dispositifs des capteurs sous Android.

En premier lieu, la fonctionnalité propre à chaque capteur, sa nature technologique ainsi que ses fréquences d'échantillonnage minimale et maximale influencent directement le volume et la régularité des données générées. Certains capteurs, tels que les accéléromètres et les gyroscopes, sont conçus pour opérer en mode continu, produisant ainsi un flux de données stable et soutenu. D'autres capteurs, en revanche, fonctionnent de manière événementielle ou intermittente, ne collectant des données qu'à l'occasion de stimulations particulières, souvent contextuelles ou prédéfinies.

En outre, il est essentiel de rappeler, conformément aux spécifications de la documentation officielle [21] du système Android, que les fréquences d'échantillonnage définies lors de l'enregistrement d'un capteur constituent uniquement des suggestions adressées au système

d'exploitation. Le respect effectif de ces fréquences n'est en aucun cas garanti. Android peut ajuster dynamiquement la fréquence d'échantillonnage, même lorsque l'application demande un mode rapide tel que *SENSOR_DELAY_FASTEST.* Le SensorDirectChannel [28] constitue une exception : Il permet au capteur d'écrire directement ses données dans un buffer partagé et fournit une fréquence dictée par le matériel sans rééchantillonnage par Android. Son utilisation reste toutefois limitée à quelques capteurs haute performance, dépend fortement du support matériel, ne permet pas de définir précisément la fréquence et nécessite une intégration plus complexe.

En pratique, le capteur est conditionné par divers paramètres, tels que les capacités matérielles du capteur, l'état de charge du processeur, ou encore les stratégies de gestion énergétique adoptées par l'appareil. Il en résulte une incertitude structurelle sur la régularité temporelle des mesures qu'il convient de prendre en compte lors des phases de traitement et d'analyse des données. Pour ces raisons, il a été préférable d'estimer la fréquence effective à partir des timestamps des événements captés ; ce qui permet d'obtenir une mesure plus fidèle du comportement réel du capteur dans son contexte d'usage. C'est dans cette perspective que nous avons procédé à une estimation empirique de la fréquence d'échantillonnage en calculant la période moyenne T séparant deux mesures successives, puis en appliquant la relation f = 1/T, où f représente la fréquence en Hertz (Hz) et T la période en secondes. Cette méthode nous a permis d'obtenir une fréquence moyenne plus représentative, utilisée comme référence dans les étapes d'analyse ultérieures. Ainsi, en observant la Figure 18, nous pouvons remarquer la forte disparité entre les fréquences d'échantillonnage des différents capteurs de la montre. Cette disparité confirme le mode de fonctionnement non contrôlé des capteurs sous Android.

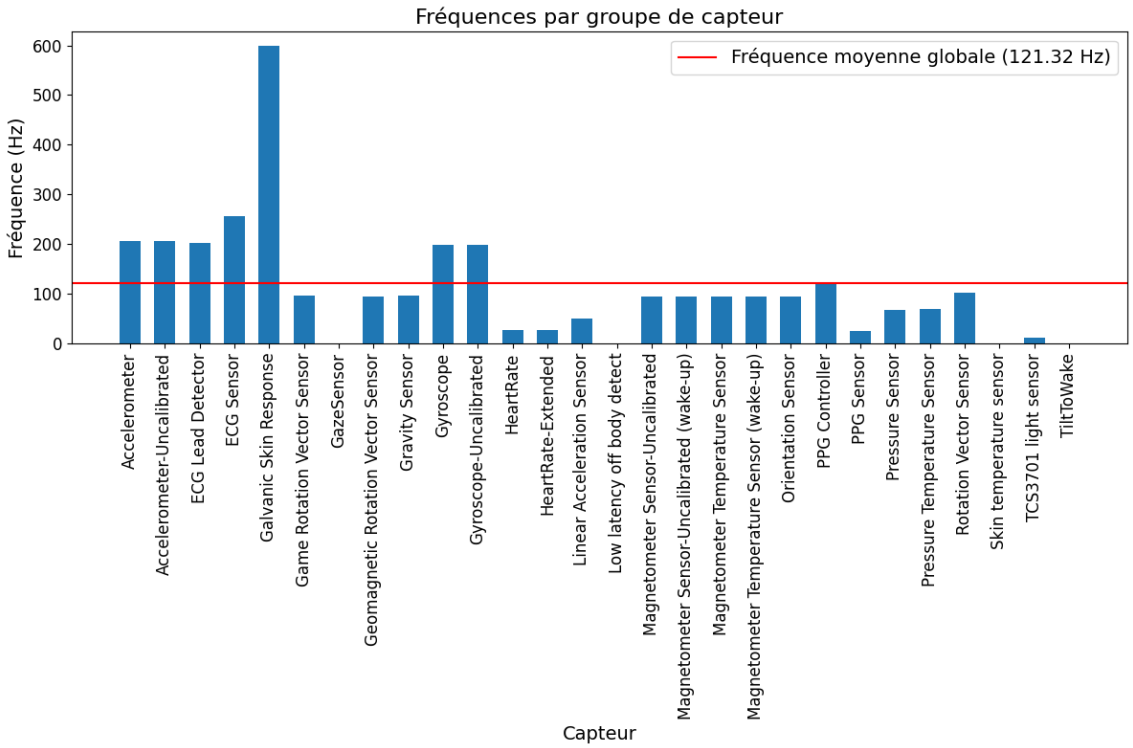

Figure 18 Fréquence d'échantillonnage réelle des capteurs

5.2 LES ÉTAPES DE TRAITEMENT (PIPELINE)

5.2.1 LE PRÉTRAITEMENT DES DONNÉES

a) Fusion des fichiers CSV

Les données initialement recueillies sont réparties dans plusieurs dossiers et fichiers au format CSV. Chacun des dossiers correspondant à un participant. Afin de constituer un jeu de données centralisé, l'ensemble de ces fichiers a été fusionné. Le résultat de cette opération a été sauvegardé au format Pickle, un format binaire propre à l'environnement Python. Ce choix s'explique par les avantages offerts en termes de rapidité de chargement, de préservation de la structure complexe des objets (notamment les DataFrames à index multiples) et de facilité de reprise du traitement sans transformation supplémentaire. L'utilisation du format Pickle s'inscrit ainsi dans une démarche d'efficacité et de reproductibilité des expériences.

b) Traitement des valeurs manquantes

Les valeurs manquantes observées dans le jeu de données s'expliquent par la nature et la diversité des capteurs mobilisés. Certains capteurs, comme les accéléromètres, génèrent plusieurs composantes (généralement X, Y et Z), tandis que d'autres, tels que les capteurs de température, ne fournissent qu'une seule variable. Les composantes ont été renommées selon le format "valeur x", où "x" représente l'indice de la variable. Ainsi, certaines colonnes restent naturellement vides lorsqu'elles ne s'appliquent pas à un capteur donné. Ces valeurs manquantes ne constituent donc pas une anomalie, mais reflètent la structure hétérogène du dispositif de mesure.

c) Gestion de doublons

Nous n'avons pas eu à gérer de données en double, puisque celles-ci ne se retrouvaient pas dans nos jeux de données.

d) Gestion des valeurs aberrantes

Aucune procédure de filtrage des valeurs extrêmes n'a été mise en œuvre. Les données ont été conservées dans leur intégralité, dans le but de respecter l'intégrité des mesures issues directement des capteurs. Toutefois, une exception a été faite pour les capteurs dont la fréquence effective d'échantillonnage était inférieure à 50 Hz. Ces capteurs ont été écartés de l'analyse, car leur cadence de mesure était jugée insuffisante pour capturer avec précision la dynamique des gestes.

e) Rééchantillonnage et synchronisation

Dans un contexte multicapteur, la synchronisation des flux de données représente un enjeu méthodologique majeur, en raison des disparités de fréquence d'échantillonnage et des décalages temporels inhérents à chaque capteur. Afin de remédier à ces désalignements, une procédure de rééchantillonnage uniforme a été mise en œuvre, accompagnée d'une interpolation linéaire. Cette stratégie visait à produire des séries temporelles homogènes, caractérisées par des intervalles réguliers entre les échantillons, facilitant ainsi l'alignement temporel des mesures.

Néanmoins, cette méthode s'est révélée limitée dans notre cas. En effet, les capteurs embarqués ne génèrent pas des données simultanément, et leur fréquence effective peut varier en fonction du système d'exploitation ou de la sollicitation des ressources. Par conséquent, le rééchantillonnage a conduit à l'introduction massive de valeurs manquantes (NaN) ou nulles, particulièrement dans les intervalles où certains capteurs n'émettaient aucun signal. Cette perte d'intégrité des données nuit directement à la qualité des analyses ultérieures.

Face à cette contrainte, nous avons opté pour une alternative qui consiste d'abord à faire un fenêtrage temporel des données brutes, suivi de l'extraction de caractéristiques statistiques et fréquentielles dans chaque fenêtre. Cette approche permet de résumer localement l'information contenue dans les signaux sans nécessiter un alignement parfait des échantillons à chaque instant, tout en préservant les dynamiques essentielles pour la modélisation des gestes.

f) Fenêtrage temporel

Les données ont été segmentées à l'aide d'une approche par fenêtres temporelles glissantes appliquée individuellement à chaque combinaison unique de capteur (sensor_name, sensor_type, sensor_vendor) et de prise (take_id). Chaque fenêtre a une durée fixe de 1 seconde (window_duration_sec = 1.0) et se déplace avec un pas de 0.3 seconde (step_duration_sec = 0.3), ce qui permet un recouvrement partiel entre les segments.

g) Extraction des caractéristiques

Afin de résumer efficacement le comportement du signal dans chaque fenêtre temporelle contenant au moins cinq échantillons, un ensemble de caractéristiques statistiques et fréquentielles a été extrait. Parmi les descripteurs temporels figurent la moyenne et l'écart-type, qui renseignent respectivement sur la tendance centrale et la dispersion du signal, ainsi que l'énergie RMS et l'écart interquartile (IQR), qui mesurent l'intensité et la variabilité de manière robuste. D'autres indicateurs tels que le taux de passage par zéro (ZCR), la skewness, la kurtosis et le nombre de pics détectés permettent de capturer la forme, la symétrie et la complexité du mouvement. Ces caractéristiques

sont couramment utilisées dans la littérature en reconnaissance d'activités, car elles offrent une représentation informative des signaux inertiels [29] [30], [31].

