

UQAC

Université du Québec
à Chicoutimi

**RECONNAISSANCE D'ACTIVITÉS DANS UN HABITAT INTELLIGENT À
L'AIDE DE RADARS ULTRA-WIDEBAND**

PAR CAMILLE BERTUGLIA

**MÉMOIRE PRÉSENTÉ À L'UNIVERSITÉ DU QUÉBEC À CHICOUTIMI
COMME EXIGENCE PARTIELLE EN VUE DE L'OBTENTION DU GRADE
DE MAÎTRE ÈS SCIENCES EN INFORMATIQUE**

QUÉBEC, CANADA

© CAMILLE BERTUGLIA, 2020

RÉSUMÉ

Le vieillissement démographique de la population humaine et la prédominance de ce phénomène dans les pays occidentaux nous pousse à repenser notre système de santé et de prise en charge des personnes âgées. Les nouvelles technologies comme l'Internet Of Things (IOT) et l'Intelligence Artificielle (IA) se présentent comme une partie de la solution pour relever ce défi sociétal. Les habitats intelligents sont des outils puissants pour assister les personnes âgées à domicile et leur permettre de vivre plus longtemps en autonomie. Ils permettent notamment de faire de la reconnaissance d'activité pour s'assurer de la qualité du train de vie des habitants. Les radars Ultra-WideBand (UWB) représentent une alternative intéressante aux capteurs les plus en vogue dans les habitats intelligents. C'est pourquoi ce mémoire cherche à répondre à la problématique suivante : Comment effectuer de la reconnaissance d'activités dans un habitat intelligent à l'aide de radars ultra-wideband ?

Une introduction présentant le contexte du projet de recherche et définissant ses concepts fondamentaux précédera un état de l'art des techniques d'apprentissage machine, discipline clé de l'intelligence artificielle. Il sera suivi par un autre état de l'art qui se penchera sur les radars UWB et sur les publications scientifiques dont les sujets sont connexes à notre problématique. Ensuite, plusieurs chapitres se concentreront sur la contribution apportée dans le cadre de ce projet de recherche. En premier lieu, les architectures physiques et logicielles du projet seront présentées. Ensuite, les deux jeux de données créés pour l'occasion seront décrits. Enfin, l'application de plusieurs techniques d'apprentissage machine sur l'un des jeux de données créé sera présentée, avec les résultats des tests effectués en conséquence.

ABSTRACT

Population ageing and the strength with which it affects the western world makes us strive to reshape our health care system and the way we take care of the elderly. New technologies such as Internet Of Things (IOT) and Artificial Intelligence (AI) seems to be part of the solution to face this societal challenge. Smart homes are powerful tools to assist seniors in their daily life and help them stay autonomous as long as possible. They are especially helpful for activity recognition, which can help uphold inhabitants quality of life. In this context, Ultra-WideBand (UWB) radars might be a viable option as smart home sensors to perform activity recognition. That's why this memoir aims to answer the following research problem : How to achieve activity recognition in a smart home environment using UWB radars ?

A brief introduction will unfold this research project context and will define key concepts that will be used in further chapters. A state of the art regarding machine learning will then precede another state of the art about UWB radars, in which a brief review of scientific studies linked to our research problem will be made. The next chapters will focus on the contribution of this research project, first by presenting the hardware and software aspect of the project. Then, a brief showcase of the two datasets which were made for the occasion will be made. Last but not least, tests and results from machine learning methods used on one of the two datasets will be exhibited.

TABLE DES MATIÈRES

RÉSUMÉ	ii
ABSTRACT	iii
LISTE DES TABLEAUX	vii
LISTE DES FIGURES	x
LISTE DES ABRÉVIATIONS	xvi
REMERCIEMENTS	xvii
CHAPITRE I – INTRODUCTION	1
1.1 CONTEXTE DE RECHERCHE	1
1.1.1 LES ENJEUX DE LA RECONNAISSANCE D’ACTIVITÉS	1
1.1.2 TECHNOLOGIES UTILISÉES	4
1.2 PROBLÉMATIQUE DE RECHERCHE	11
1.2.1 PROBLÉMATIQUE	11
1.2.2 MOTIVATIONS	13
1.2.3 STRUCTURE DE L’OUVRAGE	15
CHAPITRE II – L’APPRENTISSAGE MACHINE	16
2.1 INTRODUCTION À L’APPRENTISSAGE MACHINE	16
2.2 ARBRES DE DÉCISION ET K PLUS PROCHES VOISINS	18
2.2.1 ARBRES DE DÉCISION	18
2.2.2 APPRENTISSAGE D’ENSEMBLES	22
2.2.3 KNN OU LES K PLUS PROCHES VOISINS	25
2.3 LES RÉSEAUX DE NEURONNES ET L’APPRENTISSAGE PROFOND	27
2.3.1 PRINCIPE DU PERCEPTRON	27
2.3.2 LES RÉSEAUX DE NEURONNES PROFONDS	29
2.3.3 LES RÉSEAUX DE NEURONNES CONVOLUTIFS	34
CHAPITRE III – LES RADARS ULTRA-WIDEBAND	40
3.1 FONDAMENTAUX DES RADARS	40
3.1.1 THÉORIE DES RADARS	40

3.1.2	LES RADARS ULTRA-WIDEBAND	45
3.2	APPLICATIONS EXISTANTES DES RADARS UWB	48
3.2.1	ESTIMATION DU POULS ET DE LA RESPIRATION	48
3.2.2	IMAGERIE	51
3.2.3	DÉTECTION DE POSITION ET D'OCCUPATION	52
3.2.4	DÉTECTION DE CHUTES ET D'ACTIVITÉS	55
CHAPITRE IV – ARCHITECTURE DU PROJET		58
4.1	ARCHITECTURE MATÉRIELLE	58
4.2	ARCHITECTURE LOGICIELLE	61
4.2.1	LOGICIEL D'AQUISITION	61
4.2.2	LOGICIEL CENTRAL	63
4.2.3	LOGICIEL DE VISUALISATION	64
4.3	TESTS DE PRÉCISION DES CAPTEURS	65
4.3.1	TESTS DE PRÉCISION CIRCULAIRE	65
4.3.2	TESTS DE PRÉCISION LINÉAIRE	68
4.3.3	CONCLUSION SUR LES TESTS	72
CHAPITRE V – JEUX DE DONNÉES		73
5.1	OBJECTIFS	73
5.2	PROTOCOLE DE RÉCOLTE	74
5.2.1	L'ENVIRONNEMENT DE RÉCOLTE	74
5.2.2	PROCÉDURE DE RÉCOLTE D'UNE INSTANCE	76
5.3	LE JEU DE DONNÉES DES ACTIVITÉS	76
5.3.1	DÉFINITION DU JEU DE DONNÉES	76
5.3.2	RÉSULTAT	78
5.4	LE JEU DE DONNÉES DES CHUTES	79
5.4.1	DÉFINITION DU JEU DE DONNÉES	79
5.4.2	RÉSULTAT	81
5.5	VISUALISATION DES DONNÉES, TEMPS RAPIDE ET TEMPS LENT	81
5.5.1	TEMPS RAPIDE	81

5.5.2	TEMPS LENT	82
CHAPITRE VI – CONTRIBUTION		84
6.1	JEUX DE DONNÉES À SIMPLE FENÊTRE	87
6.1.1	INTRODUCTION AUX JEUX DE DONNÉES	87
6.1.2	ARBRES DE DÉCISION	88
6.1.3	FORÊT ALÉATOIRE	97
6.1.4	ADABOOST	104
6.1.5	K PLUS PROCHES VOISINS	110
6.1.6	ANALYSE COMPARATIVE	115
6.2	JEUX DE DONNÉES À DOUBLE FENÊTRES	116
6.2.1	INTRODUCTION AUX JEUX DE DONNÉES	116
6.2.2	ARBRES DE DÉCISIONS	118
6.2.3	FORÊT ALÉATOIRE	119
6.2.4	ADABOOST	122
6.2.5	K PLUS PROCHES VOISINS	123
6.2.6	ANALYSE COMPARATIVE	125
CONCLUSION		127
BIBLIOGRAPHIE		129
ANNEXE A – RÉSULTATS DES TESTS DE PRÉCISION CIRCULAIRE		137
ANNEXE B – RÉSULTATS DES TESTS DE PRÉCISION LINÉAIRE .		143
ANNEXE C – CERTIFICAT D’ÉTHIQUE POUR LE JEU DE DON- NÉES DES ACTIVITÉS		146
ANNEXE D – EXTRAIT DES DONNÉES RETOURNÉ PAR UN RA- DAR UWB		147
ANNEXE E – CERTIFICAT D’ÉTHIQUE POUR LE JEU DE DON- NÉES DES CHUTES		148

LISTE DES TABLEAUX

TABLEAU 6.1 :	TABLEAU REPRÉSENTANT L' <i>ACCURACY</i> ET LE <i>COHEN KAPPA</i> SELON LA TAILLE DES FENÊTRES TEMPORELLES (TIME WINDOW) ET LE CHEVAUCHEMENT (OVERLAP) DES JEUX DE DONNÉES. ALGORITHME CART.	91
TABLEAU 6.2 :	TABLEAU REPRÉSENTANT LES INDICES D'ÉVALUATION <i>ACCURACY</i> ET <i>COHEN KAPPA</i> SELON LA VALEUR DES PARAMÈTRES DE TAILLE MINIMALE DE L'ÉCHANTILLON POUR UNE SÉPARATION ET POUR UNE FEUILLE. JEU DE DONNÉES "20_0.9"..	94
TABLEAU 6.3 :	TABLEAU REPRÉSENTANT LES INDICES D'ÉVALUATION <i>ACCURACY</i> ET <i>COHEN KAPPA</i> SELON LES FONCTIONS LIMITANT LE NOMBRE D'ATTRIBUTS À ÉVALUER À CHAQUE SÉPARATION. JEU DE DONNÉES "20_0.9".	94
TABLEAU 6.4 :	TABLEAU COMPARATIF DE TROIS CONFIGURATIONS DE L'ALGORITHME CART SUR LES 6 VARIATIONS DE PRÉ-TRAITEMENT DU JEU DE DONNÉES..	96
TABLEAU 6.5 :	TABLEAU REPRÉSENTANT LES INDICES D'ÉVALUATION <i>ACCURACY</i> ET <i>COHEN KAPPA</i> EN FONCTION DE LA TAILLE DES FENÊTRES TEMPORELLES (TIME WINDOW) ET DU CHEVAUCHEMENT (OVERLAP) DES JEUX DE DONNÉES. ALGORITHME RANDOM FOREST.	99
TABLEAU 6.6 :	TABLEAU REPRÉSENTANT LES INDICES D'ÉVALUATION <i>ACCURACY</i> ET <i>COHEN KAPPA</i> SELON LES FONCTIONS RESTREIGNANT LE NOMBRE D'ATTRIBUTS À ÉVALUER À CHAQUE SÉPARATION. TESTS SUR LES JEUX DE DONNÉES "5_0.9" ET "15_0.9".	102
TABLEAU 6.7 :	TABLEAU REPRÉSENTANT LES INDICES D'ÉVALUATION <i>ACCURACY</i> ET <i>COHEN KAPPA</i> SELON LA TAILLE DES FENÊTRES TEMPORELLES (TIME WINDOW) ET LE CHEVAUCHEMENT (OVERLAP) DES JEUX DE DONNÉES. ALGORITHME ADABOOST.	106
TABLEAU 6.8 :	TABLEAU COMPARATIF DE TROIS CONFIGURATIONS DE L'ALGORITHME ADABOOST SUR LES 6 VARIATIONS DE PRÉ-TRAITEMENT DU JEU DE DONNÉES.	110

TABLEAU 6.9 :	TABLEAU REPRÉSENTANT LES INDICES D'ÉVALUATION <i>ACCURACY</i> ET <i>COHEN KAPPA</i> SELON LA TAILLE DES FENÊTRES TEMPORELLES (TIME WINDOW) ET LE CHEVAUCHEMENT (OVERLAP) DES JEUX DE DONNÉES. ALGORITHME KNN..	112
TABLEAU 6.10 :	TABLEAU COMPARATIF DE TROIS CONFIGURATIONS DE L'ALGORITHME KNN SUR LES 6 VARIATIONS DE PRÉ-TRAITEMENT DU JEU DE DONNÉES..	115
TABLEAU 6.11 :	TABLEAU COMPARATIF DES QUATRES MÉTHODES D'APPRENTISSAGE MACHINE TESTÉES SUR 3 DES 6 VARIATIONS DE PRÉ-TRAITEMENT DU JEU DE DONNÉES.	116
TABLEAU 6.12 :	TABLEAU REPRÉSENTANT PLUSIEURS INDICES DE PERFORMANCE SELON LA TAILLE DES FENÊTRES TEMPORELLES (TIME WINDOW) ET LE CHEVAUCHEMENT (OVERLAP) DES JEUX DE DONNÉES. ALGORITHME CART.	119
TABLEAU 6.13 :	TABLEAU REPRÉSENTANT PLUSIEURS INDICES DE PERFORMANCE SELON LA TAILLE DES FENÊTRES TEMPORELLES (TIME WINDOW) ET LE CHEVAUCHEMENT (OVERLAP) DES JEUX DE DONNÉES. ALGORITHME RANDOM FOREST.	121
TABLEAU 6.14 :	TABLEAU REPRÉSENTANT PLUSIEURS INDICES DE PERFORMANCE SELON LA TAILLE DES FENÊTRES TEMPORELLES (TIME WINDOW) ET LE CHEVAUCHEMENT (OVERLAP) DES JEUX DE DONNÉES. ALGORITHME ADABOOST..	123
TABLEAU 6.15 :	TABLEAU REPRÉSENTANT PLUSIEURS INDICES DE PERFORMANCE SELON LA TAILLE DES FENÊTRES TEMPORELLES (TIME WINDOW) ET LE CHEVAUCHEMENT (OVERLAP) DES JEUX DE DONNÉES. ALGORITHME KNN.	124
TABLEAU 6.16 :	TABLEAU COMPARATIF DES QUATRES MÉTHODES D'APPRENTISSAGE MACHINE TESTÉES SUR LES 6 NOUVELLES VARIATIONS DE PRÉ-TRAITEMENT DU JEU DE DONNÉES..	126

TABLEAU 6.17 : TABLEAU COMPARATIF FINAL REGROUPANT LES MEILLEURES
PERFORMANCES DE CHAQUE ALGORITHME TESTÉ
DANS CE CHAPITRE. 126

LISTE DES FIGURES

FIGURE 1.1 – SCHÉMA DE LA CATÉGORISATION DES DIFFÉRENTS CAPTEURS PERTINENTS POUR LA RECONNAISSANCE D'ACTIVITÉS	6
FIGURE 1.2 – LE PLAN VU DU DESSUS DE L'HABITAT INTELLIGENT DU LIARA	14
FIGURE 2.1 – EXEMPLE D'ARBRE DE DÉCISION ET ILLUSTRATION DE LA MÉTAPHORE FILÉE.	19
FIGURE 2.2 – EXEMPLE DE LA MÉTHODE DES K PLUS PROCHE VOISINS AVEC $K = 5$	26
FIGURE 2.3 – SCHÉMA DU PERCEPTRON.	28
FIGURE 2.4 – SCHÉMA D'UN PERCEPTRON MULTI-COUCHES À UNE COUCHE CACHÉE POUR UNE CLASSIFICATION BINAIRE	30
FIGURE 2.5 – ILLUSTRATION D'UNE ÉTAPE DE L'OPÉRATION DE CONVOLUTION	35
FIGURE 2.6 – APPLICATION D'UN FILTRE DE DÉTECTION DE BORD COMME NOYAU DE CONVOLUTION SUR UNE IMAGE NUMÉRIQUE	35
FIGURE 2.7 – ILLUSTRATION D'UNE ÉTAPE DE L'OPÉRATION DE MISE EN COMMUN "MAX POOLING"	36
FIGURE 2.8 – ARCHITECTURE DU RÉSEAU DE NEURONNES CONVOLUTIONNIF ALEXNET	38
FIGURE 2.9 – ARCHITECTURE DU RÉSEAU DE NEURONNES CONVOLUTIONNIF GOOGLNET.	38
FIGURE 2.10 – ARCHITECTURE DU RÉSEAU DE NEURONNES CONVOLUTIONNIF RESNET	39
FIGURE 3.1 – ILLUSTRATION DE L'EFFET DOPPLER	43
FIGURE 4.1 – CHAMP DE DÉTECTION FRONTAL DU RADAR ULTRA-WIDEBAND SCHÉMATISÉ À PARTIR DES INFORMATIONS FOURNIES PAR SA FICHE TECHNIQUE	59
FIGURE 4.2 – ARCHITECTURE MATÉRIELLE DU PROJET	60

FIGURE 4.3 – ARCHITECTURE LOGICIELLE DU PROJET.	61
FIGURE 4.4 – CAPTURE D’ÉCRAN DE L’INTERFACE DE VISUALISATION WEB	64
FIGURE 4.5 – SCHÉMA ILLUSTRANT LES MARQUEURS DES TESTS CIRCULAIRES EFFECTUÉS	66
FIGURE 4.6 – TEST DE PRÉCISION CIRCULAIRE SUR UN DEMI-CERCLE DE 0,5M DE RAYON	67
FIGURE 4.7 – TEST DE PRÉCISION CIRCULAIRE SUR UN DEMI-CERCLE DE 2M DE RAYON	68
FIGURE 4.8 – TEST DE PRÉCISION CIRCULAIRE SUR UN DEMI-CERCLE DE 3,5M DE RAYON	68
FIGURE 4.9 – SCHÉMA ILLUSTRANT LES MARQUEURS DES TESTS LINÉAIRES EFFECTUÉS	69
FIGURE 4.10 – TEST DE PRÉCISION LINÉAIRE SUR L’AXE CENTRAL DE 0,5M À 5M	70
FIGURE 4.11 – TEST DE PRÉCISION LINÉAIRE SUR L’AXE CENTRAL DE 5M À 10M, AVEC LES PARAMÈTRES DE PORTÉE PAR DÉFAUT ÉTENDUS	70
FIGURE 4.12 – TEST DE PRÉCISION LINÉAIRE SUR LA PERPENDICULAIRE DE 0M À 5M, SÉRIE 1	71
FIGURE 4.13 – TEST DE PRÉCISION LINÉAIRE SUR LA PERPENDICULAIRE DE 0M À 5M, SÉRIE 2	72
FIGURE 5.1 – LE PLAN VU DU DESSUS DE L’HABITAT INTELLIGENT DU LIARA AVEC LES POSITIONS DES RADARS ULTRA-WIDEBAND EN ROUGE	75
FIGURE 5.2 – LE PLAN VU DU DESSUS DE L’HABITAT INTELLIGENT DU LIARA AVEC LES POSITIONS DES CHUTES EFFECTUÉES DÉSIGNÉES PAR LES RECTANGLES ROUGES.	80
FIGURE 5.3 – REPRÉSENTATION D’UNE IMAGE RADAR EXTRAITE DE L’INSTANCE D’ACTIVITÉ “BOIRE_0” ET VISUALISATION DE LA NOTION DE TEMPS RAPIDE.	82

FIGURE 5.4 – REPRÉSENTATION D’UNE FENÊTRE TEMPORELLE DE 100 MILLISECONDES (5 IMAGES) ET DES NOTIONS DE TEMPS RAPIDE ET DE TEMPS LENT..	83
FIGURE 6.1 – <i>ACCURACY</i> EN FONCTION DE LA PROFONDEUR MAXIMUM DE L’ARBRE AVEC EN BLEU L’ENTROPIE COMME CRITÈRE D’IMPURETÉ ET EN ROUGE L’INDICE DE GINI. JEU DE DONNÉES “20_0.9”..	92
FIGURE 6.2 – <i>ACCURACY</i> EN FONCTION DE LA PROFONDEUR MAXIMUM DE L’ARBRE AVEC EN BLEU L’ENTROPIE COMME CRITÈRE D’IMPURETÉ ET EN ROUGE L’INDICE DE GINI. JEU DE DONNÉES “15_0.9”..	93
FIGURE 6.3 – <i>ACCURACY</i> EN FONCTION DU SEUIL DE RÉDUCTION MINIMUM D’IMPURETÉ. JEU DE DONNÉES “20_0.9”.	95
FIGURE 6.4 – <i>ACCURACY</i> EN FONCTION DU NOMBRE D’ARBRES CONSTITUANTS LA FORÊT, AVEC EN BLEU LE JEU DE DONNÉES “5_0.9” ET EN ROUGE LE JEU DE DONNÉES “15_0.9”..	100
FIGURE 6.5 – <i>ACCURACY</i> EN FONCTION DE LA PROFONDEUR MAXIMALE AVEC EN BLEU L’ENTROPIE COMME CRITÈRE D’IMPURETÉ ET EN ROUGE L’INDICE DE GINI. JEU DE DONNÉES “5_0.9”..	101
FIGURE 6.6 – <i>ACCURACY</i> EN FONCTION DE LA PROFONDEUR, MAXIMALE AVEC EN BLEU L’ENTROPIE COMME CRITÈRE D’IMPURETÉ ET EN ROUGE L’INDICE DE GINI. JEU DE DONNÉES “15_0.9”.	101
FIGURE 6.7 – <i>ACCURACY</i> EN FONCTION DU SEUIL DE RÉDUCTION MINIMUM D’IMPURETÉ. JEU DE DONNÉES “5_0.9”.	103
FIGURE 6.8 – TABLEAU COMPARATIF DE TROIS CONFIGURATIONS POUR LA MÉTHODE DE FORÊT ALÉATOIRE SUR LES 6 VARIATIONS DE PRÉ-TRAITEMENT DU JEU DE DONNÉES.	104
FIGURE 6.9 – <i>ACCURACY</i> EN FONCTION DU TAUX D’APPRENTISSAGE AVEC EN BLEU 10 ARBRES, EN ROUGE 25 ARBRES, EN VERT 50 ARBRES, EN VIOLET 75 ARBRES ET EN ORANGE 100 ARBRES PAR GROUPE. JEU DE DONNÉES “5_0.9”.	107

FIGURE 6.10 – <i>ACCURACY</i> EN FONCTION DU TAUX D’APPRENTISSAGE AVEC EN BLEU 10 ARBRES, EN ROUGE 25 ARBRES, EN VERT 50 ARBRES, EN VIOLET 75 ARBRES ET EN ORANGE 100 ARBRES PAR GROUPE. JEU DE DONNÉES “15_0.9”. . .	107
FIGURE 6.11 – <i>ACCURACY</i> EN FONCTION DE LA PROFONDEUR MAXIMALE, AVEC EN BLEU L’ENTROPIE COMME CRITÈRE D’IMPURETÉ ET EN ROUGE L’INDICE DE GINI. JEU DE DONNÉES “5_0.9”.	108
FIGURE 6.12 – <i>ACCURACY</i> EN FONCTION DE LA PROFONDEUR MAXIMALE, AVEC EN BLEU L’ENTROPIE COMME CRITÈRE D’IMPURETÉ ET EN ROUGE L’INDICE DE GINI. JEU DE DONNÉES “15_0.9”.	109
FIGURE 6.13 – <i>ACCURACY</i> EN FONCTION DU NOMBRE DE VOISINS, AVEC EN BLEU UNE RÉPARTITION DES POIDS UNIFORME ET EN ROUGE UNE RÉPARTITION DES POIDS SELON LA FONCTION DE DISTANCE. JEU DE DONNÉES “15_0.9”. . .	113
FIGURE 6.14 – <i>ACCURACY</i> EN FONCTION DU NOMBRE DE VOISINS, AVEC EN BLEU UNE RÉPARTITION DES POIDS UNIFORME ET EN ROUGE UNE RÉPARTITION DES POIDS SELON LA FONCTION DE DISTANCE. JEU DE DONNÉES “20_0.9”. . .	113
FIGURE 6.15 – <i>ACCURACY</i> EN FONCTION DE LA VALEUR DE P AVEC LE JEU DE DONNÉES “15_0.8” EN BLEU ET LE JEU DE DONNÉES “20_0.9” EN ROUGE.	114
FIGURE 6.16 – REPRÉSENTATION D’UNE FENÊTRE SECONDAIRE DE 5 SECONDES CENTRÉE SUR UNE FENÊTRE PRINCIPALE DE 15 SECONDES.	117
FIGURE A.1 – TEST DE PRÉCISION CIRCULAIRE SUR UN DEMI-CERCLE DE 0,5M DE RAYON - 01	137
FIGURE A.2 – TEST DE PRÉCISION CIRCULAIRE SUR UN DEMI-CERCLE DE 0,5M DE RAYON - 02	137
FIGURE A.3 – TEST DE PRÉCISION CIRCULAIRE SUR UN DEMI-CERCLE DE 1M DE RAYON - 01	138
FIGURE A.4 – TEST DE PRÉCISION CIRCULAIRE SUR UN DEMI-CERCLE DE 1M DE RAYON - 02	138
FIGURE A.5 – TEST DE PRÉCISION CIRCULAIRE SUR UN DEMI-CERCLE DE 1,5M DE RAYON - 01	139

FIGURE A.6 – TEST DE PRÉCISION CIRCULAIRE SUR UN DEMI-CERCLE DE 1,5M DE RAYON - 02	139
FIGURE A.7 – TEST DE PRÉCISION CIRCULAIRE SUR UN DEMI-CERCLE DE 2M DE RAYON - 01	140
FIGURE A.8 – TEST DE PRÉCISION CIRCULAIRE SUR UN DEMI-CERCLE DE 2M DE RAYON - 02	140
FIGURE A.9 – TEST DE PRÉCISION CIRCULAIRE SUR UN DEMI-CERCLE DE 2.5M DE RAYON - 01	141
FIGURE A.10 –TEST DE PRÉCISION CIRCULAIRE SUR UN DEMI-CERCLE DE 2.5M DE RAYON - 02	141
FIGURE A.11 –TEST DE PRÉCISION CIRCULAIRE SUR UN DEMI-CERCLE DE 3M DE RAYON - 01	141
FIGURE A.12 –TEST DE PRÉCISION CIRCULAIRE SUR UN DEMI-CERCLE DE 3M DE RAYON - 02	142
FIGURE A.13 –TEST DE PRÉCISION CIRCULAIRE SUR UN DEMI-CERCLE DE 3.5M DE RAYON - 01	142
FIGURE A.14 –TEST DE PRÉCISION CIRCULAIRE SUR UN DEMI-CERCLE DE 3.5M DE RAYON - 02	142
FIGURE B.1 – TEST DE PRÉCISION LINÉAIRE SUR L'AXE CENTRAL DE 0,5M À 5M	143
FIGURE B.2 – TEST DE PRÉCISION LINÉAIRE SUR L'AXE CENTRAL DE 0,5M À 6M	143
FIGURE B.3 – TEST DE PRÉCISION LINÉAIRE SUR L'AXE CENTRAL DE 5M À 10M, AVEC LES PARAMÈTRES DE PORTÉE PAR DÉFAUT ÉTENDUS	144
FIGURE B.4 – TEST DE PRÉCISION LINÉAIRE SUR L'AXE CENTRAL DE 5M À 10M, AVEC LES PARAMÈTRES DE PORTÉE PAR DÉFAUT MAINTENUS	144
FIGURE B.5 – TEST DE PRÉCISION LINÉAIRE SUR L'AXE CENTRAL DE 5M À 11M, AVEC LES PARAMÈTRES DE PORTÉE PAR DÉFAUT MAINTENUS	145

FIGURE D.1 – EXTRAIT D’UN FICHIER CONTENANT LES DONNÉES
RETOURNÉES PAR LE RADAR. ICI, L’ENTIÈRETÉ D’UNE
IMAGE D’UN UNIQUE RADAR EST REPRÉSENTÉE. 147

LISTE DES ABRÉVIATIONS

IOT	Internet Of Things
IA	Intelligence Artificielle
AI	Artificial Intelligence
CHSLD	Centre d'Hébergement et de Soins de Longue Durée
LIARA	Laboratoire d'Intelligence Ambiante pour la Reconnaissance d'Activités
UQAC	Université du Québec À Chicoutimi
DIM	Département d'Informatique et de Mathématique
CER	Comité d'Éthique de la Recherche
ETS	École de Technologie Supérieure
INSA	Institut National des Sciences Appliquées
RADAR	RAdio Detection And Ranging
UWB	Ultra-WideBand
KNN	K Nearest Neighbors
SVM	Support Vector Machines
ReLU	Rectified Linear Unit
CNN	Convolutional Neural Network
LSTM	Long Short Term Memory

REMERCIEMENTS

En premier lieu, je souhaite remercier M. Kévin Bouchard, professeur au Département d'Informatique et de Mathématique (DIM) de l'Université du Québec À Chicoutimi (UQAC) et membre actif du Laboratoire d'Intelligence Ambiante pour la Reconnaissance d'Activités (LIARA). Merci à lui pour l'opportunité de poursuivre ce projet de recherche sous sa direction, pour son aide précieuse dans le cadre de ma maîtrise, et pour ses conseils concernant mon évolution dans le monde de la recherche et dans le milieu académique.