En complément, des caractéristiques fréquentielles ont été extraites à partir de la transformée de Fourier, notamment la fréquence dominante et l'énergie spectrale, afin de capter la structure rythmique et énergétique du signal, souvent déterminante pour distinguer des gestes similaires. L'ensemble de ces descripteurs permet ainsi de réduire la complexité des données tout en conservant les éléments discriminants nécessaires à la classification. Enfin, les vecteurs de caractéristiques sont enrichis de métadonnées contextuelles (telles que gesture_code, skin_type, sensor_type, etc.), puis exportés dans des fichiers CSV distincts pour chaque capteur et chaque prise, facilitant ainsi l'organisation et l'analyse ultérieure.

h) Fusion et structuration des caractéristiques par fenêtre

Une fois les caractéristiques extraites pour chaque capteur de manière individuelle, une phase de fusion a été réalisée afin de regrouper l'ensemble des descripteurs dans une structure cohérente. Concrètement, tous les fichiers CSV contenant les caractéristiques extraites par fenêtre (wi*ndow_index*) ont été chargés depuis le répertoire de sortie. Chaque fichier correspond à un capteur donné pour un *take_id* spécifique, et contient les statistiques extraites dans chaque fenêtre temporelle. Les fichiers valides c'est-à-dire ceux contenant les identifiants de fenêtre (*take_id, window_index*) ont été concaténés dans un unique DataFrame. Un regroupement a ensuite été effectué sur la base des identifiants de fenêtre pour éviter les duplications, en conservant la première occurrence de chaque combinaison (*groupby(...).first()).* Cette étape a permis de constituer une représentation tabulaire consolidée des signaux multi-capteurs, où chaque ligne correspond à une fenêtre temporelle unique, et chaque colonne à une caractéristique extraite. Enfin, un tri a été appliqué pour ordonner chronologiquement les fenêtres, et les colonnes entièrement vides ont été supprimées afin de nettoyer la structure. Ce jeu de données final, organisé par *take_id* et *window_index*, constitue la base de travail pour l'entraînement des modèles de classification, avec des vecteurs de caractéristiques homogènes et bien alignés.

5.2.2 STRUCTURATION DES DONNÉES POUR L'APPRENTISSAGE

a) Encodage et préparation finale

Avant l'entraînement des modèles de classification, les données ont été préparées à travers une série d'opérations de prétraitement. La variable cible (*gesture_code*) a été extraite, et les variables explicatives ont été isolées dans une matrice distincte. Les variables d'identifications de la session de collecte, le type de peau et celui identifiant le geste ont été converties en valeurs numériques par encodage ordinal rendant ainsi les données compatibles avec les algorithmes d'apprentissage automatique. Les valeurs manquantes dans les variables explicatives ont été imputées par la moyenne de chaque colonne, tandis que celles de la variable cible ont été remplacées par la valeur modale. Ce traitement a permis d'obtenir un ensemble de données complet, homogène et exclusivement numérique, prêt à être utilisé pour les phases d'entraînement et de validation des modèles.

b) Sélection de variables

Une sélection de variables par SelectKBest a été intégrée à un pipeline de validation croisée. Cette étape permet d'identifier les variables les plus informatives, d'éliminer les redondances et d'améliorer les performances des modèles.

5.2.3 ENTRAINEMENT ET ÉVALUATION DES MODÈLES

a) Entraînement

Au cours de cette phase, les caractéristiques extraites ont été utilisées comme variables explicatives pour l'entraînement de modèles de classification visant à reconnaître les gestes effectués. Pour ce faire, nous avons mobilisé plusieurs algorithmes d'apprentissage supervisé, notamment *K-Nearest Neighbors (KNN), XGBoost et Support Vector Machines (SVM),* l'optimisation des performances de ces modèles a été assurée par un ajustement systématique des hyperparamètres, réalisée à l'aide d'une recherche par grille (*grid search*), combinée à une validation

croisée. Les performances des différents modèles ont été évaluées par des métriques, telles que le F1-score, la précision, le rappel et la courbe d'apprentissage.

b) Sauvegarde

Les modèles et leurs hyperparamètres optimaux ont été enregistrés au format Pickle pour réutilisation.

5.3 EXPLORATION DES ALGORITHMES UTILISÉS

Afin d'analyser nos données, nous avons utilisé trois algorithmes différents. Tout d'abord, le SVM (Support Vector Machine), largement utilisé dans de nombreux cas d'études, il a la capacité à gérer efficacement des problèmes de classification et de régression. Il a été sélectionné en raison de sa capacité éprouvée à effectuer des tâches de classification avec une grande précision, même dans des espaces de caractéristiques complexes. Il est particulièrement adapté pour des données issues de capteurs où les classes ne sont pas facilement distinguables. Ensuite, nous avons eu recours au K-Nearest Neighbors (KNN), un algorithme simple, mais puissant pour la classification, basé sur la proximité dans l'espace des caractéristiques. Il a été intégré comme un algorithme de base afin de fournir une référence simple mais efficace. Sa logique intuitive fondée sur la proximité permet d'évaluer la cohérence de la structure des données dans l'espace des caractéristiques, et de comparer les performances avec des modèles plus complexes. Enfin, nous avons utilisé le XGBoost (eXtreme Gradient Boosting), une méthode d'ensemble avancée et polyvalente, optimisée non seulement pour la classification et la régression, mais aussi adaptée à des contextes plus complexes, tels que le ranking et la prédiction sur séries temporelles. Il a été choisi pour sa puissance prédictive, notamment dans les contextes où les relations entre variables sont non linéaires et multiples. Son approche d'ensemble basée sur le boosting va permettre d'atteindre des performances élevées et d'explorer des structures de données plus subtiles.

Dans cette section, nous présenterons de manière générale les algorithmes sélectionnés ainsi que leur fonctionnement. Nous décrirons également le pipeline complet mis en place, depuis le traitement des données jusqu'à l'interprétation des résultats. Enfin, nous explorerons les métriques

d'évaluation utilisées pour mesurer la performance des modèles et analyser leur pertinence dans le contexte de notre étude.

5.3.1 SUPPORT VECTEUR MACHINE (SVM)

Le Support Vector Machines (SVM), désignée dans l'article fondateur de Cortes et Vapnik [32] sous l'appellation « Support-Vector Networks » et traduit en français par Machine à Vecteurs de Support, est une méthode d'apprentissage automatique développée dans les années 1990. Il est utilisé pour résoudre des problèmes de classification et de régression. Son principe repose sur la séparation des données en différentes classes en traçant une frontière, appelée hyperplan, qui maximise la distance (ou marge) entre les groupes de données et cette frontière. Cette approche garantit une robustesse particulière pour la classification binaire et multiclasse.

Les SVM se concentrent sur la recherche de l'hyperplan de séparation optimal dans l'espace des caractéristiques, ce qui permet de gérer efficacement les cas où les données ne sont pas parfaitement séparables. Grâce à l'utilisation de kernels (ou noyaux), les SVM peuvent également traiter des données non linéaires, ce qui les rend très flexibles et adaptés à une grande variété de problèmes.

Les machines à vecteurs de support (SVM) présentent plusieurs avantages notables. Elles excellent dans les espaces de haute dimension, ce qui les rend idéales pour différentes applications, comme la classification des gestes. Grâce à des fonctions de noyau comme RBF ou polynomiales, elles gèrent efficacement les relations non linéaires. La fonctionnalité de marge souple leur confère une stabilité face aux valeurs aberrantes, ce qui est utile dans des domaines comme la détection d'anomalies très utile dans notre cas lié à la détection des gestes. De plus, les SVM sont adaptées à la classification binaire et multiclasse tout en étant économes en mémoire, car elles se concentrent uniquement sur les vecteurs de support. Cependant, elles ont aussi des limites. Leur entraînement peut être lent pour des ensembles de données volumineux, et le réglage des paramètres, comme le choix du noyau ou de la valeur, est souvent complexe et nécessite un réglage minutieux. Par ailleurs, elles sont sensibles aux données bruitées ou aux classes qui se chevauchent ; et leur modèle,

particulièrement dans les espaces de grande dimension, est difficile à interpréter. Enfin, une mise à l'échelle appropriée des caractéristiques est essentielle pour garantir des performances optimales, sous peine d'obtenir des résultats sous-optimaux.

5.3.2 K-NEAREST NEIGHBORS

Utilisé aussi bien pour la régression que pour la classification, le *K-Nearest Neighbors (KNN)[33]*, ou méthode des *K-Plus Proches Voisins* en français, est un algorithme conçu pour les analyses discriminantes, notamment lorsque l'estimation paramétrique fiable des densités de probabilité est inconnue ou difficile à établir. Cet algorithme, simple à comprendre, repose sur la distance entre une donnée à tester et celles de l'ensemble d'entraînement.

Le principe du *KNN* peut être illustré par l'analogie suivante : *"Dis-moi qui sont tes voisins, et je te dirai qui tu es."* Concrètement, l'algorithme identifie parmi les données d'entraînement les observations les plus proches de celles à analyser. Ensuite, pour une tâche de classification, l'étiquette de la donnée à prédire est déterminée en fonction de la majorité des classes parmi les K Plus proches voisins. Pour une tâche de régression, c'est la moyenne (ou la médiane) des valeurs cibles de ces voisins qui est utilisée pour prédire la valeur. L'importance du paramètre K réside dans le fait qu'il ne se limite pas à l'observation la plus proche, mais étend l'analyse à un nombre K fixé de voisins.

Pouvant être utilisé pour la régression et la classification, le principal avantage du KNN est qu'il est très facile à comprendre et ne nécessite pas de créer un modèle, de régler plusieurs paramètres ou de formuler des hypothèses supplémentaires. Cependant, il devient beaucoup plus lent à mesure que le nombre d'observations et de variables indépendantes augmente.

5.3.3 XGBOOST (EXTREME GRADIENT BOOSTING)

Développé en 2015, l'eXtreme Gradient Boosting (XGBoost) [34] est un algorithme d'apprentissage automatique évolutif devenu célèbre pour avoir permis à de nombreuses équipes de remporter des compétitions Kaggle. Basé sur une implémentation optimisée des méthodes

d’ensemble utilisant le Gradient Boosting, XGBoost repose sur des arbres de décision successifs pour corriger les erreurs des prédictions précédentes en minimisant une fonction de perte spécifique.