J'aimerais également remercier M. Julien Maître, stagiaire post-doctoral au LIARA puis professeur au DIM de l'UQAC, pour son suivi quotidien durant la phase de développement du projet, pour son aide régulière, ainsi que pour son encadrement en tant que co-directeur du projet de recherche.

Merci également à M. Sébastien Gaboury, professeur au DIM de l'UQAC et directeur du LIARA, pour son encadrement en tant que co-directeur du projet et son intérêt constant envers celui-ci.

Mes remerciements à M. Pascal Berthomé, Mme Charlotte Renard et M. Guy Feuillard de l'Institut National des Sciences Appliquées (INSA) Centre-Val de Loire ainsi qu'à M. Bob Ménélas, Mme Suzie Girard et Mme Audrey Fortin de l'UQAC pour le temps qu'ils ont consacré pour me permettre de changer de maîtrise à mon arrivée à Chicoutimi. J'ai ainsi pu rediriger mon parcours vers le monde de la recherche sans compromettre mon diplôme d'ingénieur INSA. Merci à eux pour avoir trouvé une entente entre les deux établissements au sujet de ma double diplomation et pour avoir pérennisé leurs efforts avec la création d'un accord de double diplomation qui sera proposé à tous les futurs étudiants INSAiens souhaitant effectuer une maîtrise en informatique volet recherche à l'UQAC.

Enfin, je voudrais remercier l'entière équipe du LIARA pour leur accueil chaleureux au sein du laboratoire et pour la convivialité qu'ils y apportent au quotidien. Merci tout particulièrement à Cédric pour ses conseils, son aide technique au cours du projet, et son soutien régulier. Merci aussi à Lucas avec qui j'ai eu la chance de travailler brièvement dans le cadre de ce projet.

CHAPITRE I

INTRODUCTION

1.1 CONTEXTE DE RECHERCHE

1.1.1 LES ENJEUX DE LA RECONNAISSANCE D'ACTIVITÉS

LE VIEILLISSEMENT DE LA POPULATION

Bien que sa croissance ait légèrement ralenti ces dernières années, la durée de vie humaine ne cesse d'augmenter à l'échelle mondiale¹. Cette tendance s'accompagne, de manière assez logique, d'un vieillissement global de la population. Ce phénomène est particulièrement accentué dans les zones géographiques où l'espérance de vie est la plus élevée et où le taux de natalité est relativement faible. Ainsi, en Europe, la part de la population âgée de plus de 60 ans représente 25% de la population totale et est amenée à atteindre 35% en 2050 d'après les prévisions de l'Organisation des Nations unies (Department of Economic and Social Affairs, United Nations, 2017). De même, l'Institut de la Statistique du Québec anticipe qu'en 2066, les personnes âgées de 65 ans et plus représenteront 28% de la population du Québec, face à 18% en 2016 (Institut de la Statistique du Québec, 2019).

Le vieillissement s'accompagne souvent d'un accroissement des troubles de la santé pouvant porter atteinte à l'autonomie des individus. En effet, il entraîne la dégra-

1. Augmentation d'un peu plus de 17 ans entre 1950 et 1985, puis d'un peu moins de 10 ans entre 1985 et 2020 (Department of Economic and Social Affairs, United Nations, 2019)

dation des capacités sensorielles (notamment l'ouïe et la vue), l'atrophie des muscles, la fragilisation des os, ainsi que la réduction des capacités cognitives et mémorielles. Pour une partie des personnes âgées, cela rend difficile la complétion de tâches jusque-là aisées, et impossible la réalisation de tâches complexes sur le plan physique ou cognitif.

De plus, l'autonomie de ces personnes est fortement impactée lors de la présence de maladies dégénératives, plus spécifiquement les maladies neuro-dégénératives comme la maladie d'Alzheimer ou la maladie de Parkinson. En effet, les symptômes de ces maladies sont sévères avec, entre autres, la perte de mémoire et la confusion pour la maladie d'Alzheimer; et la lenteur des mouvements, les raideurs musculaires et les tremblements pour la maladie de Parkinson (Organisation Mondiale de la Santé, 2006). Cela pose un problème conséquent pour la réalisation de n'importe quelle tâche mettant en jeu une suite ordonnée et complexe d'actions nécessitant de la mémoire, de la coordination et de la précision. Ainsi, un intervenant extérieur est souvent nécessaire pour aider les personnes atteintes de ces maladies à effectuer les tâches de la vie quotidienne.

GESTION DU VIELLISSEMENT ET SOLUTIONS PROMETTEUSES

Afin de satisfaire ce besoin d'assistance quotidien et pour conserver le bien-être des personnes âgées en perte d'autonomie, les familles de ces dernières et les professionnels de santé s'accordent parfois à les faire domicilier dans des résidences spécialisées. Cependant, la qualité des soins proposés et la qualité de vie des résidents peuvent être insuffisants. Au Québec, les préposés aux bénéficiaires des Centre d'Hébergement et de Soins de Longue Durée (CHSLD) estiment la qualité des soins prodigués assez bonne,

mais dénoncent un épuisement du personnel et un ratio préposés / personnes âgées trop faible (Riendeau, 2006). Cela entraîne des comportements jugés inadéquats envers les personnes vivant en centre d'hébergement et une réduction des interactions sociales entre ces dernières et les préposés. Or, la participation sociale est un facteur important du bien-être des personnes âgées et devrait être davantage pris en compte dans les CHSLD (Gravelle, 2019).

De plus, les résidences spécialisées sont souvent plus onéreuses socialement que l'assistance et les soins à domicile. En effet, hormis pour les cas de dépendance les plus sévères, le coût global pour maintenir une personne en perte d'autonomie chez elle est moins cher que le coût engendré dans le cas d'une domiciliation dans un CHSLD. C'est ce que montre une étude effectuée au Canada à Victoria et Winnipeg par Chappell *et al.* (2004). Une étude plus ancienne montre des résultats similaires pour les personnes atteintes de la maladie d'Alzheimer aux États-Unis d'Amérique (Leon et Moyer, 1999). C'est pourquoi les services et les technologies permettant le maintien à domicile des personnes âgées se présentent comme des solutions attractives. Ils permettent de conserver la qualité de vie et la dignité des individus, tout en réduisant la charge de travail nécessaire aux professionnels de la santé et l'impact budgétaire associé pour les particuliers comme pour le gouvernement provincial.

1.1.2 TECHNOLOGIES UTILISÉES

LES HABITATS INTELLIGENTS

Cette tendance à favoriser le maintien chez soi des personnes en perte d'autonomie se traduit à la fois par une augmentation du personnel dédié au suivi des patients à domicile et par un essor des technologies d'assistance informatisées et de suivi à distance. Dans un contexte où la domotique et l'Internet Of Things (IOT) sont des notions de plus en plus populaires et dont les technologies se développent rapidement, les habitats intelligents se présentent comme une solution adéquate pour répondre au besoin d'assistance et de suivi des personnes en perte d'autonomie (Chan *et al.*, 2009).

Un habitat intelligent est un regroupement de systèmes informatisés ubiquitaires dans une habitation, géré à l'aide d'une intelligence ambiante pour fournir des services adaptés au contexte environnemental et faciliter un contrôle des équipements à distance (Alam *et al.*, 2012). Plus simplement, on peut définir un habitat intelligent comme un habitat équipé à la fois de capteurs pour recueillir des informations sur l'environnement, et de systèmes ubiquitaires recueillant et traitant les informations de ces capteurs . On retrouve parfois la présence d'effecteurs, c'est-à-dire de mécanismes informatisés permettant d'interagir sur l'environnement. Leurs actions sont décidées en conséquence des informations des capteurs et de l'objectif du système. Par exemple, une maison équipée de thermomètres, de chauffages programmables et d'une unité de prise de décision régulant les chauffages en fonction des températures peut constituer un petit environnement intelligent. De nombreux effecteurs possibles peuvent être envisagés, par exemple des ampoules connectées, des enceintes, des écrans, des outils de chauffage,

des robinets, des serrures ou encore des robots. Quant aux capteurs, ils peuvent être regroupés dans différentes catégories selon le type d'informations qu'ils collectent. Ces catégories seront spécifiées par la suite.

LA RECONNAISSANCE D'ACTIVITÉS

Dans le cadre du maintien chez soi des personnes en perte d'autonomie, les habitats intelligents peuvent s'avérer utiles pour effectuer de la reconnaissance d'activités. En effet, l'accumulation de données sur l'environnement via différents capteurs peut, grâce aux techniques de l'intelligence artificielle, permettre d'inférer les activités réalisées par les personnes se trouvant dans l'habitat. Ces informations peuvent être utiles pour s'assurer qu'une personne atteinte de troubles de la mémoire n'oublie pas d'effectuer certaines actions cruciales comme la prise de médicaments, la fermeture des robinets, l'extinction de certains appareils à risques incendiaires, etc (Orpwood *et al.*, 2005). Elles peuvent également permettre de détecter certaines activités critiques comme des chutes ou des arrêts brutaux d'activité pouvant être dus à un malaise ou à un arrêt cardiaque (Planinc et Kampel, 2013). Enfin, l'analyse qualitative de ces activités peut permettre de déceler des modifications comportementales pouvant témoigner de la santé du patient et permettre la détection anticipée de certains symptômes de maladies neuro-dégénératives (Alberdi *et al.*, 2018).

LES DIFFÉRENTS CAPTEURS

Les capteurs influent sur les techniques d'intelligence artificielle pouvant être utilisées selon le type de données qu'ils émettent et selon l'exhaustivité de ces données. Cependant, certains facteurs comme le prix, la difficulté d'installation ou l'impact sur la vie privée sont tout aussi déterminants. C'est pourquoi les capteurs pour la reconnaissance d'activités les plus présents dans la littérature seront présentés dans cette section, classés dans des catégories générales selon la répartition représentée en Figure 1.1.

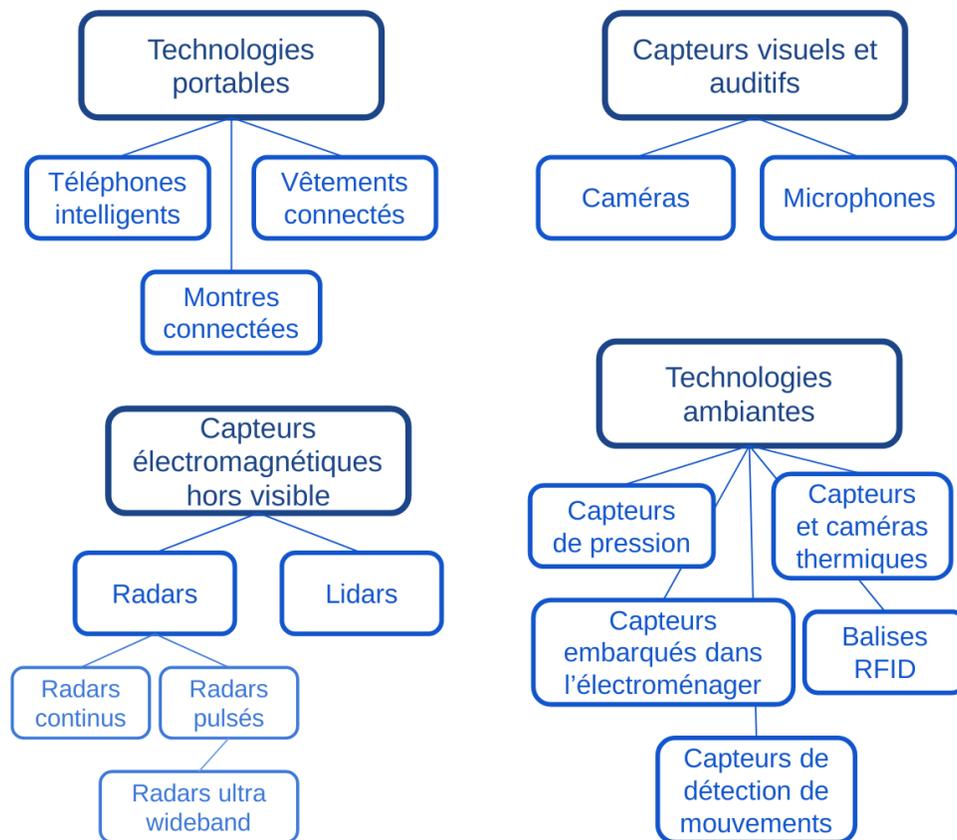


Figure 1.1 : Schéma de la catégorisation des différents capteurs pertinents pour la reconnaissance d'activités

Technologies portables Les technologies portables regroupent les systèmes ubiquitaires et capteurs portés par un utilisateur ou présents dans ses vêtements. Elles ne font pas à proprement parler partie des habitats intelligents, mais elles sont souvent utilisées pour la reconnaissance d'activités, seules ou en synergie avec d'autres capteurs installés dans l'habitat (Hammerla *et al.*, 2016; Thullier *et al.*, 2017; Chapron *et al.*, 2017). Les systèmes technologiques portables les plus présents au quotidien sont les téléphones intelligents, dont la plupart sont équipés d'un grand nombre de capteurs comme des accéléromètres, des gyroscopes, des magnétomètres, des GPS, des microphones, etc (Su *et al.*, 2014). Les montres connectées illustrent bien les capacités de ces systèmes à obtenir des informations pertinentes pour la reconnaissance d'activités, car bon nombre d'entre elles sont utilisées pour la collecte de données sur les performances sportives de l'utilisateur. Les solutions sous forme de montres sont assez populaires dans la littérature et parmi les solutions commerciales, car elles sont ergonomiques et peuvent être portées à même le corps (Bhattacharya et Lane, 2016). Les technologies portables peuvent détecter de manière fiable certaines informations comme le rythme cardiaque, l'activité cérébrale ou les mouvements. Elles ont également la particularité d'être très facile à mettre en place, car elles ne dépendent pas du tout de l'habitat. Enfin, leur coût est souvent nettement inférieur à celui d'un ensemble de capteurs à installer dans un logement.

En revanche, ces technologies sont applicables à l'échelle de l'individu et non de l'habitat, et sont ainsi victimes de défauts majeurs. Le premier d'entre eux est la nécessité d'être porté par l'utilisateur pour en garantir le fonctionnement. Il est difficile d'imposer le port constant d'un appareil portable pour une personne atteinte de troubles

mémoriels qui pourrait l'enlever et oublier de le remettre, ou d'une personne atteinte de démence qui pourrait être amenée à vouloir s'en séparer sans raison rationnelle. Aussi, l'exigence d'un port constant, par exemple pendant la toilette ou le sommeil, peut être inconfortable pour l'utilisateur. De plus, leur bon fonctionnement dépend de leur autonomie énergétique et les plus énergivores nécessiteront un rechargement régulier. Celui-ci, en plus d'être susceptible d'être oublié par les personnes atteintes de perte de mémoire, pose le problème suivant. Soit le système peut être porté lors de son rechargement, auquel cas il force l'utilisateur à être relativement statique, soit il ne peut pas être porté, auquel cas l'utilisateur n'est pas couvert par ses services durant la période de rechargement.

Capteurs visuels et auditifs Les caméras sont des capteurs déjà présents massivement dans notre environnement quotidien, et dont les principes théoriques et technologiques sont maîtrisés. Leur abondance engendre un panel de choix massif et varié. Bien qu'utilisées essentiellement à des fins de sécurité et de reconnaissance d'individus, elles peuvent être employées à reconnaître des mouvements ou des interactions avec certains objets permettant d'inférer des activités (Xia et Aggarwal, 2013). Les microphones sont également omniprésents et peuvent aussi permettre, dans une moindre mesure, de faire de la reconnaissance d'activités. Ils manifestent surtout une synergie intéressante avec d'autres capteurs (Stager *et al.*, 2004; Garcia-Ceja *et al.*, 2018).

En revanche, on note deux problèmes majeurs pour les capteurs visuels et auditifs. Le premier est l'impact du bruit sur la performance des capteurs. Dans le cas des microphones, la notion de bruit renvoyant traditionnellement au bruit sonore, le problème

semble assez évident. Cependant la pollution lumineuse est tout aussi apte à noyer l'information utile des caméras et peut impacter grandement la qualité des informations inférées. De plus, les données récoltées par ces capteurs ont la particularité d'être très facilement interprétables par l'homme. Une série de coordonnées, une série de valeurs de pression ou un ensemble de données renvoyées par des radars semblent abstraites et nécessitent un traitement informatisé pour toute interprétation. En revanche, les vidéos ou les enregistrements audio sont très facilement et immédiatement compréhensibles pour quiconque s'en procurerait l'accès. Pour cette raison, l'atteinte à la vie privée que pourrait occasionner ces capteurs les rend peu attractifs.

Technologies ambiantes La technologie ambiante regroupe tous les capteurs axés sur la détection des changements dans l'habitat plutôt que sur la détection directe de l'individu et de ses mouvements. On y trouve par exemple les capteurs thermiques, les capteurs de pression, les capteurs de détection de mouvement et les capteurs embarqués dans les objets du quotidien (Hevesi *et al.*, 2014; Cheng *et al.*, 2016). Parmi ceux-ci, on peut citer les capteurs de détection d'ouverture de portes, de débit des robinets, d'utilisation de l'équipement électro-ménager, ou les balises RFID (Radio-Frequency Identification) annexées aux objets non connectés (Li *et al.*, 2016; Fortin-Simard *et al.*, 2015). La technologie ambiante a l'avantage de se baser sur l'état de l'environnement plutôt que sur les sujets humains eux-mêmes, et pose donc moins de problèmes liés à la vie privée que les capteurs visuels et auditifs. Un environnement équipé de manière exhaustive propose des prédictions fiables et complètes, car on peut à la fois inférer la position des individus dans l'habitat via la position des capteurs activés, mais également

la nature de leurs activités. L'utilisation prolongée d'un robinet situé dans une cuisine peut assez facilement être liée à un nettoyage de la vaisselle, déduction que l'on peut appuyer si la vaisselle est munie de balises RFID et qu'un changement de position est détecté.

Cependant, l'installation de ces capteurs peut être difficile et coûteuse. En effet, elle est extrêmement dépendante de l'agencement de l'habitat et de son équipement. Ainsi, le nombre et la nature des capteurs à installer doivent être étudiés presque au cas par cas, ce qui rend une solution basée sur un écosystème de capteurs ambiants difficile à mettre en place à grande échelle. La réduction du nombre de capteurs ou de leur variété compense cette limitation, mais réduit le nombre d'éléments contextuels susceptibles d'être analysés et impacte ainsi les performances de la reconnaissance d'activités.

Radars Les radars, pour RADio Detection And Ranging (RADAR), sont des systèmes utilisant les ondes radio, c'est-à-dire le spectre électromagnétique dont la fréquence se situe en dessous des infrarouges, pour détecter la distance ou la vitesse d'un objet (Skolnik, 2001). Ainsi, ils sont insensibles à la pollution lumineuse qui ne handicape que les capteurs propres à la partie visible du spectre électromagnétique. Les radars Ultra-WideBand (UWB), une sous-famille des radars, sont également capables de détecter une présence à travers des surfaces opaques grâce à la distorsion du signal renvoyé par la respiration et le pouls du sujet (Taylor, 1994). Cela les rend particulièrement propices pour des environnements où la présence d'obstacles obstruerait le champ de détection d'autres capteurs comme les caméras. Les données n'étant pas interprétables sans traitements mathématiques, ils permettent de conserver la vie privée des individus présents

dans l'habitat. Pour un habitat d'une surface raisonnable, il n'est pas nécessaire d'utiliser un grand nombre de capteurs. Et bien que leur position générale dans l'appartement ait un impact sur leur performance, leur installation reste assez aisée et généralisable pour un grand nombre d'agencements d'habitats. Enfin, étant présents dans l'appartement, ils ne présentent aucun des inconvénients propres aux systèmes portatifs.

Bien que versatiles, les radars UWB ne sont pas dénués de défauts et ne fournissent pas d'informations permettant d'inférer parfaitement l'activité des habitants d'un logement. Leur portée est limitée et leur précision a tendance à diminuer lorsque l'utilisateur s'éloigne de l'axe central du champ de détection. Certains phénomènes qui seront détaillés par la suite montrent qu'ils ne sont pas immunisés aux interférences et au bruit, il est donc nécessaire de discriminer certaines données le cas échéant. Néanmoins, les radars ultra-wideband offrent un compromis intéressant entre quantité d'information, vie privée, logistique, coût énergétique et coût monétaire, qui les rend propices à des applications dans la reconnaissance d'activités.

1.2 PROBLÉMATIQUE DE RECHERCHE

1.2.1 PROBLÉMATIQUE

Face au constat établi ci-dessus, les radars UWB se présentent comme une solution intéressante et peu contraignante pour acquérir des données et faire de la reconnaissance d'activités. Simples à installer et peu dépendants de leurs positions dans l'habitat, ils sont plus intéressants à mettre en place que les capteurs ambiants. De plus, recueillant des données dans un domaine électromagnétique en dehors du visible, ils ne posent

pas de problèmes majeurs pour la vie privée et la dignité des habitants. Enfin, n'étant pas portables, ils ne sont pas dépendants de la précaution et de l'attention régulière de l'utilisateur. Ils peuvent ainsi être utilisés passivement en tout temps.

C'est pourquoi le projet de recherche effectué dans le cadre de ce mémoire vise à répondre à la problématique suivante : Comment effectuer de la reconnaissance d'activités dans un habitat intelligent à l'aide de radars ultra-wideband ? Ici, le domaine de recherche renvoie aux méthodes et techniques d'intelligence artificielle qui seront recensées et expliquées au Chapitre 2. La reconnaissance d'activités est définie comme le procédé, permettant à partir de données recueillies sur l'environnement d'un utilisateur ou directement sur lui, d'en déduire l'activité ou la suite d'activités réalisées. On considérera ici surtout des activités de haut-niveau, c'est-à-dire générales et représentatives d'un ensemble d'activités spécifiques. Par exemple "faire la cuisine" plutôt que "faire cuire des pâtes" ou "faire cuire du riz". Enfin, le concept d'habitat intelligent a été défini précédemment et la notion de radars UWB sera détaillée dans le Chapitre 3.

Il s'agit ici de définir les capacités techniques des capteurs, plus particulièrement dans le cadre d'un habitat standardisé, puis de recenser et hiérarchiser les techniques d'intelligence artificielle existantes selon leur performance pour effectuer de la reconnaissance d'activités. Cette étude comparative permettra alors de mettre en lumière les méthodes les plus efficaces pour répondre à la problématique ci-dessus et de les implémenter dans un système permettant, de bout en bout, d'acquérir les données des capteurs installés dans un habitat intelligent, de les traiter, de les synthétiser et de les afficher sur une interface utilisateur.

1.2.2 MOTIVATIONS

La prise en charge des aînés et leur qualité de vie doit être un enjeu majeur pour toute société développée, et s'impose comme un défi de taille lorsqu'ils représentent une part importante de la population. Tout le monde est concerné de près ou de loin par cette problématique, que ça soit pour ses proches et ses parents, ou en anticipation de l'avenir pour soi-même. Le maintien chez soi des personnes âgées est, lorsque la situation le permet, souvent préférable à un départ en résidence. L'intelligence ambiante et la reconnaissance d'activités permettent de compenser en partie la perte d'autonomie des individus et d'assurer un suivi constant de la personne pour simplifier le travail des professionnels de la santé effectuant de l'assistance à domicile. Cela retarde au minimum leur départ en résidence et, au mieux, supprime sa nécessité.

À ce titre, le Laboratoire d'Intelligence Ambiante pour la Reconnaissance d'Activités (LIARA) se présente comme une force active pour répondre à cette problématique, et contribue à la résoudre via de nombreux projets de recherche. Il s'intéresse à la fois aux technologies portables (Thullier *et al.*, 2017; Chapron *et al.*, 2017; Maitre *et al.*, 2019), aux technologies ambiantes (Fortin-Simard *et al.*, 2015) et à des domaines connexes comme la reconnaissance de symptômes de maladies rares à partir d'articles scientifiques (Cousyn *et al.*, 2020). Ce projet de recherche permet de diversifier la collection de capteurs présents au laboratoire et d'étudier l'efficacité d'un plus grand nombre de méthodes et de synergies. Ces capteurs sont installés dans une reproduction d'un habitat entièrement équipé pour une personne, permettant de faire des tests grandeur nature et représentatifs des conditions d'une application concrète. Un plan de l'habitat est

représenté sur la Figure 1.2².

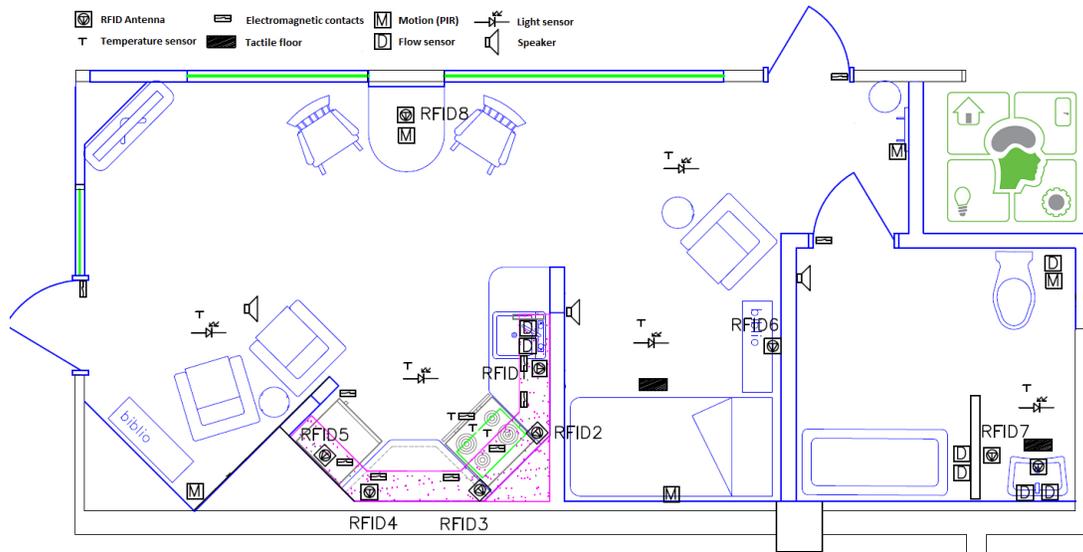


Figure 1.2 : Le plan vu du dessus de l'habitat intelligent du LIARA

Ce projet de recherche soulève plusieurs défis. Parmi eux, la familiarisation et l'évaluation de capteurs peu répandus et avec lesquels le laboratoire n'est pas familier. De plus, bien que la littérature scientifique fasse état de diverses applications connexes, la reconnaissance d'activités avec les radars ultra-wideband n'est pas très répandu dans le monde de la recherche. Enfin, le projet s'inscrit dans le cadre d'une collaboration avec l'École de Technologie Supérieure (ETS) de Montréal, utilisant les mêmes capteurs pour des travaux complémentaires. L'architecture logicielle est ainsi partiellement dépendante de leurs travaux et de l'efficacité de ceux-ci.