Ce qui distingue XGBoost, ce sont ses nombreuses optimisations telles que la régularisation intégrée (L1 et L2) qui réduit le risque de sur-apprentissage ; et la gestion native des valeurs manquantes qui simplifie le prétraitement des données. L’algorithme est également conçu pour tirer parti des ressources modernes, avec un support natif des données clairsemées, une exécution parallélisée, et la possibilité de s’exécuter de manière distribuée sur plusieurs machines ou via des GPU pour accélérer considérablement le traitement.

Sa flexibilité lui permet de s’adapter à une variété de tâches, y compris la classification, la régression, et le ranking, tout en prenant en charge des fonctions de perte personnalisées pour répondre à des besoins spécifiques. Malgré sa puissance, *XGBoost* nécessite un ajustement minutieux des hyperparamètres pour atteindre des performances optimales et reste moins adapté aux données non structurées (comme les images ou le texte brut) où les réseaux neuronaux profonds sont souvent préférables.

CHAPITRE 6 : RÉSULTAT

Cette section présente de manière structurée les principaux résultats obtenus à l'issue du protocole expérimental. Conformément aux hypothèses formulées dans la section méthodologique, les analyses réalisées visent à évaluer la pertinence des choix techniques ainsi que l'efficacité des solutions mises en œuvre. Les résultats sont organisés en fonction des algorithmes utilisés, afin de mettre en lumière les différentes dimensions explorées dans cette étude. Pour chaque algorithme, les performances sont détaillées à travers des indicateurs clés (ex. : précision, rappel, F-mesure), accompagnés de visualisations et de commentaires permettant d'en faciliter l'interprétation.

6.1 RÉSULTAT DU SUPPORT VECTEUR MACHINE (SVM)

Les performances du Support Vector Machine (SVM) ont été évaluées à l'aide de plusieurs indicateurs, dont la précision globale, la courbe d'apprentissage et le rapport de classification. La précision moyenne obtenue sur l'ensemble de validation est de 0,148, ce qui indique des performances limitées dans la tâche de classification multi-classes considérée.

TABLEAU 4 : Tableau récapitulatif des résultats du modèle SVM par classe

Gestes	Classe	Précision	Rappel	F1-score	Support
DT	0	0.16	0.08	0.11	109
LP	1	0.0	0.0	0.0	104
NP-DL	2	0.0	0.0	0.0	101
NP-DR	3	0.19	0.04	0.06	103
NP-TL	4	0.18	0.02	0.03	104
NP-UR	5	0.12	0.07	0.09	99
RT-DN	6	0.17	0.17	0.17	114
RT-UP	7	0.12	0.32	0.18	116
SL-HL	8	0.03	0.03	0.03	101
SL-HR	9	0.06	0.03	0.04	107
SW-HL	10	0.08	0.14	0.1	104
SW-HR	11	0.0	0.0	0.0	105
SW-VB	12	0.14	0.16	0.15	107
SW-VT	13	0.2	0.01	0.02	105
Test-001-Freq	14	0.85	0.54	0.66	204
ZM-IN	15	0.07	0.52	0.13	102
ZM-Out	16	0.0	0.0	0.0	106
Moyenne (macro)	**Moyenne (macro)**	0.14	0.13	0.1	--

Moyenne pondérée	**Moyenne pondérée**	0.18	0.15	0.13	1891

6.1.1 COURBE D'APPRENTISSAGE

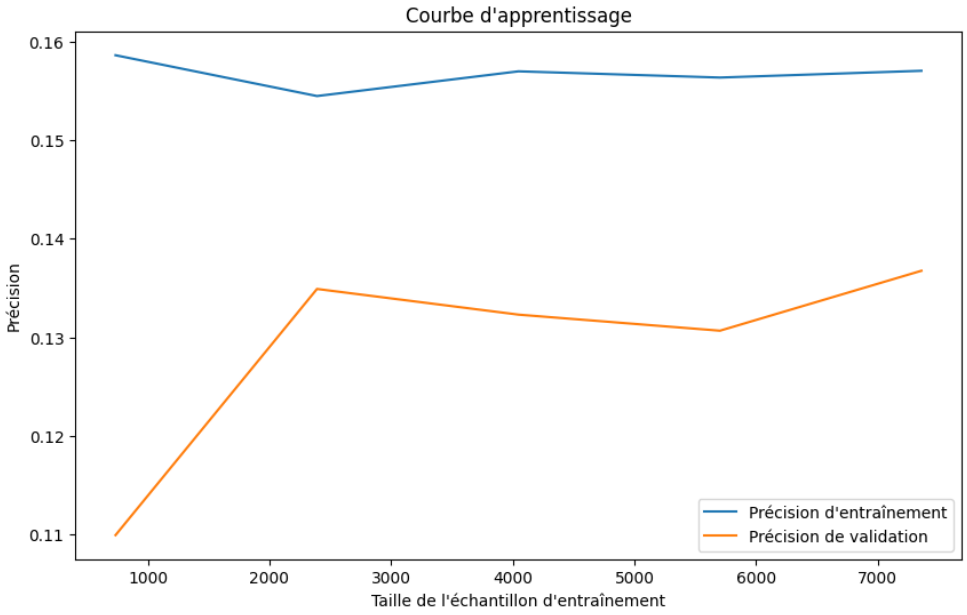

Figure 19 : Courbe d'apprentissage du modèle SVM

La figure (Figure 19) illustre l'évolution de la précision en fonction de la taille de l'échantillon d'entraînement. On observe un écart relativement stable entre la précision sur l'ensemble d'entraînement (autour de 0,155) et celle sur l'ensemble de validation, qui reste globalement inférieure (autour de 0,13). Cette courbe indique que le modèle n'est pas en surapprentissage (overfitting), car la précision d'entraînement est relativement basse. Toutefois, la précision de validation n'augmente pas significativement avec la taille des données, ce qui peut refléter une capacité limitée du modèle à généraliser ou un sous-apprentissage (underfitting). Une hypothèse serait celle liée à la quantité insuffisante de données permettant de faire la classification de nos gestes étant donné que nous n'atteignons pas de plateau au niveau de la performance de validation. Nous émettons l'hypothèse qu'un volume de données insuffisant freine l'amélioration de la performance de validation, qui n'atteint pas de plateau. Cette hypothèse se trouve renforcée par le

fait que le geste Wakeup qui dispose du plus grand nombre d'exemples est également celui qui est le mieux prédit.

6.1.2 RAPPORT DE CLASSIFICATION

Le tableau présenté expose les scores de précision, de rappel et de F1-score obtenus pour chaque classe à l'aide du modèle SVM. Il ressort que certaines classes, notamment la classe 14 (*Test-001-Freq*) qui est celle du geste WakeUp, se distinguent par des performances nettement supérieures (F1-score de 0,66), ce qui suggère une meilleure représentativité de ces données ou une plus grande facilité de discrimination par le modèle. À l'inverse, plusieurs classes telles que les classes 1, 2, 11 et 16 qui sont respectivement les gestes d'appuie long, de clavier numérique touche bas gauche, balayage horizontale gauche vers droite, et le zoom en arrière, obtiennent des scores nuls, indiquant une incapacité totale du modèle à les reconnaître correctement. Cette disparité dans les performances laisse supposer que le modèle favorise certaines classes au détriment d'autres, probablement en raison d'un déséquilibre dans la répartition des données d'apprentissage ou d'une complexité inhérente à la reconnaissance de certaines gestuelles. De manière générale, ces résultats suggèrent que, dans sa configuration actuelle, le SVM n'offre pas des performances satisfaisantes pour la classification multi-classes envisagées. Une optimisation plus poussée du modèle, incluant le réglage des hyperparamètres (comme le choix du noyau ou la régularisation) ainsi qu'un prétraitement plus rigoureux des données (par exemple via une réduction de dimensionnalité ou un rééquilibrage des classes), pourrait potentiellement améliorer les résultats obtenus.

6.1.3 MATRICE DE CONFUSION

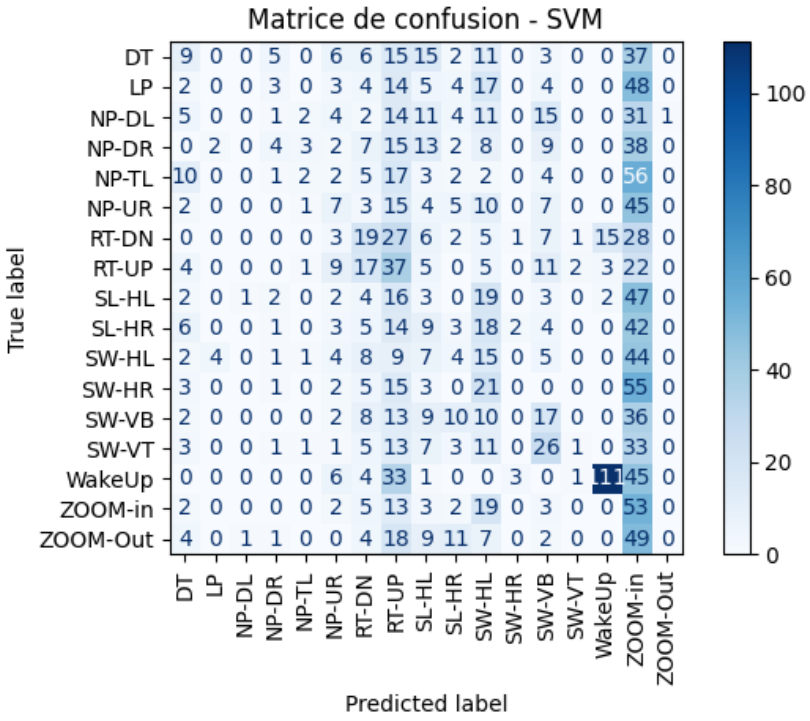

Figure 20 : Matrice de confusion du SVM

L'analyse de la matrice de confusion (Figure 20) permet d'approfondir la compréhension des performances de chaque algorithme, au-delà des simples taux de précision globaux. Pour le modèle SVM, bien que la précision atteigne environ 14,8 %, les erreurs de classification révèlent une tendance nette à surclasser de nombreux gestes dans la classe 14 (Test-001-Freq ou WakeUp). Cette prédominance peut s'expliquer par une surreprésentation de cette classe dans l'ensemble d'entraînement, mais également par sa gestuelle plus distincte, conduisant le modèle à y projeter les exemples ambigus. Les gestes appartenant à des familles proches, comme ceux du clavier numérique (NP-DL, NP-DR, NP-TL) ou les mouvements directionnels (SW-HL, SW-HR, SL-HR), sont fréquemment confondus entre eux, ce qui souligne la difficulté du modèle à capter les différences subtiles dans des signaux parfois très proches sur le plan spatial.