2. Basé sur la documentation du LIARA, plan par Sermax Automatisation inc.

1.2.3 STRUCTURE DE L'OUVRAGE

Ce mémoire est organisé de la manière suivante. La présente introduction fait office de premier chapitre. Le deuxième chapitre est dédié aux techniques d'apprentissage machine. Il s'ouvre sur les principes généraux de la matière avant de s'attarder sur les arbres de décision et les autres méthodes classiques, puis sur les réseaux de neurones. Il est suivi par un chapitre faisant état de l'art sur le sujet des radars, plus particulièrement des radars UWB, et de leurs applications dans la littérature scientifique. Le quatrième chapitre présente la structure du projet, c'est-à-dire son architecture physique et son architecture logicielle. Il se conclut par une présentation des radars utilisés et une analyse empirique de leurs performances. Le cinquième chapitre détaille le contexte de la récolte des jeux de données et leurs protocoles expérimentaux. S'en suit le dernier chapitre, consacré à l'application de méthodes d'intelligence artificielle sur un des deux jeux de données. Enfin, une conclusion sur les travaux effectués précédera la bibliographie et les annexes référencées au fur et à mesure de l'ouvrage.

CHAPITRE II

L'APPRENTISSAGE MACHINE

2.1 INTRODUCTION À L'APPRENTISSAGE MACHINE

L'intelligence artificielle est, d'après S. Russel et P. Norvig, l'étude d'agents intelligents capables de percevoir des informations sur leur environnement et d'effectuer des actions en conséquence (Russell et Norvig, 2020). Cette définition n'est qu'un exemple, car elle s'appuie sur la notion d'intelligence, elle-même sujette à de nombreuses nuances philosophiques. Afin de se distancier de la philosophie et de la sémantique, l'intelligence artificielle sera ici définie comme l'utilisation de la neurobiologie, de la logique, des mathématiques et de l'informatique pour faire résoudre à des machines des problèmes à forte complexité algorithmique actuellement uniquement réalisable par des êtres biologiques aux capacités cognitives avancées (humains, certains autres animaux). L'apprentissage et la capacité d'adaptation sont des concepts clé de la notion d'intelligence, ainsi, l'apprentissage machine est une notion majeure de l'intelligence artificielle.

L'apprentissage machine peut se résumer comme la capacité, pour un système informatisé, à améliorer ses résultats et adapter son comportement en fonction de ses expériences passées. Ainsi, les algorithmes d'apprentissage machine permettent de résoudre des problèmes qui peuvent sembler simples pour un être humain mais qui sont en réalité difficiles, voir impossibles, à réaliser via des algorithmes classiques. On peut citer, par exemple, la capacité à reconnaître un objet à partir d'une image ou à

identifier une phrase dans un enregistrement audio. La reconnaissance de l'activité d'une personne en fonction de données extraites de son environnement (image, son, position de la personne, mouvement des objets de l'environnement, etc.) fait partie de ces problèmes complexes nécessitant l'utilisation des techniques d'apprentissage machine pour être résolus.

Pour cela, il est nécessaire d'acquérir un ensemble conséquent de données associées à des activités et de les utiliser à titre d'exemples pour faire apprendre l'algorithme. On parle alors de la récolte d'un jeu de données. Dans celui-ci, chaque instance est associée à une classe à l'aide d'une étiquette. Ces couples instance de données/étiquette vont être fournis en entrée de la phase d'apprentissage de l'algorithme, appelée entraînement. Une fois l'apprentissage effectué, un modèle propre à la résolution du problème pour lequel il a été entraîné est généré. Il suffit alors de fournir de nouvelles données non-étiquetées pour que le modèle déduise, grâce à son expérience, la classe à laquelle appartiennent ces données.

Cette méthode d'apprentissage est désignée sous le terme d'apprentissage supervisé (Russell et Norvig, 2020). Elle nécessite une intervention humaine importante pour étiqueter les données. C'est cette méthode qui est utilisée pour la contribution de ce mémoire. Cependant, d'autres méthodes existent. L'apprentissage non-supervisé implique pour l'algorithme de créer lui-même des classes permettant de distinguer les données, dont le nombre peut lui être communiqué à l'avance ou non selon les techniques. Cette méthode a l'avantage d'être beaucoup moins chronophage pour l'acquisition des jeux de données, mais les performances peuvent être moins qualitatives. L'apprentissage par

renforcement, quant à lui, peut être illustré par la méthode de la carotte et du bâton (Sutton et Barto, 2018). L'algorithme est amené à recevoir une récompense lors d'une classification correcte, et optionnellement une punition en cas d'échec. La maximisation des récompenses entraîne alors naturellement l'algorithme à améliorer ses performances.

De nombreuses techniques existent pour effectuer de l'apprentissage machine. Elles se différencient de diverses manières et possèdent toutes leurs propres avantages et inconvénients. Plusieurs grandes catégories d'algorithmes seront présentées dans cette section, illustrées par quelques algorithmes de référence. Un résumé de leurs caractéristiques phares et de leurs avantages et inconvénients permettra de les mettre en contraste les unes par rapport aux autres. Par la suite, certains des algorithmes issus de ces catégories seront testés pour résoudre la problématique de la reconnaissance d'activité. Ensuite, leurs performances pour la résolution de ce problème seront évaluées et discutées.

2.2 ARBRES DE DÉCISION ET K PLUS PROCHES VOISINS

2.2.1 ARBRES DE DÉCISION

Un arbre de décision est une série de questions permettant de discriminer des données selon leurs caractéristiques dans différentes classes. Le schéma permettant de visualiser cette méthode ressemble à un arbre, d'où le nom d'arbre de décision. La métaphore est filée pour décrire ses caractéristiques. On appelle ainsi "branche" un cheminement partant de la première question, la "racine", jusqu'à la dernière question. Après celle-ci, une "feuille" est présente pour attribuer une classe aux données. Un

exemple d'arbre de décision pour déterminer la possibilité de jouer au Tennis selon la météo est illustré en Figure 2.1.

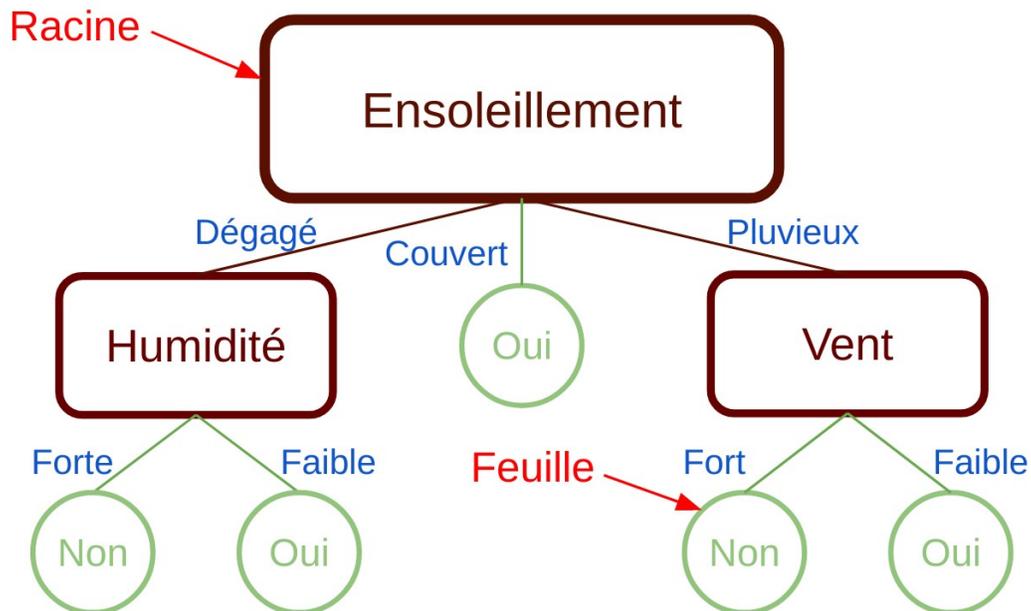


Figure 2.1 : Exemple d'arbre de décision et illustration de la métaphore filée.

De nombreux algorithmes de construction d'arbres de décision existent. Parmi les plus connus on retrouve l'algorithme ID3 (Quinlan, 1986), son amélioration C4.5 (Quinlan, 1993), ou l'algorithme Classification And Regression Tree (CART) (Breiman *et al.*, 1984). ID3 et C4.5 sont tout les deux basés sur les notions de gain d'information et d'entropie. Afin de choisir l'attribut du nœud courant sur lequel va être appliqué la discrimination, nous cherchons à maximiser le gain d'information. Le gain d'information est donné par l'équation (2.1), avec D_p le jeu de données avant séparation, D_j le jeu de données après séparation, f l'attribut utilisé pour la séparation, N_p et N_j les nombres respectifs d'échantillons du parent et des enfants, et $I()$ le calcul de l'impureté.

Pour l'impureté on utilise l'entropie, valeur entre 0 et 1, qui permet de représenter l'homogénéité d'un échantillon. L'équation (2.2) représente l'entropie, avec $p(i|t)$ la proportion d'appartenance à la classe i au noeud t . Une entropie de 0 signifie que toutes les instances de l'échantillon appartiennent à la même classe, alors qu'une entropie de 1 signifie que chaque classe est représentée de manière égale dans l'échantillon.

$$IG(D_p, f) = I(D_p) - \sum_{j=1}^m \frac{N_j}{N_p} I(D_j) \quad (2.1)$$

$$I_H(t) = - \sum_{i=1}^c p(i|t) \log_2 p(i|t) \quad (2.2)$$

Ainsi, lorsqu'on essaie de maximiser le gain d'information, on cherche à sélectionner l'attribut qui, après son évaluation, donne des échantillons avec l'entropie globale la plus faible. Pour choisir un attribut, on sélectionne, parmi ceux restants, celui dont le gain d'information est le plus élevé. On effectue ensuite le même procédé pour chaque nœud enfant généré. Lorsque l'entropie est égale à 0, alors le nœud est pur et devient une feuille, c'est-à-dire une attribution de classe. Une fois que chaque branche possède une feuille, l'algorithme est terminé. L'algorithme CART, quant à lui, utilise l'indice de Gini pour évaluer son gain d'information. L'indice de Gini est calculé à l'aide de l'équation (2.3). Cet indice représente la probabilité d'erreur, pour le choix aléatoire d'une instance dans un échantillon, d'une classification aléatoire pondérée selon la proportion des classes dans ce même échantillon. De la même manière que pour l'entropie, un échantillon d'indice de Gini 0 est considéré comme pur.

$$I_G(t) = 1 - \sum_{i=1}^c p(i|t)^2 \quad (2.3)$$

En effectuant un apprentissage maximal, c'est-à-dire en s'assurant que chaque instance soit classée correctement sans se soucier de limiter la profondeur ou la largeur de l'arbre, on peut tomber dans une situation de sur-apprentissage. Cela signifie que l'arbre de décision est trop calqué sur le jeu de données. Il présentera alors des performances médiocres lors de son utilisation avec de nouvelles données. L'exemple le plus sévère de sur-apprentissage consisterait à avoir autant de feuilles que d'instances dans le jeu de données. On aurait alors 100% de précision sur le jeu de données d'entraînement, mais une précision beaucoup plus faible sur le jeu de données de test. Ainsi, il est utile de limiter la taille de l'arbre ou d'appliquer certains algorithmes d'élagage visant à supprimer les parties trop spécifiques des branches. À l'inverse, trop limiter le modèle conduirait à du sous-apprentissage, c'est-à-dire un cas où l'arbre de décision est tellement simpliste qu'il n'effectue pas de classification efficace. L'exemple le plus extrême consiste à classer toutes les instances dans une seule classe dès le début. On aura alors quelques cas justes, mais le modèle ne démontrerait aucune forme d'apprentissage. Les notions de sur-apprentissage et de sous-apprentissage ne s'appliquent pas uniquement aux arbres de décision et peuvent être étendues à tout autre méthode d'apprentissage machine.

2.2.2 APPRENTISSAGE D'ENSEMBLES

Afin d'améliorer les performances des arbres de décisions pour la prédiction, il existe plusieurs techniques visant à les utiliser en groupe de diverses manières. Ces techniques sont celles de l'apprentissage d'ensembles. Deux concepts clés permettent de catégoriser ces techniques. Le premier est le dopage (boosting en anglais) (Russell et Norvig, 2020). Il consiste à entraîner plusieurs arbres de décision sur différents sous-ensembles du jeu de données en mettant, au fur et à mesure, l'emphase sur les cas mal classés par les classeurs précédents. Cela permet aux arbres de compenser leurs faiblesses mutuelles. Le second concept consiste également à entraîner plusieurs arbres sur différents sous-ensembles du jeu de données, mais avec remise : c'est le "bagging", pour *bootstrap aggregating* (Russell et Norvig, 2020). La différence repose essentiellement sur le fait que le dopage va utiliser des arbres peu performants individuellement et peu complexes sur des échantillons qui leur sont propres, et va chercher à créer une synergie particulière entre ces arbres. Le bagging regroupe plutôt des arbres très spécifiques, proches du sur-apprentissage, mais sur des échantillons dont les instances peuvent être ré-utilisées aléatoirement pour l'apprentissage d'autres arbres. Ainsi, on se repose essentiellement sur l'idée qu'un grand nombre d'arbres spécifiques va, collectivement, se moyenner de manière à limiter le sur-apprentissage.

LE DOPAGE ET ADABOOST

Le dopage fait apprendre plusieurs arbres séquentiellement dans le but d'améliorer itérativement les performances sur les cas difficiles. Par exemple, considérons trois

arbres A1, A2 et A3. L'arbre A1 a des difficultés à classer certains cas. Ils seront donc inclus dans l'échantillon utilisé pour l'entraînement de l'arbre A2. Cependant, certains cas ne font alors pas consensus entre A1 et A2. L'arbre A3 va alors être entraîné avec ces éléments. Après entraînement, un échantillon de test à classer va être soumis aux trois arbres et un vote majoritaire est effectué. On retient alors la classe majoritairement prédite par les arbres individuels. Ainsi, cette technique repose sur la synergie entre des arbres simples compensant leurs faiblesses mutuelles.

L'algorithme AdaBoost (Freund et Schapire, 1997), pour adaptative boosting (dopage adaptatif), utilise ce principe avec un système de pondération des instances du jeu de données d'entraînement. Initialement, chaque instance possède le même poids. On entraîne un premier arbre et on regarde ses résultats. Un poids est attribué à cet arbre selon ses performances : s'il a une précision supérieure à 50%, son poids sera positif. Sinon, il sera négatif. Ainsi, il suffit de combiner le poids avec les résultats du classeur pour qu'il soit utile en tout temps : un classeur avec 30% de précision qu'on inverse à cause de son poids permet au final d'avoir une prédiction d'une précision de 70%³. Ensuite, on met à jour les poids des instances du jeu de données. Un cas mal classé par un arbre de décision de poids positif voit son poids augmenter. Un deuxième arbre est alors entraîné, mais la nouvelle distribution des poids fait qu'il sera plus enclin à apprendre à partir des cas difficiles, ici ceux mal classés par le premier arbre. En répétant le processus autant que souhaité, on obtient un ensemble d'arbres très simples et peu performants individuellement, mais beaucoup plus précis collectivement. Lors de la prédiction, le poids de chaque arbre est pris en compte lors du vote. Ainsi, les arbres les

3. Dans le cas d'une classification binaire uniquement

plus performants influent plus fortement sur la décision du groupe et compensent les arbres les plus faibles.

LE BAGGING ET LA FORÊT ALÉATOIRE

Le bagging, quant à lui, repose sur le principe de la sagesse collective : un ensemble d'arbres très spécifiques et non corrélés sera collectivement plus performant que chaque arbre individuellement. Pour cela, on entraîne plusieurs arbres de décision sur des sous-ensembles du jeu de données dont les instances sont tirées avec remise aléatoire. Ces sous-ensembles d'entraînement sont de même taille que le jeu de données, mais les instances qui les composent peuvent varier. Par exemple, dans le jeu de données : {1,2,3}, on peut obtenir les jeux d'entraînements aléatoires {1,2,2}, {1,1,3}, {3,3,3}, {1,2,3}, ... Dans cette méthode, les arbres ne sont généralement pas élagués et sont parfois sujets au sur-apprentissage pour garantir le principe de spécificité individuelle. L'un des effets de l'entraînement de plusieurs arbres sur des échantillons légèrement différents se trouve dans la sélection de l'attribut de la racine qui peut alors varier d'un arbre à l'autre. Dans le cas de la forêt aléatoire, une méthode de bagging populaire, le phénomène est amplifié (Breiman, 2001; Ho, 1995). En effet, parmi P paramètres, un sous-ensemble M aléatoire pour chaque arbre est choisi, et seuls les attributs appartenant à ce sous-ensemble sont autorisés pour discriminer les instances au niveau des feuilles. La taille de ces sous-ensembles M est généralement $\frac{P}{3}$, \sqrt{P} ou $\log_2 P$. Ainsi, la variété des attributs est forcée, rendant la forêt très diversifiée. Ici, il est important de noter que le vote des arbres n'est pas pondéré lors de la prédiction de l'ensemble.

2.2.3 KNN OU LES K PLUS PROCHES VOISINS

L'algorithme des K plus proches voisins, ou *K Nearest Neighbors (KNN)*, est un algorithme de classification qui consiste à prédire une classe d'une instance particulière en fonction de la classe majoritaire parmi ses K voisins les plus proches (Cover et Hart, 1967). Considérons un ensemble de points dans un espace à deux dimensions auxquels on aurait associés des classes. Pour prédire la classe d'un nouveau point, on va déterminer quels sont ses K voisins les plus proches à l'aide d'une fonction de distance, puis on va regarder quelle est la classe la plus prépondérante parmi ces voisins. On associe alors cette classe au nouveau point. Par exemple, pour un K égal à 1, la classe d'un nouveau point est directement la classe de son voisin le plus proche. La Figure 2.2 illustre cette méthode pour un K égal à 5. Dans cet exemple, l'instance inconnue se verra attribuer la classe B.

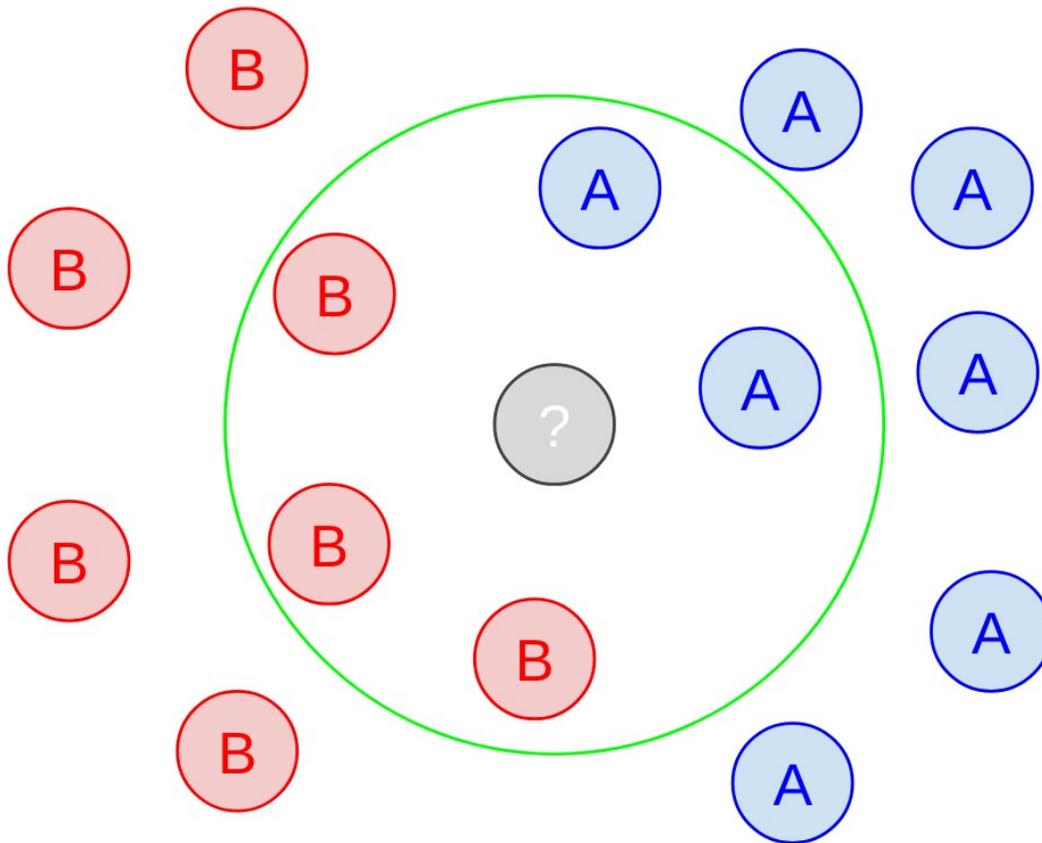


Figure 2.2 : Exemple de la méthode des K plus proche voisins avec $K = 5$

Un petit K permet d'avoir des frontières très distinctes entre les classes, mais rend la prédiction vulnérable au bruit. En effet, les cas proches anormaux, dus à l'imperfection des données ou à des exceptions très spécifiques, ont une forte influence. Au contraire, un grand K permet d'éviter l'influence d'anomalies, mais rend la frontière entre les classes moins précise. Certaines subtilités qu'un K plus faible pourrait prendre en compte sont ainsi ignorées. On retrouve ici le même dilemme sur-apprentissage / sous-apprentissage illustré précédemment pour le cas des arbres de décision. Des heuristiques existent pour définir automatiquement un K proche de l'optimalité (Muja et Lowe, 2009). Sans l'utilisation de ces heuristiques, son paramétrage nécessite des tests empiriques pour un

réglage efficace.

L'algorithme KNN est un algorithme non-paramétrique, comme les arbres de classification. Ils sont particulièrement efficaces lorsque le jeu de données est très fourni. En revanche, ils sont très susceptibles au phénomène de sur-apprentissage et peuvent être gourmands en mémoire comme en temps de calcul. La version la plus simple de la méthode KNN est aussi impactée par une asymétrie des données au niveau des classes, c'est-à-dire par des proportions déséquilibrées dans le nombre d'instances appartenant à chaque classe. Néanmoins, cela peut être compensé par une pondération des classes sous-représentées. Il existe aussi plusieurs variantes de l'algorithme, comme le KNN pondéré qui consiste à accorder une importance croissante au voisin selon sa proximité avec les données à évaluer (Dudani, 1976). L'algorithme KNN peut aussi être utilisé pour effectuer de la régression plutôt que de la classification, c'est-à-dire pour estimer la valeur de certains paramètres en fonction de la valeur des voisins proches (Altman, 1992).

2.3 LES RÉSEAUX DE NEURONES ET L'APPRENTISSAGE PROFOND

2.3.1 PRINCIPE DU PERCEPTRON

Le perceptron est un modèle de classification binaire inspiré du fonctionnement d'un neurone biologique. Les neurones sont des cellules excitables qui forment, à l'aide de leurs inter-connexions, un système nerveux. Ils communiquent à l'aide de synapses et peuvent, en fonction des signaux qu'ils reçoivent, s'exciter et émettre eux-mêmes des influx nerveux. Le perceptron est une représentation mathématique simplifiée de ce

principe (Rosenblatt, 1958). Le neurone du perceptron va avoir en entrée une instance d'un jeu de données sous forme de plusieurs entrées numériques, ainsi qu'un biais. Ce biais et ces entrées correspondent aux influx des autres neurones du système nerveux. Ici, les paramètres à apprendre lors de la phase d'entraînement sont les poids des synapses. Grâce à une pondération des différents influx nerveux reçus, une fonction d'activation va définir si le neurone émettra, ou n'émettra pas, un influx nerveux. La fonction d'activation correspond à la comparaison d'une somme pondérée des signaux d'entrées avec un seuil. Ainsi, on peut effectuer une classification binaire avec présence d'influx : 1 et absence d'influx : 0. Si on souhaite effectuer de la classification non-binaire, il suffit alors d'utiliser la méthode "one versus all" qui consiste à entraîner un perceptron par classe (Nilsson, 1965). Chaque perceptron doit être actif pour sa classe respective, et inactif pour toute autre classe. Une représentation du perceptron est illustrée sur la Figure 2.3.

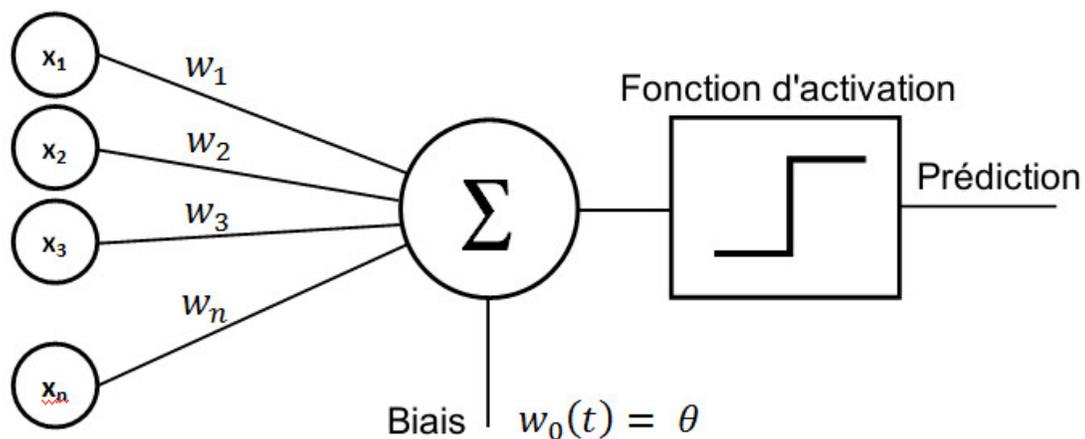


Figure 2.3 : Schéma du perceptron

Adaline, pour ADAPtive LINEar neurons, est une évolution du perceptron qui permet d'établir un ordre de confiance sur la prédiction de classe (Widrow et Hoff, 1960). La fonction d'activation est une fonction linéaire, elle renvoie donc une valeur continue permettant de quantifier la certitude de la prédiction. Par exemple, après un passage dans une fonction de normalisation entre 0 et 1, plus la valeur est proche de 1 ou de 0, plus la certitude de détecter ou respectivement de ne pas détecter la classe augmente. Une valeur proche de 0.5 représente un cas incertain. La régression logistique est un modèle probabiliste similaire à Adaline dont la fonction d'activation est une fonction sigmoïde (équation (2.4)) (Kleinbaum *et al.*, 2002; Hosmer Jr *et al.*, 2013). Ainsi, la sortie correspond à la probabilité, entre 1 et 0, que l'instance appartienne à une classe particulière. La probabilité d'appartenir à l'autre classe est le résultat soustrait à 1.

$$\frac{e^x}{e^x + 1}; \forall x \in \mathbb{R} \quad (2.4)$$

2.3.2 LES RÉSEAUX DE NEURONES PROFONDS

L'apprentissage profond est une méthode mettant en scène des neurones dans un réseau sous forme de couches. Reprenons le modèle du perceptron et considérons qu'il existe n neurones, chacun connectés avec toutes les entrées. Chaque neurone a ses propres poids associés à ces entrées. Toutes les sorties de cette couche de neurones sont reliées à un autre neurone, possédant lui-même des poids pour pondérer ses entrées, et renvoyant une prédiction binaire en sortie de sa fonction d'activation. On a ici un perceptron multi-couches avec une couche d'entrée (N entrées + biais), une couche

cachée (M neurones) et une couche de sortie (un neurone). Le schéma de ce perceptron multi-couches est illustré en Figure 2.4. On considère par la suite qu'un réseau de neurones possédant plus d'une couche cachée est un réseau de neurones profond. Le modèle précédent peut facilement s'adapter à de la classification non-binaire en ajoutant autant de neurones que de classes à détecter sur la couche de sortie. Chaque neurone de sortie est connecté à tous les neurones de la couche précédente, ce qui revient à procéder de même manière similaire à la méthode "one versus all" décrite dans la sous-section précédente. Cependant, contrairement au "one versus all", cette technique n'est pas impactée par le déséquilibre de classes.