6.2 RÉSULTAT DU XGBOOST (EXTREME GRADIENT BOOSTING)

Le modèle XGBoost a été évalué sur les mêmes données que les autres algorithmes afin de mesurer sa capacité à classifier les gestes à partir des caractéristiques extraites. La précision moyenne atteinte sur l'ensemble de validation est de 0,148 ; similaire à celle obtenue avec le SVM.

TABLEAU 5 : Résultats du modèle XGBoost par classe

Gestes	Classe	Précision	Rappel	F1-score	Support
DT	0	0.33	0.06	0.11	109
LP	1	0.07	0.03	0.04	104
NP-DL	2	0.23	0.03	0.05	101
NP-DR	3	0.0	0.0	0.0	103
NP-TL	4	0.1	0.08	0.09	104
NP-UR	5	0.0	0.0	0.0	99
RT-DN	6	0.12	0.11	0.11	114
RT-UP	7	0.15	0.39	0.22	116
SL-HL	8	0.07	0.09	0.08	101
SL-HR	9	0.13	0.08	0.1	107
SW-HL	10	0.08	0.1	0.09	104
SW-HR	11	0.06	0.03	0.04	105
SW-VB	12	0.11	0.06	0.07	107
SW-VT	13	0.15	0.03	0.05	105
Test-001-Freq	14	0.2	0.77	0.31	204
ZM-IN	15	0.05	0.03	0.04	102
ZM-Out	16	0.14	0.03	0.05	106
Moyenne (macro)	**Moyenne (macro)**	0.12	0.11	0.09	--
Moyenne pondérée	**Moyenne pondérée**	0.12	0.15	0.1	1891

6.2.1 COURBE D'APPRENTISSAGE

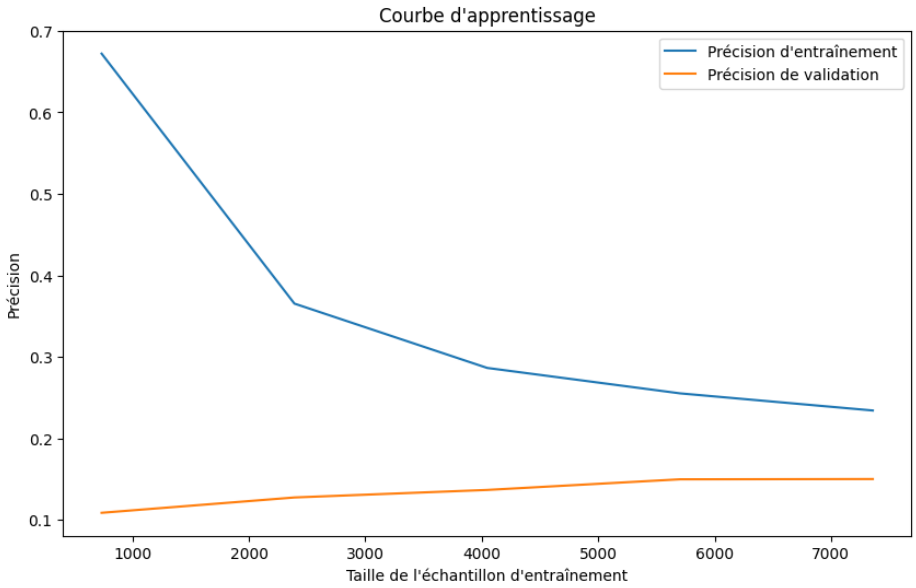

Figure 21 : Courbe d'apprentissage du modèle XGBOOST

La courbe d'apprentissage (Figure 21) montre une forte décroissance de la précision d'entraînement à mesure que la taille de l'échantillon augmente, ce qui témoigne d'un comportement initial de surapprentissage rapidement corrigé. En revanche, la précision de validation progresse lentement et reste relativement faible, ce qui laisse entrevoir une limitation dans la capacité de généralisation du modèle, possiblement en raison de la complexité du jeu de données ou de la difficulté à capturer des motifs discriminants suffisants. Une autre hypothèse serait toujours celle liée à la quantité insuffisante de données permettant de faire la classification de nos gestes étant donné que nous n'atteignons pas de plateau au niveau de la performance de validation.

6.2.2 RAPPORT DE CLASSIFICATION

Le rapport de classification détaillé (Tableau 3) met en évidence une forte variabilité des performances selon les classes. La classe 14 (WakeUp) obtient un score F1 élevé (0,31) grâce à un bon rappel (0,77), indiquant que cette classe est bien identifiée par le modèle. D'autres classes,

comme la classe 7 (F1-score de 0,22), présentent également des résultats acceptables. En revanche, plusieurs classes (par exemple les classes 3, 5 ou 11) affichent des scores nuls ou très faibles, ce qui témoigne d'une incapacité du modèle à les reconnaître correctement. Cette hétérogénéité pourrait s'expliquer par un déséquilibre dans la distribution des données ou par une similitude entre les signaux de certaines classes rendant leur différenciation difficile.

Dans l'ensemble, bien que le modèle XGBoost offre des performances comparables à celles du SVM, il ne parvient pas à fournir une classification fiable sur l'ensemble des gestes. Une amélioration pourrait être envisagée via un réglage plus fin des hyperparamètres (ex. : profondeur des arbres, taux d'apprentissage), une augmentation de la quantité ou de la qualité des données, ou encore l'usage de méthodes d'équilibrage pour corriger la distribution des classes.

6.2.3 MATRICE DE CONFUSION

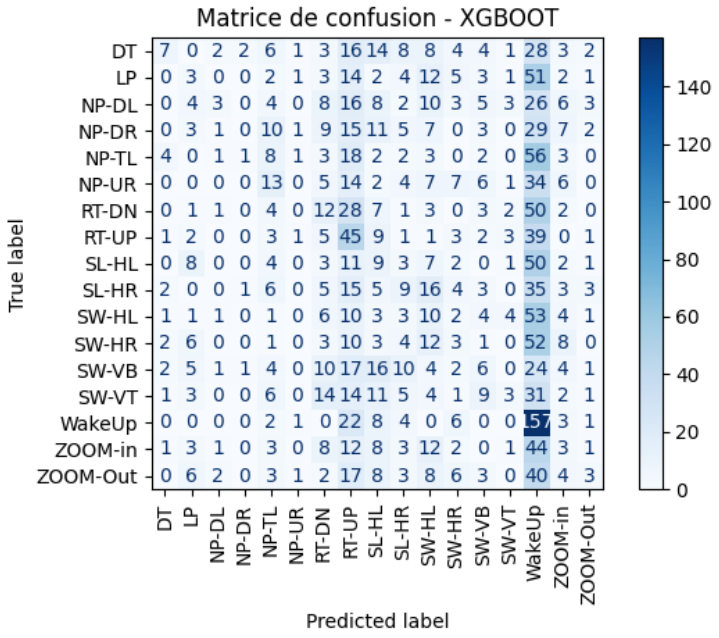

Figure 22 : Matrice de confusion du XGBOOST

Bien que la précision globale soit équivalente à celle du SVM, la distribution des erreurs diffère légèrement. Le modèle semble plus sensible à la variabilité entre classes et présente une dispersion plus équilibrée des erreurs, sans pour autant échapper à une confusion persistante autour de la classe WakeUp. Les gestes dynamiques comme les balayages (SW) ou les glissements (SL) restent particulièrement difficiles à différencier, ce qui peut s'expliquer par leur forte similarité directionnelle et leur temporalité continue, peu évidente à discriminer à partir des données de capteurs brutes.

6.3 RÉSULTAT DU K-NEAREST NEIGHBORS

Le modèle K-Nearest Neighbors (KNN) a été évalué sur la même tâche de classification multi-classes. Il affiche une précision moyenne relativement faible, atteignant 0,116, ce qui constitue la performance la plus basse parmi les trois algorithmes testés. Le rapport de classification (Tableau 6) met en évidence une faiblesse généralisée dans la reconnaissance des gestes, avec des scores de F1 très bas pour la majorité des classes. Seule la classe 14 se distingue avec un F1-score de 0,39 grâce à un rappel élevé (0,55), ce qui suggère une meilleure détectabilité de cette classe possiblement liée à des caractéristiques distinctives plus marquées. En revanche, plusieurs classes telles que les classes 0, 2, ou 5, affichent des scores inférieurs à 0,05 témoignant de la difficulté du modèle à identifier correctement ces gestes.

TABLEAU 6 : Résultats du modèle KNN par classe

Gestes	Classe	Précision	Rappel	F1-score	Support
DT	0	0.03	0.02	0.02	109
LP	1	0.07	0.08	0.08	104
NP-DL	2	0.02	0.02	0.02	101
NP-DR	3	0.05	0.05	0.05	103
NP-TL	4	0.07	0.07	0.07	104
NP-UR	5	0.05	0.04	0.04	99
RT-DN	6	0.09	0.05	0.07	114
RT-UP	7	0.16	0.18	0.17	116
SL-HL	8	0.06	0.07	0.06	101
SL-HR	9	0.06	0.07	0.07	107
SW-HL	10	0.07	0.05	0.06	104
SW-HR	11	0.05	0.05	0.05	105
SW-VB	12	0.06	0.05	0.05	107

SW-VT	13	0.13	0.09	0.1	105
Test-001-Freq	14	0.3	0.55	0.39	204
ZM-IN	15	0.07	0.07	0.07	102
ZM-Out	16	0.06	0.06	0.06	106
Moyenne (macro)	**Moyenne (macro)**	0.08	0.09	0.08	--
Moyenne pondérée	**Moyenne pondérée**	0.09	0.12	0.1	1891

6.3.1 RAPPORT DE CLASSIFICATION

Contrairement à d'autres modèles, le KNN ne bénéficie pas d'une phase d'apprentissage explicite, ce qui le rend particulièrement sensible à la structure locale des données et aux choix des paramètres (notamment la valeur de k et la distance utilisée). Les résultats obtenus ici suggèrent que le modèle KNN, dans sa configuration actuelle, manque de capacité de généralisation pour traiter efficacement des données complexes et bruitées, comme celles utilisées dans cette étude. Une amélioration potentielle passerait par un meilleur réglage de k, l'utilisation de pondérations adaptées à la distance, ou encore une réduction de la dimensionnalité pour atténuer les effets du "fléau de la dimension".

6.3.2 COURBE D'APPRENTISSAGE

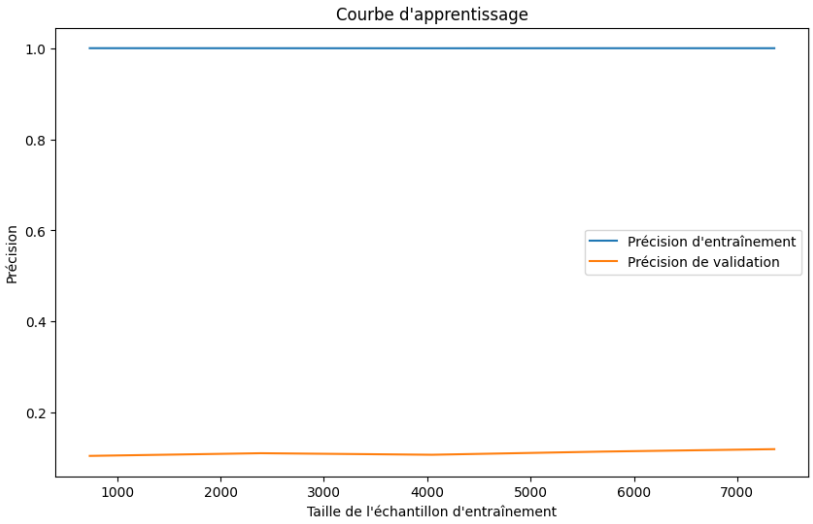

Figure 23 : Courbe d'apprentissage du modèle KNN

La courbe d'apprentissage du modèle KNN (voir Figure 23) révèle un comportement caractéristique d'un surapprentissage massif (overfitting). En effet, la précision sur l'ensemble d'entraînement est quasi parfaite (1.0) quelle que soit la taille de l'échantillon, ce qui signifie que le modèle mémorise les exemples sans généraliser. En revanche, la précision sur l'ensemble de validation demeure très faible et reste globalement constante autour de 0,11 à 0,12, sans amélioration notable avec l'augmentation des données d'entraînement. Cette divergence marquée entre les courbes illustre l'incapacité du modèle à apprendre des représentations généralisables, et reflète une forte sensibilité aux données d'entraînement, typique du KNN lorsque les données sont complexes ou de haute dimension. Ce constat est cohérent avec les faibles scores observés dans le rapport de classification, et confirme que le modèle, sans traitement préalable ou ajustement fin des paramètres, ne parvient pas à capturer efficacement les structures sous-jacentes du problème de classification des gestes.

6.3.3 MATRICE DE CONFUSION

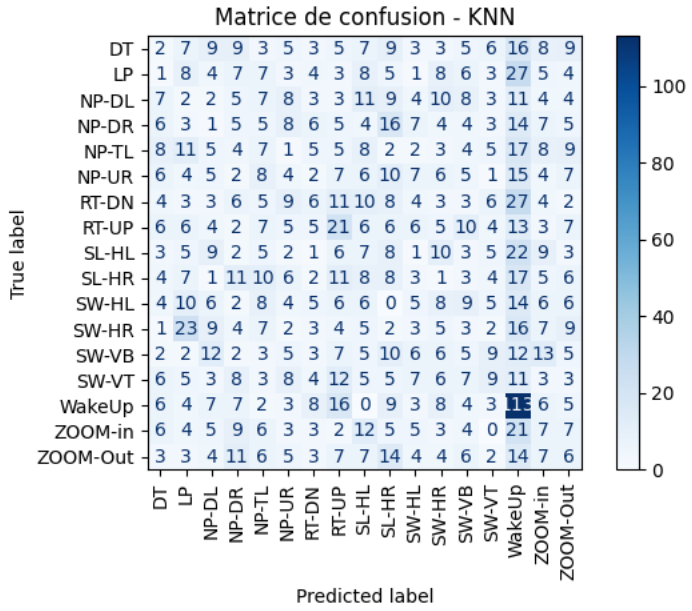

Figure 24 : Matrice de confusion du KNN

La matrice de confusion (Figure 24) est relativement homogène dans ses erreurs. Aucun geste ne domine clairement les prédictions, ce qui reflète une certaine difficulté du modèle KNN à établir des frontières fiables entre les classes dans un espace fortement bruité et multidimensionnel. Cette faiblesse du KNN confirme la similarité dans les caractéristiques des gestes ainsi qu'au déséquilibre des classes.

6.4 COMPARAISON ET DÉDUCTION GÉNÉRALE

L'analyse comparative des trois modèles de classification testés SVM, XGBoost et KNN met en lumière des performances globalement faibles, mais révèle des comportements distincts face aux données multicapteurs liées à la reconnaissance de gestes. Le SVM affiche une précision moyenne de 14,8 %, avec une courbe d'apprentissage relativement stable. Il montre une certaine capacité à généraliser sans surapprentissage excessif, mais peine à distinguer correctement plusieurs classes, sans doute en raison de frontières de décision trop rigides dans un espace de données complexe. Le modèle XGBoost, plus flexible, atteint un niveau de précision similaire, mais se distingue par une

meilleure capacité à détecter certaines classes spécifiques, notamment la classe 14, avec un rappel de 0,77, ce qui témoigne de sa faculté à modéliser des interactions non linéaires. Toutefois, cette performance reste hétérogène selon les classes. Le modèle KNN, bien que simple à implémenter, se révèle clairement le moins performant. Il présente un surapprentissage extrême (précision d'entraînement de 1.0) tout en échouant à généraliser (précision de validation autour de 11 %), ce qui illustre une incapacité à extraire des régularités générales à partir de données bruitées et complexes.

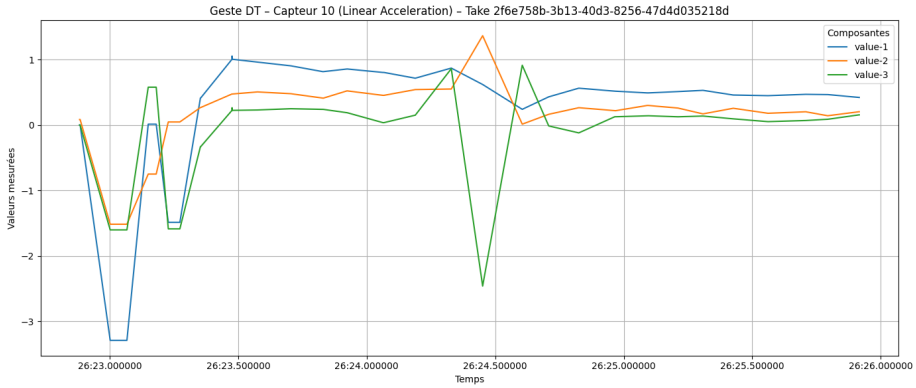

Figure 25 : Visualisation d'un signal d'une prise de données pour le geste de double tapotement

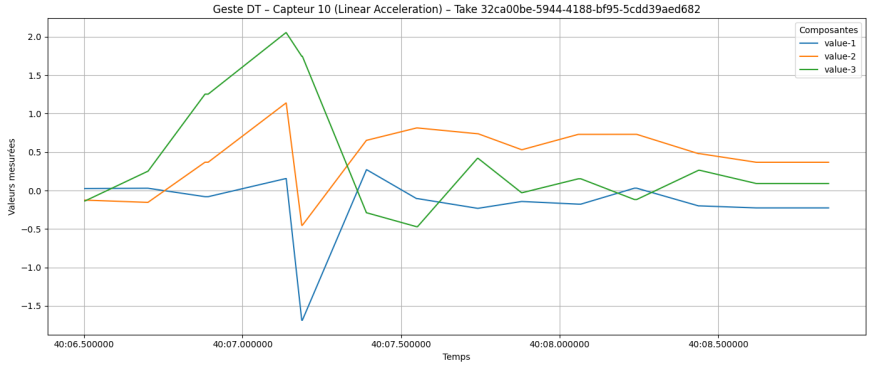

Figure 26 : Visualisation d'un signal d'une prise de données pour le geste de double tapotement

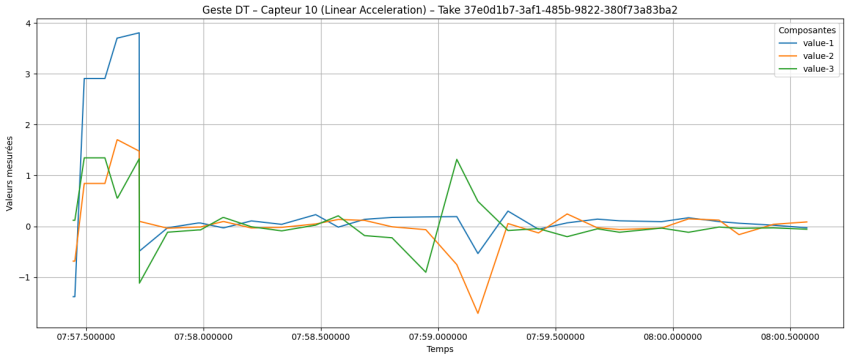

Figure 27 : Visualisation d'un signal d'une prise de données pour le geste de double tapotement

Au-delà des choix algorithmiques, ces résultats soulignent que la qualité des données, leur équilibrage entre les classes, ainsi que la structure intrinsèque des signaux des capteurs jouent un rôle déterminant dans la performance des modèles. Nous pouvons le voir sur les figures 25, 26 et 27 que, par exemple pour un même geste issu des données d'un même capteur, nous avons visuellement une très grande différence dans l'amplitude des données collectées.

CHAPITRE 7 : DISCUSSIONS

L'analyse des données collectées lors de l'entraînement des modèles a permis de mettre en évidence plusieurs obstacles majeurs à la reconnaissance fiable des gestes à partir des capteurs intégrés aux montres connectées fonctionnant sous Android/WearOS. Ces difficultés tiennent à la fois aux caractéristiques des capteurs embarqués, à leur gestion par le système Android, et à l'absence de documentation technique unifiée.