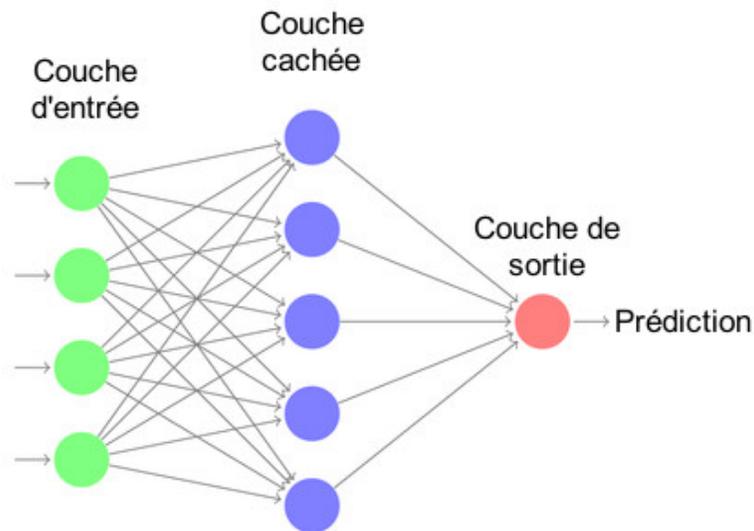


Figure 2.4 : Schéma d'un perceptron multi-couches à une couche cachée pour une classification binaire

Là où le perceptron utilise une fonction d'activation binaire et où Adaline utilise une fonction d'activation linéaire, les réseaux de neurones profonds vont utiliser des fonctions d'activation non-linéaires (Karlik et Olgac, 2011). En effet, l'utilisation de ce

type de fonction est nécessaire pour appliquer le principe de propagation arrière (Rumelhart *et al.*, 1995), central pour l'entraînement de ces réseaux, et dont le fonctionnement est détaillé plus bas. Parmi les fonctions d'activations les plus utilisées, on peut citer les fonctions \tanh (équation (2.5)) et sigmoïde (équation (2.4)). Elles sont toutes les deux monotones et ont un intervalle de sortie fini, garantissant ainsi la convexité de la descente de gradient pour la propagation arrière. Cela signifie qu'il est plus facile d'optimiser l'étape d'apprentissage du réseau de neurones. Cependant, elles utilisent des calculs qui peuvent être coûteux et impactent fortement le temps dédié à l'entraînement pour des réseaux de neurones très profonds.

$$\tanh x = \frac{\sinh x}{\cosh x} = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.5)$$

La fonction d'activation Rectified Linear Unit (ReLU) s'apparente à une fonction linéaire (Nair et Hinton, 2010a). Sa formule est représentée par l'équation (2.6). C'est pourtant une fonction non-linéaire, elle permet donc la rétro-propagation du gradient tout en garantissant une convergence rapide du modèle lors de l'entraînement. Elle possède néanmoins quelques inconvénients. Par exemple, contrairement à la fonction \tanh , elle n'est pas centrée autour de 0. De plus, elle n'a pas non plus d'intervalle de sortie fini. Enfin, elle présente le problème dit des "neurones morts", c'est-à-dire que certains neurones ne s'activent que très rarement, voire jamais, peu importe leurs entrées. Ces neurones morts contribuent à la complexité du modèle, augmentant ainsi son temps d'exécution et son coût mémoire sans amélioration significative des performances. Plusieurs variantes de ReLU existent pour compenser ces limites ou pour s'adapter à

des problématiques spécifiques. Parmi celles-ci, on retrouve par exemple la fonction ReLU paramétrique, la fonction ReLU à fuite ou la fonction ReLU à bruit Gaussien (Xu *et al.*, 2015; Nair et Hinton, 2010b).

$$y = \max(0, x) \quad (2.6)$$

La classification fonctionne sur le principe de la propagation avant, où la couche d'entrée communique avec la première couche cachée, lui permettant ainsi de calculer ses valeurs de sorties. Ces nouvelles valeurs sont alors communiquées à la deuxième couche qui calcule à son tour ses sorties, processus répété jusqu'à la couche N-1. Cette dernière envoie ses sorties à la couche N aussi appelée couche de sortie, renvoyant la prédiction finale. C'est un procédé assez naturel et facile à imaginer qui permet de se représenter le modèle de manière linéaire, sous forme de couches successives s'activant chacune à leur tour depuis la couche d'entrée jusqu'à la couche de sortie.

Quant à l'apprentissage, il fonctionne sur le principe de la propagation arrière, aussi appelé rétro-propagation du gradient ou propagation de l'erreur (Rumelhart *et al.*, 1995). Imaginons un réseau de neurones profond qui, en fonction d'un certain nombre de caractéristiques, classe un fruit inconnu dans l'une des trois classes suivante : banane, pomme ou orange. Un réseau de neurones mal entraîné va, lorsqu'on lui présentera une pomme, prédire que c'est une banane à 72%, une pomme à 19% et une orange à 9%. Idéalement, on souhaiterait qu'il réponde respectivement 0%, 100% et 0%. On va alors essayer d'influencer positivement les poids des neurones de la dernière couche cachée ayant un impact sur l'activation de la classe "pomme", et négativement ceux influençant

l'activation de la classe "banane" et de la classe "orange". On va néanmoins impacter les poids des synapses proportionnellement à la taille de l'erreur, c'est-à-dire l'écart entre la prédiction fournie et la prédiction souhaitée pour chaque classe. Ici, on va impacter très négativement les synapses qui activent la classe banane, et moins négativement ceux de la classe orange.

Une fois les synapses reliant l'avant-dernière couche à la couche de sortie impactés, on va répéter ce processus entre l'avant-dernière couche et l'avant-avant-dernière couche. En faisant la somme des impacts désirés à l'étape précédente, on peut déterminer la différence entre l'activation souhaitée des neurones de l'avant-dernière couche, et l'activation réelle. Prenons un neurone aléatoire de cette couche et admettons qu'il impacte très positivement la classe pomme, un peu positivement la classe orange et très négativement la classe banane. L'étape précédente nous amènera à vouloir augmenter son impact global fortement pour corriger le résultat des classes "banane" et "pomme", et à le diminuer légèrement pour corriger le résultat de la classe "orange". Dans l'étape actuelle, on va donc augmenter le poids des synapses ayant entraîné son activation (et plus intensément celles ayant un fort impact) et diminuer le poids de celles contribuant à sa désactivation. En étendant ce principe à tous les neurones de l'avant-dernière couche, on se retrouve dans un cas similaire à l'étape précédente, avec non pas trois, mais N neurones. On répète ce procédé jusqu'à ce que les poids des synapses reliant la couche d'entrée à la première couche cachée soient impactés, puis on répète la propagation arrière pour chaque instance du jeu de données. C'est l'entièreté de ce procédé qui permet d'entraîner le réseau de neurones.

2.3.3 LES RÉSEAUX DE NEURONES CONVOLUTIFS

Les réseaux de neurones convolutifs (Convolutional Neural Network (CNN)) sont des réseaux de neurones profonds dont le fonctionnement est basé sur une opération de calcul : le produit de convolution. Ce sont des réseaux de neurones particulièrement adaptés pour du traitement visuel comme la reconnaissance d'images, car leur fonctionnement est inspiré du cortex visuel des animaux (LeCun *et al.*, 1995; Goodfellow *et al.*, 2016). L'opération de convolution dans les CNN consiste en l'application du produit de convolution aux matrices entre un noyau de taille (k, l) et chaque projection de taille (k, l) possibles dans une matrice de taille (m, n) . On obtient ainsi une matrice de taille $(m - k + 1, n - l + 1)$ plus petite que la matrice initiale. Cette opération est illustrée en Figure 2.5. On observe que les éléments situés au centre de la matrice sont plus représentés dans les calculs que les éléments en bordure. Ils ont ainsi un impact plus important sur le résultat. C'est pourquoi un rembourrage, ou "padding", peut être appliqué. En rajoutant un rembourrage neutre de dimension $(k - 1, l - 1)$ en bordure de la matrice initiale, on obtient alors, par une convolution avec un noyau de taille $(k - l)$, une matrice finale de taille (m, n) . Ainsi, la matrice finale possède les mêmes dimensions que celles de l'image initiale sans rembourrage. La Figure 2.6 représente la convolution d'une image avec un noyau permettant la détection de bords⁴.

4. Wikimedia, https://commons.wikimedia.org/wiki/File:Edge_detection.png

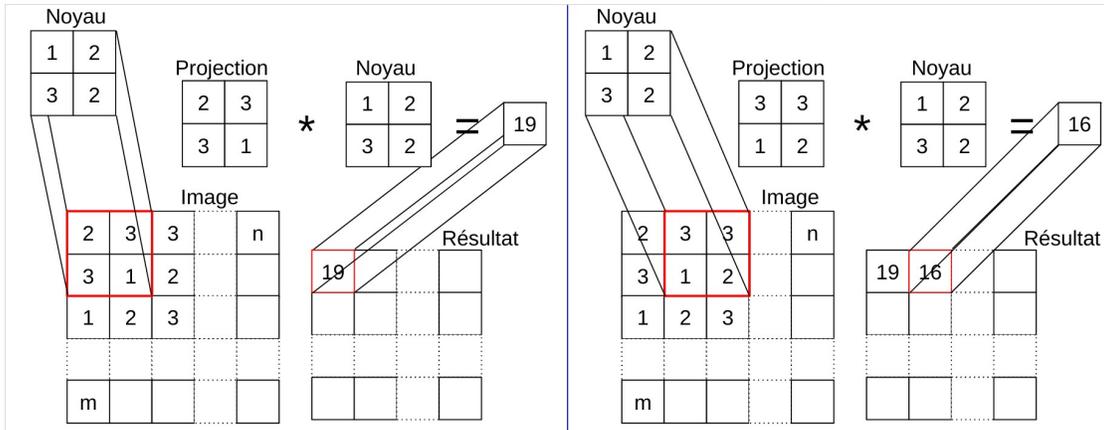


Figure 2.5 : Illustration d'une étape de l'opération de convolution



Figure 2.6 : Application d'un filtre de détection de bord comme noyau de convolution sur une image numérique

Une autre opération, dite de mise en commun ou "pooling", est également centrale dans le fonctionnement des réseaux de neurones convolutifs. Elle sert à réduire la dimensionnalité des matrices de sorties en condensant l'information révélée par l'opération de convolution. Prenons un exemple de mise en commun standard, le "max pooling". Cette opération fonctionne de manière similaire à l'opération de convolution vue précéd-

demment en remplaçant le produit de convolution par une maximisation. On ferait alors glisser une projection de taille (k, l) sur une matrice initiale de taille (m, n) . Cependant, à la place d'effectuer un produit de convolution entre un noyau (k, l) et la projection obtenue, on se contente de prendre la valeur maximale présente dans la projection. Dans ce cas, on obtient une matrice finale de taille $(m - k + 1, n - l + 1)$, comme l'opération de convolution précédente sans rembourrage. Pour une réduction de dimensionalité plus importante, on peut ajouter un pas. C'est-à-dire qu'on ne considère pas chaque projection de taille (k, l) possible sur la matrice initiale, mais seulement une fraction d'entre-elles. Par exemple, un pas de 2 réduirait chaque dimension de l'image de sortie de moitié. La Figure 2.7 illustre une opération de "max pooling" avec un noyau de dimension $(2, 2)$ et un pas de 1 sur l'image obtenue précédemment après convolution.