Une première difficulté concerne la variabilité du flux de données entre les modes debug et release. En mode debug, l'utilisation du fils d'exécution principal pour la journalisation ralentit la collecte, ce qui réduit considérablement la densité des données. Par exemple, certains fichiers ne contiennent que 2 800 instances pour 915 Ko, contre plus de 10 000 pour 3,1 Mo en mode release pour une même durée. De plus, un comportement progressif dans l'activation des capteurs a été observé au lancement des enregistrements. Concrètement, les capteurs ne délivrent pas immédiatement un flux de données constant et complet. Un délai est nécessaire avant que le débit atteigne un niveau relativement stable. Cette latence pourrait s'expliquer par les mécanismes internes d'optimisation énergétique mis en œuvre par Android, qui limitent temporairement l'activité des capteurs pour préserver l'autonomie de l'appareil. Durant cette phase transitoire, les données recueillies sont souvent incomplètes ou peu représentatives, entraînant une sous-représentation systématique des premières secondes de chaque enregistrement.

Le développement de modèles exploitant ces données est également entravé par une documentation Android incomplète ou imprécise. Il est souvent difficile de connaître avec certitude la signification exacte des variables, les unités de mesure utilisées, ou encore les différences de comportement selon les modèles de montre et les versions d'Android. Cette opacité complique fortement l'interprétation des données, rendant la conception de modèles robustes plus incertaine.

Un autre obstacle majeur concerne la désynchronisation entre les capteurs. Si des travaux comme TapSkin[7] se fondent sur une localisation de pic pour recaler les flux inertiels et acoustiques

en conditions contrôlées, ils supposent un horodatage cohérent entre capteurs. En revanche les capteurs des montres que nous utilisons fonctionnent de manière asynchrone, de façon indépendante et avec des fréquences différentes pour chaque capteur. Cela complique fortement l'alignement temporel des flux et rend les analyses multi-capteurs difficiles, notamment pour les gestes rapides ou composés. Certaines fenêtres d'analyse peuvent même être invalidées lorsqu'un ou plusieurs capteurs restent inactifs.

S'ajoute à cela une interrogation fondamentale qui est de savoir si tous les gestes sont réellement capturables par les capteurs embarqués. Dans l'étude Serendipity [18], les auteurs rapportent un F1-score moyen de 0,87 pour cinq gestes fins (pincer, tapoter, frotter, presser, agiter), mais uniquement dans un protocole de laboratoire où postures et orientations sont fixées, sans aucun matériel externe. À l'inverse, notre protocole inclut des gestes peu variés, parfois très similaires, mais naturels sans capteur complémentaire ni déclencheurs d'activation, ce qui augmente les faux positifs et diminue la précision. Ce questionnement est renforcé par les résultats de l'étude de Yang et al. [16], qui utilisent des capteurs EMG pour démontrer la faisabilité de la reconnaissance de postures de la main en laboratoire, mais sans fournir de métriques de performance chiffrées. Toutefois, leur système repose sur un bracelet externe (MYO) et une configuration de laboratoire.

Malgré ces questionnements et ces difficultés, notre travail apporte une contribution méthodologique concrète avec l'application de collecte développée qui constitue en soi un livrable scientifique réutilisable. Cette application, conçue pour Android et WearOS, permet la collecte de données multi-capteurs avec annotation en temps réel. Aucun des travaux cités (Serendipity, TapSkin, BiTipText, etc.) ne propose un outil logiciel libre, modulaire et compatible avec des montres commerciales sans matériel externe. Cette solution pourra ainsi être exploitée dans d'autres projets de recherche sur l'interaction gestuelle, la rééducation, le suivi moteur ou la santé numérique, sans dépendre d'infrastructures coûteuses ou complexes.

Même si d'autre solution existe comme Sensor Loger [35], qui offre une interface quasi complète et une compatibilité étendue avec une large gamme de capteurs (accéléromètre,

gyroscope, GPS, microphone, capteurs environnementaux, etc.), certaines de ses fonctionnalités avancées restent payantes. De manière générale, ces outils présentent des limites lorsqu'il s'agit de mettre en œuvre des protocoles expérimentaux complexes, comme ceux requis dans les études de reconnaissance gestuelle.

En particulier, aucune de ces solutions ne propose nativement un mode scénario embarqué permettant de guider dynamiquement un participant à travers une séquence structurée de gestes, avec annotation automatique, gestion précise du minutage et contrôle contextuel du déroulement. Or, ce type de fonctionnalité est essentiel pour garantir la qualité des données collectées et la rigueur de leur étiquetage, notamment dans des contextes semi-naturels, où des erreurs d'exécution ou d'annotation peuvent introduire une forte variabilité.

Notre application se démarque sur ce point, en intégrant ce mode scénario directement dans l'interface de collecte, tout en assurant une compatibilité native avec les appareils Android et les montres Android WearOS. De plus, elle dispose d'un mécanisme de détection des capteurs embarqués au lieu de se contenter de vérifier la disponibilité générique d'un type de capteur (comme Sensor.TYPE_*), elle interroge dynamiquement la liste exacte des capteurs physiquement présents sur l'appareil via l'API SensorManager.getSensorList(). Cette approche permet de démarrer uniquement les capteurs réellement installés, y compris ceux qui ne sont pas officiellement déclarés par le fabricant, tout en évitant les erreurs de démarrage sur des capteurs absents. Elle garantit ainsi une collecte plus fiable, cohérente avec la configuration matérielle réelle de chaque montre connectée. Notre application permet également la collecte simultanée de données à partir de plusieurs montres, une fonctionnalité absente des autres solutions disponibles à ce jour.

Pour améliorer la reconnaissance de nos gestes, plusieurs perspectives peuvent être envisagées. Il serait notamment pertinent de combiner notre approche actuelle avec des modèles d'apprentissage profond, tels que les réseaux de neurones. Toutefois, ces approches exigent plus de données.

Pour finir, notons que notre étude met en lumière les limites des approches en reconnaissance des gestes fondées sur les capteurs embarqués, mais aussi leur potentiel lorsqu'elles sont accompagnées d'une ingénierie logicielle permettant la collecte des données issue de ces capteurs. Elle propose des pistes concrètes pour adapter les futures expérimentations à la nature hétérogène, bruitée et instable des données de ces derniers.

CONCLUSION

Bien que notre étude n'ait pas permis d'atteindre une performance de reconnaissance des gestes périmontres attendues, elle offre des contributions tangibles et durables pour la communauté de la recherche et du développement. En effet, nous avons conçu et publié une application Android/Wear OS, libre et modulaire, qui combine la détection dynamique des capteurs embarqués sur la montre et le téléphone, l'exécution séquencée d'un scénario de gestes, l'enregistrement simultané des flux inertiels et l'annotation instantanée. Cette application, grâce à son mode « scénario embarqué » et à son absence de dépendance à tout matériel externe, constitue une preuve de concept opérationnelle et ouvre la voie à de nombreuses réutilisations, dans les domaines de la réadaptation physique et des interactions humain-machine.

De plus, notre retour d'expérience met en lumière les contraintes matérielles et logicielles souvent négligées sur Android Wear OS comme la variabilité et le manque de contrôle de la fréquence d'échantillonnage, absence d'un alignement temporel natif entre capteurs et surcharge logicielle liée aux optimisations énergétiques. En documentant ces limitations, nous décrivons les conditions de collecte avec les capteurs Android, comme une fréquence d'échantillonnage non fixe, un horodatage désynchronisé et une approche de prétraitement de ses capteurs. Cette documentation constitut un socle pour toute étude future souhaitant exploiter de manière fiable les capteurs grand public embarqué dans les dispositifs fonctionnant sous Android ou WearOS.

Nous ouvrons également une nouvelle voie en explorant des capteurs jusqu'ici peu exploités dans la reconnaissance gestuelle, telle que les capteurs PPG, SpO_2 ou environnementaux, désormais intégrés de série dans de nombreuses montres Android Wear OS. Ces capteurs intégrés imposent un traitement spécifique puisque leurs fréquences d'échantillonnage peuvent varier et ne sont pas alignées entre elles. Par ailleurs, les mécanismes d'économie d'énergie d'Android peuvent provoquer des interruptions ou des dérives temporelles. L'exploitation de ces signaux hétérogènes, en développant des pipelines capables de compenser les vides créés par les fréquences

d’échantillonnage non contrôlé constitue une piste prometteuse pour enrichir la robustesse et la précision des systèmes de reconnaissance gestuelle.

En définitive, même si nos performances de classification restent perfectibles en raison d’un nombre de participants restreint, d’une palette de gestes limitée et d’une dépendance à un seul type de montre, ce mémoire livre un outil opérationnel, un état des lieux critique des défis techniques d’Android Wear OS et un cahier de route pour conduire la reconnaissance gestuelle vers des systèmes véritablement autonomes et fiables en conditions réelles.

LISTE DE RÉFÉRENCES

[1] « Global: smartwatches number of users 2019-2028 », Statista. Consulté le: 29 mars 2024. [En ligne]. Disponible sur: https://www-statista-com.sbiproxy.uqac.ca/forecasts/1314339/worldwide-users-of-smartwatches

[2] J. Chun, A. Dey, K. Lee, et S. Kim, « A qualitative study of smartwatch usage and its usability », *Hum. Factors Ergon. Manuf. Serv. Ind.*, vol. 28, nº 4, p. 186-199, 2018, doi: 10.1002/hfm.20733.

[3] M. A. Alshamari et M. M. Althobaiti, « Usability Evaluation of Wearable Smartwatches Using Customized Heuristics and System Usability Scale Score », *Future Internet*, vol. 16, nº 6, p. 204, juin 2024, doi: 10.3390/fi16060204.

[4] J. Lai, L. Zhou, K. Wang, et D. Zhang, « Robust Text Input for Smartwatches: Compensating for Imprecise Tapping and Swiping », *Int. J. Human–Computer Interact.*, p. 1-13, mai 2025, doi: 10.1080/10447318.2025.2490709.

[5] J. Zhou, Y. Zhang, G. Laput, et C. Harrison, « AuraSense: Enabling Expressive Around-Smartwatch Interactions with Electric Field Sensing », in *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, Tokyo Japan: ACM, oct. 2016, p. 81-86. doi: 10.1145/2984511.2984568.

[6] R. Xiao, T. Cao, N. Guo, J. Zhuo, Y. Zhang, et C. Harrison, « LumiWatch: On-Arm Projected Graphics and Touch Input », in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, Montreal QC Canada: ACM, avr. 2018, p. 1-11. doi: 10.1145/3173574.3173669.

[7] C. Zhang *et al.*, « TapSkin: Recognizing On-Skin Input for Smartwatches », in *Proceedings of the 2016 ACM International Conference on Interactive Surfaces and Spaces*, Niagara Falls Ontario Canada: ACM, nov. 2016, p. 13-22. doi: 10.1145/2992154.2992187.

[8] S. Oney, C. Harrison, A. Ogan, et J. Wiese, « ZoomBoard: a diminutive qwerty soft keyboard using iterative zooming for ultra-small devices », in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Paris France: ACM, avr. 2013, p. 2799-2802. doi: 10.1145/2470654.2481387.

[9] M. Malu, P. Chundury, et L. Findlater, « Motor Accessibility of Smartwatch Touch and Bezel Input », in *The 21st International ACM SIGACCESS Conference on Computers and Accessibility*, Pittsburgh PA USA: ACM, oct. 2019, p. 563-565. doi: 10.1145/3308561.3354638.

[10] A. Neshati, B. Rey, A. S. Mohommed Faleel, S. Bardot, C. Latulipe, et P. Irani, « BezelGlide: Interacting with Graphs on Smartwatches with Minimal Screen Occlusion », in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, in CHI '21. New York, NY, USA: Association for Computing Machinery, mai 2021, p. 1-13. doi: 10.1145/3411764.3445201.

[11] D. Vogel et P. Baudisch, « Shift: a technique for operating pen-based interfaces using touch », in *Proceedings of the SIGCHI Conference on Human Factors in Computing*

Systems, San Jose California USA: ACM, avr. 2007, p. 657-666. doi: 10.1145/1240624.1240727.

[12] H. Gil, D. Lee, S. Im, et I. Oakley, « TriTap: Identifying Finger Touches on Smartwatches », in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, Denver Colorado USA: ACM, mai 2017, p. 3879-3890. doi: 10.1145/3025453.3025561.

[13] A. Goguey, M. Nancel, G. Casiez, et D. Vogel, « The Performance and Preference of Different Fingers and Chords for Pointing, Dragging, and Object Transformation », in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, San Jose California USA: ACM, mai 2016, p. 4250-4261. doi: 10.1145/2858036.2858194.

[14] A. Gupta et R. Balakrishnan, « DualKey: Miniature Screen Text Entry via Finger Identification », in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, San Jose California USA: ACM, mai 2016, p. 59-70. doi: 10.1145/2858036.2858052.

[15] M. Funk, A. Sahami, N. Henze, et A. Schmidt, « Using a touch-sensitive wristband for text entry on smart watches », in *CHI '14 Extended Abstracts on Human Factors in Computing Systems*, Toronto Ontario Canada: ACM, avr. 2014, p. 2305-2310. doi: 10.1145/2559206.2581143.

[16] Y. Yang, S. Chae, J. Shim, et T.-D. Han, « EMG Sensor-based Two-Hand Smart Watch Interaction », in *Adjunct Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, Daegu Kyungpook Republic of Korea: ACM, nov. 2015, p. 73-74. doi: 10.1145/2815585.2815724.

[17] W.-H. Chen, « Blowatch: Blowable and Hands-free Interaction for Smartwatches », in *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*, Seoul Republic of Korea: ACM, avr. 2015, p. 103-108. doi: 10.1145/2702613.2726961.

[18] H. Wen, J. Ramos Rojas, et A. K. Dey, « EMG », in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, San Jose California USA: ACM, mai 2016, p. 3847-3851. doi: 10.1145/2858036.2858466.

[19] C. Xu, P. H. Pathak, et P. Mohapatra, « Finger-writing with Smartwatch: A Case for Finger and Hand Gesture Recognition using Smartwatch », in *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications*, Santa Fe New Mexico USA: ACM, févr. 2015, p. 9-14. doi: 10.1145/2699343.2699350.

[20] Z. Xu *et al.*, « BiTipText: Bimanual Eyes-Free Text Entry on a Fingertip Keyboard », in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, Honolulu HI USA: ACM, avr. 2020, p. 1-13. doi: 10.1145/3313831.3376306.

[21] Y. Zhang, J. Zhou, G. Laput, et C. Harrison, « SkinTrack: Using the Body as an Electrical Waveguide for Continuous Finger Tracking on the Skin », in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, San Jose California USA: ACM, mai 2016, p. 1491-1503. doi: 10.1145/2858036.2858082.

[22] M. Ogata et M. Imai, « SkinWatch: skin gesture interaction for smart watch », in *Proceedings of the 6th Augmented Human International Conference*, Singapore Singapore: ACM, mars 2015, p. 21-24. doi: 10.1145/2735711.2735830.

[23] G. Laput, R. Xiao, X. « Anthony » Chen, S. E. Hudson, et C. Harrison, « Skin buttons: cheap, small, low-powered and clickable fixed-icon laser projectors », in *Proceedings of the 27th annual ACM symposium on User interface software and technology*, Honolulu Hawaii USA: ACM, oct. 2014, p. 389-394. doi: 10.1145/2642918.2647356.

[24] « Kotlin et Android », Android Developers. Consulté le: 27 janvier 2025. [En ligne]. Disponible sur: https://developer.android.com/kotlin?hl=fr

[25] « Android Jetpack Dev Resources », Android Developers. Consulté le: 27 janvier 2025. [En ligne]. Disponible sur: https://developer.android.com/jetpack

[26] « Sélectionner un type de client | Wear OS », Android Developers. Consulté le: 30 janvier 2025. [En ligne]. Disponible sur: https://developer.android.com/training/wearables/data/client-types?hl=fr

[27] « Choose a client type | Wear OS », Android Developers. Consulté le: 10 juillet 2025. [En ligne]. Disponible sur: https://developer.android.com/training/wearables/data/client-types

[28] « SensorDirectChannel | API reference », Android Developers. Consulté le: 17 novembre 2025. [En ligne]. Disponible sur: https://developer.android.com/reference/android/hardware/SensorDirectChannel

[29] W. Gomaa et M. A. Khamis, « A perspective on human activity recognition from inertial motion data », *Neural Comput. Appl.*, vol. 35, nº 28, p. 20463-20568, oct. 2023, doi: 10.1007/s00521-023-08863-9.

[30] M. Bennasar, B. A. Price, D. Gooch, A. K. Bandara, et B. Nuseibeh, « Significant Features for Human Activity Recognition Using Tri-Axial Accelerometers », *Sensors*, vol. 22, nº 19, p. 7482, oct. 2022, doi: 10.3390/s22197482.

[31] S. Rosati, G. Balestra, et M. Knaflitz, « Comparison of Different Sets of Features for Human Activity Recognition by Wearable Sensors », *Sensors*, vol. 18, nº 12, p. 4189, nov. 2018, doi: 10.3390/s18124189.

[32] C. Cortes et V. Vapnik, « Support-vector networks », *Mach. Learn.*, vol. 20, nº 3, p. 273-297, sept. 1995, doi: 10.1007/bf00994018.

[33] J. Laaksonen et E. Oja, « Classification with learning k-nearest neighbors », in *Proceedings of International Conference on Neural Networks (ICNN'96)*, juin 1996, p. 1480-1483 vol.3. doi: 10.1109/ICNN.1996.549118.

[34] T. Chen et C. Guestrin, « XGBoost: A Scalable Tree Boosting System », in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, août 2016, p. 785-794. doi: 10.1145/2939672.2939785.

[35] « Sensor Logger », Kelvin Choi. Consulté le: 17 juillet 2025. [En ligne]. Disponible sur: https://www.tszheichoi.com/sensorlogger

CERTIFICATION ÉTHIQUE

Ce mémoire a fait l'objet d'une certification éthique auprès du CER-UQAC. Le numéro du certificat est 2025-1891.

ANNEXE I

Nombres de points à additionner	0	1	2	3	4
Couleur naturelle des cheveux	Blond-roux	Blond	Châtain, blond foncé	Brun foncé	noir
Couleur naturelle des yeux	Bleu clair, gris, vert	Bleu, gris, vert	Marron	Marron foncé	Marron-noir
Couleur des parties de la peau non exposées au soleil	Rougeâtre	Très pâle	Pâle avec Nuance de brun	Brun clair	Brun foncé
Taches de rousseur sur les parties de la peau non exposées au soleil	Nombreuses	Quelques-unes	Peu	Rares	Aucune
Conséquences d'une exposition prolongée sans écran solaire	Rougeurs douloureuses, cloques, exfoliation	Cloques suivies d'exfoliation	Coup de soleil parfois suivi d'exfoliation	Rares coups de soleil	Jamais de problème
Aptitude à bronzer	Peu ou pas de bronzage	Bronzage léger	Bronzage moyen	Bronzage facile	Bronzage très rapide
Une exposition d'un jour au soleil provoque un bronzage	Jamais	Rarement	Quelque fois	Souvent	Toujours
Réaction de la peau du visage au soleil	Très sensible	Sensible	Normale	Peu sensible	Jamais de problème
Dernière exposition au soleil ou à une lampe solaire	Plus de 3 mois	2 ou 3 mois	1 ou 2 mois	Moins d'un mois	Moins de 2 semaines
La zone de traitement est parfois exposée au soleil	Jamais	Très rarement	Quelque fois	Souvent	Toujours

Formulaire classification FitzPatrick

Tableau de classification en groupe selon le résultat du formulaire de FitzPatrick

Résultats entre	Prototypes/Groupes
Entre 0 et 7	Phototype I
Entre 8 et 16	Phototype II
Entre 17 et 25	Phototype II
Entre 26 et 30	Phototype IV
Entre 31 et 35	Phototype V
Entre 36 et 40	Phototype VI

ANNEXE II

UQAC
Université du Québec
à Chicoutimi

FORMULAIRE D'INFORMATION ET DE CONSENTEMENT CONCERNANT LA PARTICIPATION À UN PROJET DE RECHERCHE

1 TITRE DU PROJET

Reconnaissance des interactions tactiles autour d'une montre intelligente à partir des capteurs intégrés.

2 RESPONSABLE(S) DU PROJET DE RECHERCHE

2.1 Responsable

ETOU Koffi Deladem Moïse, Étudiant en Maitrise Informatique profil recherche au Département d'Informatique et de Mathématique à l'Université du Québec à Chicoutimi

2.2 Direction de recherche

Pascal Fortin, Professeur au Département d'Informatique et de Mathématique, Université du Québec à Chicoutimi

3 FINANCEMENT

Ce projet n'est pas financé.

4 PRÉAMBULE

Nous sollicitons votre participation à un projet de recherche. Cependant, avant d'accepter de participer à ce projet et de signer ce formulaire d'information et de consentement, veuillez prendre le temps de lire, de comprendre et de considérer attentivement les renseignements qui suivent.

Ce formulaire peut contenir des mots que vous ne comprenez pas. Nous vous invitons à poser toutes les questions que vous jugerez utiles au chercheur responsable du projet ou aux autres membres du personnel affecté au projet de recherche et à leur demander de vous expliquer tout mot ou renseignement qui n'est pas clair.

5 DESCRIPTION DU PROJET DE RECHERCHE, OBJECTIFS ET DÉROULEMENT

5.1 Description du projet de recherche

Les montres intelligentes, plus que de simples accessoires de mode, incarnent la convergence entre la technologie portable, les besoins de suivi de santé et de productivité. Leur format de petite taille, bien qu'esthétique, impose des limites quant à leurs facilités d'usage. C'est le cas par exemple des tâches de

saisie de texte qui sont complexes à réaliser sur ces dispositifs à cause de l'obstruction du doigt sur l'écran. En réponse à ces défis, de nombreux chercheurs se sont penchés sur la conception de nouvelles manières d'interagir avec nos petits appareils de manière plus efficace.

Ce projet explore la faisabilité de détecter des entrées tactiles (ex : toucher, glisser, pincer) effectuées autour d'une montre intelligente (sur les bras, poignet et main) afin d'étendre les possibilités d'interaction avec ce type d'appareils. Nous proposons également de mener cette exploration en ne se basant uniquement que sur les capteurs déjà inclus dans les montres intelligentes commerciales (ex : capteurs de mouvement, pression, orientation, rythme cardiaque).

5.2 Objectif(s) spécifique(s)

Ce projet vise à étendre les capacités d'interaction des dispositifs à petit écran en exploitant l'espace disponible sur le bras et autour de l'appareil. Pour atteindre cet objectif, nous collecterons des données en laboratoire afin d'entraîner des modèles d'apprentissage automatique capables de prédire les gestes de l'utilisateur autour de la montre intelligente et sur la peau (tapotements, balayages, zooms, etc.).

5.3 Déroulement

Lors de votre arrivée au laboratoire, un expérimentateur prendra le temps de lire avec vous le présent formulaire d'information et de consentement et de procédé à l'obtention de votre consentement si vous acceptez de participer à l'étude. Lors de la collecte de données, il vous sera demandé d'effectuer une série de gestes sur et/ou autour d'une montre intelligente, portée sur le bras sur lequel vous porteriez normalement une montre. Un téléphone intelligent sera disposé devant vous pendant la collecte. Ce dernier vous présentera une image du geste à effectuer, une description textuelle détaillée de celui-ci ainsi que le moment où vous devrez effectuer l'action demandée.

Pour assurer un nombre suffisant de points de données, il pourrait vous être demandé de répéter un même geste à plusieurs reprises. Il est également possible que l'expérimentateur vous demande lui-même d'exécuter un ou plusieurs gestes (sans l'utilisation de l'application). La durée maximale de collecte sera de 1H30min.

Il est important pour l'équipe de recherche que les technologies créées soient inclusives et fonctionnent pour tous. Étant donné que certains des capteurs utilisés sont reconnus pour être influencés par la couleur et le type de peau du porteur de la montre, un court questionnaire vous sera administré pour mieux comprendre les caractéristiques de votre peau et en tenir compte dans l'élaboration des modèles d'apprentissage machine et d'intelligence artificielle. Remplir le questionnaire devrait prendre moins de 5 minutes.

À partir des données collectées, des modèles d'apprentissage machine et d'intelligence artificielle seront entraînés afin de faire la reconnaissance des différents gestes.

6 AVANTAGES, RISQUES ET/OU INCONVÉNIENTS ASSOCIÉS AU PROJET DE RECHERCHE

La recherche que nous menons ne présente aucun risque ou de désavantage prévisible pour les participants, à l'exception du temps nécessaire pour y participer. Les montres portées par les participants sont les mêmes que celles vendues dans le commerce, sans aucune modification. Notez également que votre participation à ce projet ne garantit aucun bénéfice direct, mais offre une opportunité intéressante de contribuer aux découvertes et aux avancées dans le domaine de l'interaction homme-machine.

Les données collectées, l'application de collecte développée, et les résultats obtenus contribueront de manière significative à l'avancement des connaissances sur les interactions avec les dispositifs à petit écran et à l'amélioration des interfaces utilisateur.

7 CONFIDENTIALITÉ, DIFFUSION ET CONSERVATION

7.1 Confidentialité

Les courriels échangés lors de la planification de la rencontre pourraient permettre votre identification. Pour assurer votre confidentialité, tous les courriels seront détruits des boîtes courriel UQAC de l'équipe de recherche dès qu'ils ne seront plus nécessaire à la planification de votre participation.

L'ensemble des données collectées à partir des capteurs de la montre intelligente et du questionnaire seront associés à un identifiant numérique unique pour chaque participant. Cependant il n'existe aucun moyen de faire le lien entre cet identifiant et votre identité.

La confidentialité des données recueillies dans le cadre de ce projet de recherche sera assurée conformément aux lois et règlements applicables dans la province de Québec et aux règlements et politiques de l'Université du Québec à Chicoutimi.

7.2 Diffusion

Avec votre accord, les données recueillies seront mises à la disposition du public sur le dépôt institutionnel Borealis. Les données serviront principalement à permettre à d'autres chercheurs de pouvoir répéter nos travaux, améliorant la crédibilité et robustesse de nos résultats.

Les résultats seront diffusés notamment, dans des articles de conférences et de journaux avec des comités de révision par les paires ainsi que dans des mémoires de maîtrise ou des thèses de doctorat des étudiants-chercheurs. Les données étant complètement anonymisées, aucune information permettant votre identification ne sera diffusée.

7.3 Conservation

Les questionnaires papier seront détruits dès que les données en seront extraites. Les données informatisées seront entreposées sur un support physique (ex : clé USB, disque dur externe) dans le bureau du Prof. Pascal Fortin, ainsi que, avec votre accord, sur le dépôt institutionnel Borealis sous Licence CC BY 4.0. Cette licence confère à toute personne le droit de télécharger, utiliser, partager et d'adapter librement ces données à condition de les attribuer correctement et de ne pas ajouter de restrictions supplémentaires.

Les formulaires de consentement signés seront conservés dans le bureau du Prof. Pascal Fortin à l'UQAC, verrouillé à clé.

Les données de recherche et formulaires de consentement seront conservés pendant une période pendant plus de 7 ans, ou jusqu'à ce qu'elles deviennent obsolètes ou inutilisables. Ils seront ensuite détruits conformément aux pratiques institutionnelles en vigueur.

8 PARTICIPATION VOLONTAIRE ET DROIT DE RETRAIT

Votre participation à ce projet de recherche est entièrement volontaire. Vous avez donc la liberté de refuser de participer ou de vous retirer à tout moment, sans avoir à fournir de raisons. Il vous suffit d'informer le chercheur responsable du projet de votre décision, sans subir de préjudices. Cependant, en raison de l'anonymat des données, celles-ci ne pourront être retirées une fois collectées, car l'équipe de recherche ne dispose d'aucun code permettant de vous identifier.

9 INDEMNITÉ COMPENSATOIRE

Chaque participant recevra une compensation de 10 $ CAD remise en argent comptant. En cas de retrait au cours de collecte, les participants resteront admissibles à la valeur totale de la compensation.

10 PERSONNES-RESSOURCES

Si vous avez des questions concernant le projet de recherche ou si vous éprouvez un problème que vous croyez relié à votre participation au projet de recherche, vous pouvez communiquer avec le responsable du projet aux coordonnées suivantes :

- Koffi Deladem Moïse Etou, étudiant en Maitrise au Département d'Informatique et de Mathématique à l'Université du Québec à Chicoutimi, par courriel : kdetou@etu.uqac.ca

- Pascal E. Fortin, Professeur, Département d'informatique et de mathématique, Université du Québec à Chicoutimi, par téléphone (418) 545-5011 poste 2636 ou par courriel pascal1_fortin@uqac.ca

Pour toute question d'ordre éthique concernant votre participation à ce projet de recherche, vous pouvez communiquer avec le Comité d'éthique de la recherche (par téléphone au 418-545-5011 poste 4704 (ligne sans frais : 1-800-463-9880 poste 4704) ou par courriel à l'adresse cer@uqac.ca.

11 CONSENTEMENT DU PARTICIPANT

Dans le cadre du projet intitulé ***« Reconnaissance des interactions tactiles autour d'une montre intelligente à partir des capteurs intégrés »*** j'ai pris connaissance du formulaire d'information et de consentement et je comprends suffisamment bien le projet pour que mon consentement soit éclairé. Je suis satisfait des réponses à mes questions et du temps que j'ai eu pour prendre ma décision. Je consens donc à participer à ce projet de recherche aux conditions qui y sont énoncées. Je comprends que je suis libre d'accepter de participer et que je pourrai me retirer en tout temps de la recherche si je le désire, sans aucun préjudice ni justification de ma part. Une copie signée et datée du présent formulaire d'information et de consentement m'a été remise.

☐ Je consens à ce que les données anonymisées recueillies soient rendues accessibles via le dépôt institutionnel Borealis. Celles-ci pourraient être utilisées pour d'autres projets de recherche ayant reçu au préalable une approbation éthique.

Nom et signature du participant Date

Signature de la personne qui a obtenu le consentement.

J'ai expliqué au participant à la recherche les termes du présent formulaire d'information et de consentement et j'ai répondu aux questions qu'il m'a posées.

Nom et signature de la personne qui obtient le consentement Date

Signature et engagement du chercheur responsable du projet

Je certifie avoir moi-même, ou un membre autorisé de l'équipe de recherche, expliqué au participant les termes du présent formulaire d'information et de consentement, répondu aux questions qu'il a posées et lui avoir clairement indiqué qu'il pouvait à tout moment mettre un terme à sa participation, et ce, sans préjudice. Je m'engage, avec l'équipe de recherche, à respecter ce qui a été convenu au formulaire d'information et de consentement et à en remettre une copie signée au participant à cette recherche.

Nom et signature du chercheur responsable du projet de recherche Date