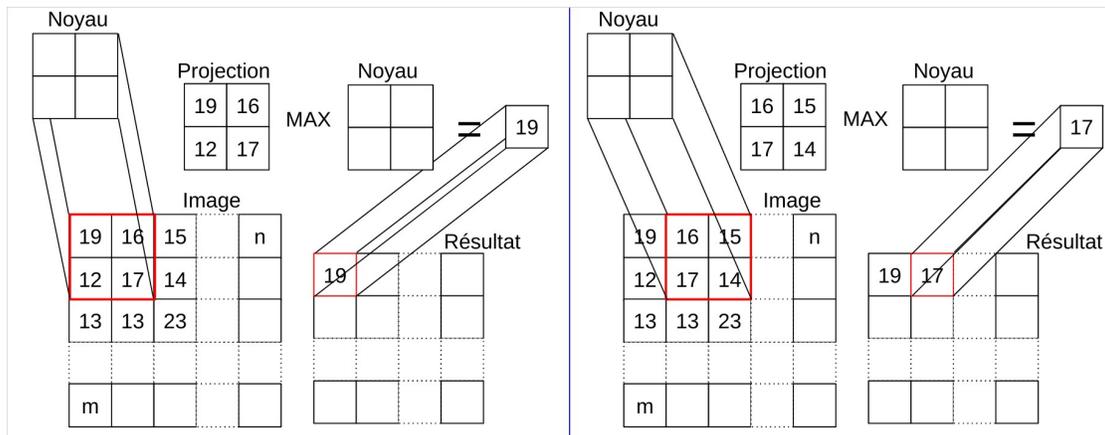


Figure 2.7 : Illustration d'une étape de l'opération de mise en commun "max pooling"

Dans les réseaux de neurones convolutifs, la principale fonction de la convolution est d'extraire des caractéristiques sur les données d'entrées à l'aide de noyaux de convolution faisant office de filtres. L'application de divers filtres les uns à la suite

des autres permet d'extraire des caractéristiques plus complexes, d'où la présence de plusieurs couches dans les réseaux de neurones convolutifs. Par exemple, dans les réseaux appliqués à la reconnaissance d'objets, les premières convolutions permettent de détecter des caractéristiques simples comme les bords, les coins ou des formes élémentaires. Les convolutions intermédiaires vont permettre de détecter des caractéristiques plus avancées comme des formes complexes ou des textures. Enfin, les convolutions appliquées en fin de réseau vont permettre la reconnaissance de caractéristiques complexes comme la présence d'objets spécifiques (Goodfellow *et al.*, 2016). L'objectif de l'opération de mise en commun, outre sa propriété de réduction de dimensionalité, est de réduire l'impact des différences mineures observables sur des données très similaires. Dans l'exemple de la reconnaissance d'objets, elle limite les conséquences de légères translations latérales, de rotations, de variations de textures mineures, etc. Enfin, la réduction de dimensionalité produite par la mise en commun permet de réduire le coût en mémoire et en calcul, bien que cet aspect peut aussi être obtenu avec l'opération de convolution en l'absence de rembourrage ou lorsqu'un pas supérieur à 1 est utilisé.

Les réseaux de neurones convolutifs ont la particularité d'être moins coûteux en mémoire que les réseaux de neurones profonds étudiés précédemment, car ils mutualisent les poids d'une même couche sous la forme du noyau de convolution. Ces derniers peuvent être initialisés de manière aléatoire et appris de manière non-supervisée, ne nécessitant ainsi aucune intervention humaine. De plus, là où le perceptron multicouche renvoie le résultat sous la forme d'une classe, les réseaux de neurones convolutifs peuvent renvoyer des structures plus complexes. Dans le cas de la reconnaissance d'objets, on peut souhaiter que le réseau ne se contente pas de déterminer la classe du ou des objets

présents, mais qu'il renvoie également leurs délimitations spatiales dans l'image. Bien évidemment, si on désire obtenir des structures de ce type, il est impératif de limiter la réduction de dimensionalité en utilisant les opérations de mise en commun avec parcimonie. Quelques exemples de réseaux de neurones convolutifs sont représentés sur les Figures 2.8, 2.9 et 2.10 (Krizhevsky *et al.*, 2012; Szegedy *et al.*, 2015; He *et al.*, 2016).

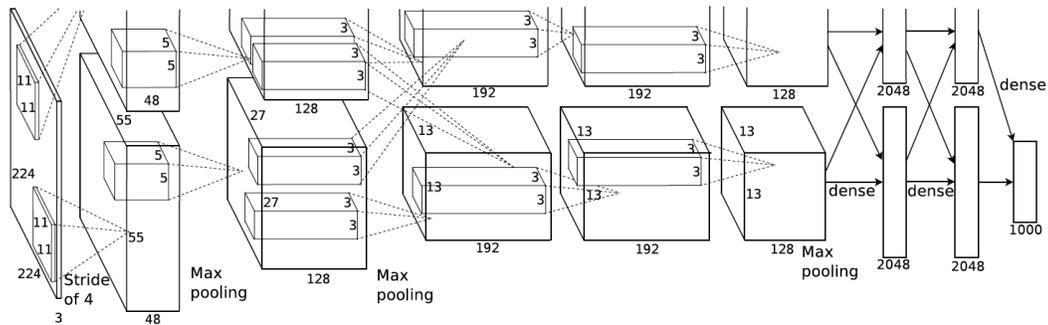


Figure 2.8 : Architecture du réseau de neurones convolutif Alexnet

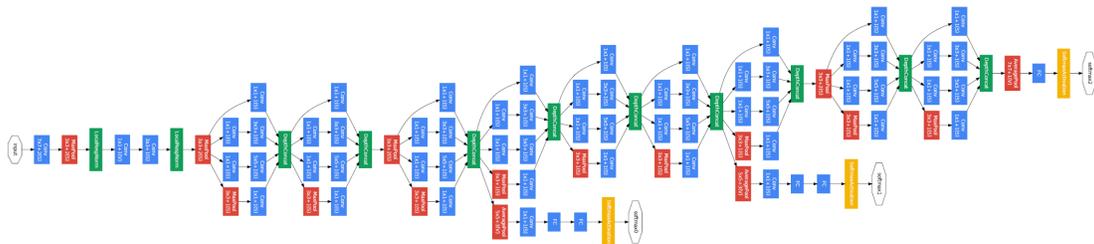


Figure 2.9 : Architecture du réseau de neurones convolutif GoogLeNet

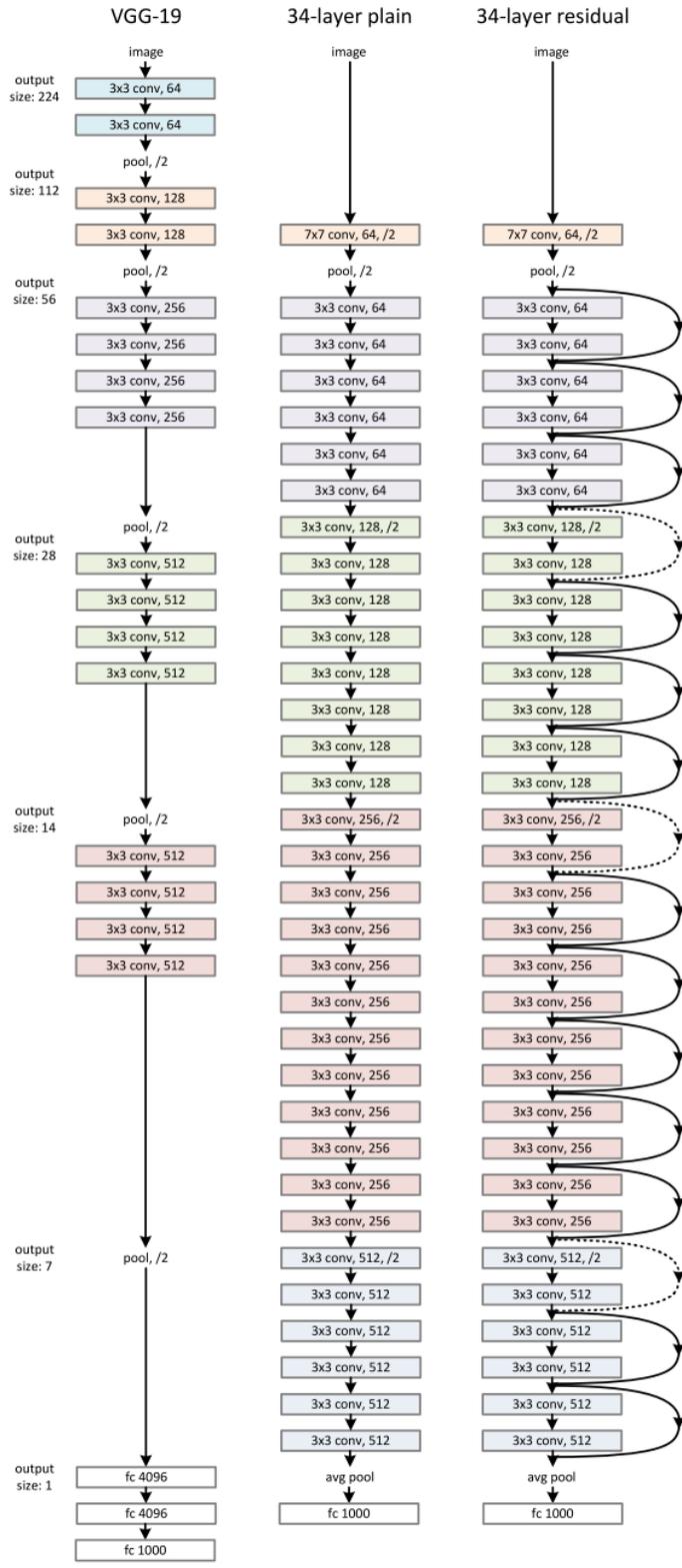


Figure 2.10 : Architecture du réseau de neurones convolutif Resnet

CHAPITRE III

LES RADARS ULTRA-WIDEBAND

3.1 FONDAMENTAUX DES RADARS

3.1.1 THÉORIE DES RADARS

PRINCIPE DE BASE ET FONCTIONNEMENT D'UN RADAR

Le terme radar, bien qu'utilisé comme un nom commun dans beaucoup de situations, est en réalité un acronyme anglais des mots RAdio Detection And Ranging, qui signifient "détection et télémétrie radio". Ainsi, un radar est un système utilisant la bande radio du champ électromagnétique pour détecter un objet et estimer sa distance par rapport à lui-même (Skolnik, 2001). Le principe physique à la source de ce système peut être simplifié à une analogie de l'écho qui peut apparaître en criant dans une vallée. Le son de la voix est projeté dans la direction du cri et, au contact d'une falaise, va être réfléchi. C'est pourquoi on peut entendre un écho du cri, bien qu'atténué. Lorsque de multiples parois différentes et distantes sont présentes, on peut entendre l'écho plusieurs fois, car le signal est réfléchi par plusieurs surfaces différentes. Un radar fonctionne de manière similaire, mais avec des ondes électromagnétiques plutôt qu'avec des ondes sonores. Ainsi, un système radar est équipé d'un émetteur, substitut de nos cordes vocales dans l'exemple précédent, et d'un récepteur, substitut de nos oreilles. Des antennes sont généralement présentes, une pour l'émetteur et une pour le récepteur, bien que parfois, une seule antenne est suffisante pour les deux. Dans ce cas, un duplexeur est ajouté en

complément, permettant d'effectuer la commutation entre l'émetteur et le récepteur.

Le fonctionnement d'un radar est le suivant : l'émetteur va, à l'aide de l'antenne, diffuser une onde radio. Selon le type d'antenne, l'onde peut-être concentrée dans un champ limité, bien qu'au moins une fraction de l'amplitude maximale est envoyée dans toutes les directions. C'est la phase d'illumination. Le signal est ensuite reflété sur tous les objets d'une densité atomique contrastant fortement avec leur environnement, que ça soit de l'air, de l'eau ou du vide. La réflexion est particulièrement forte sur les objets conducteurs, comme le métal. C'est la phase de réflexion. L'antenne reçoit alors ces réflexions du signal émis, atténuées, à différents instants selon la distance de l'objet. Un signal réfléchi par un objet situé à 1 mètre de distance de l'émetteur sera reçu trois fois plus tôt qu'un signal réfléchi par un objet situé à 3 mètres de distance. C'est la phase de réception. Connaissant la vitesse de la lumière dans un milieu donné, il est ainsi possible de déterminer la distance d'un objet selon le temps différenciant l'émission et la réception du signal. C'est la phase de calcul. La distance d'un objet avec le radar est calculée à l'aide de l'équation (3.1) où d est la distance, c la célérité de la lumière dans le vide, n l'indice de réfraction de la lumière dans le milieu, et t le temps entre l'émission et la réception du signal. Cependant, un autre effet rentre en jeu dans le fonctionnement des radars, effet qui peut être exploité pour déterminer précisément la vitesse des objets détectés. Cela permet aussi de différencier des petites cibles positionnées à proximité de cibles plus importantes, dont les signaux de retours pourraient être confondus. Cet effet, c'est l'effet Doppler.

$$d = \frac{c.n.t}{2} \quad (3.1)$$

L'EFFET DOPPLER

L'effet Doppler-Fizeau, ou plus simplement l'effet Doppler, est le phénomène physique décrivant le décalage de la fréquence d'une onde réfléchie après émission lorsque l'objet à l'origine du phénomène de réflexion est en mouvement sur la direction radar-objet (Doppler, 1842). En effet, si la distance d'un objet varie pendant le cycle d'émission-réception d'un radar, alors la fréquence de l'onde reçue obéit à l'équation (3.2) où f_e est la fréquence de l'onde émise, f_r la fréquence de l'onde reçue, c la célérité de la lumière dans le vide, n l'indice de réfraction de la lumière dans le milieu et v la vitesse de la cible. On peut alors facilement calculer la variation de fréquence f_s causée par l'effet doppler selon l'équation (3.3). Un renversement de cette formule en connaissant l'écart de fréquence permet de déduire facilement la vitesse de l'objet réfléchissant.

$$f_r = \left(\frac{c.n + v}{c.n - v} \right) . f_e \quad (3.2)$$

$$f_s = f_r - f_e = \frac{2.v.f_e}{c.n - v} \quad (3.3)$$

Bien qu'exploité pour les systèmes radars, l'effet Doppler se manifeste à des occasions plus démonstratives dans la vie de tous les jours. Par exemple, on peut

fréquemment entendre le changement de tonalité d'une sirène d'ambulance se déplaçant par rapport à nous. Lorsque l'ambulance se rapproche, la fréquence augmente et donc la hauteur tonale augmente. Une fois que l'ambulance nous croise et s'éloigne, la hauteur tonale diminue. Ce phénomène peut être facilement représenté sous forme visuelle, comme le montre la Figure 3.1. Par analogie, on observe que lorsque l'objet se déplace vers le radar, alors la fréquence des ondes reçues augmente. A l'inverse, lorsqu'il se déplace dans le sens opposé, la fréquence diminue. En pratique, lorsqu'un objet se déplace parfaitement perpendiculairement avec la droite de direction radar-objet, l'effet Doppler n'est pas observé. Ainsi, un objet se déplaçant selon un cercle parfait dont le radar serait le centre ne serait pas détecté comme un objet mouvant en prenant uniquement en compte l'effet Doppler. En revanche, l'onde reçue formerait alors un angle avec l'onde émise qui pourrait être utilisé pour en inférer une vitesse angulaire.

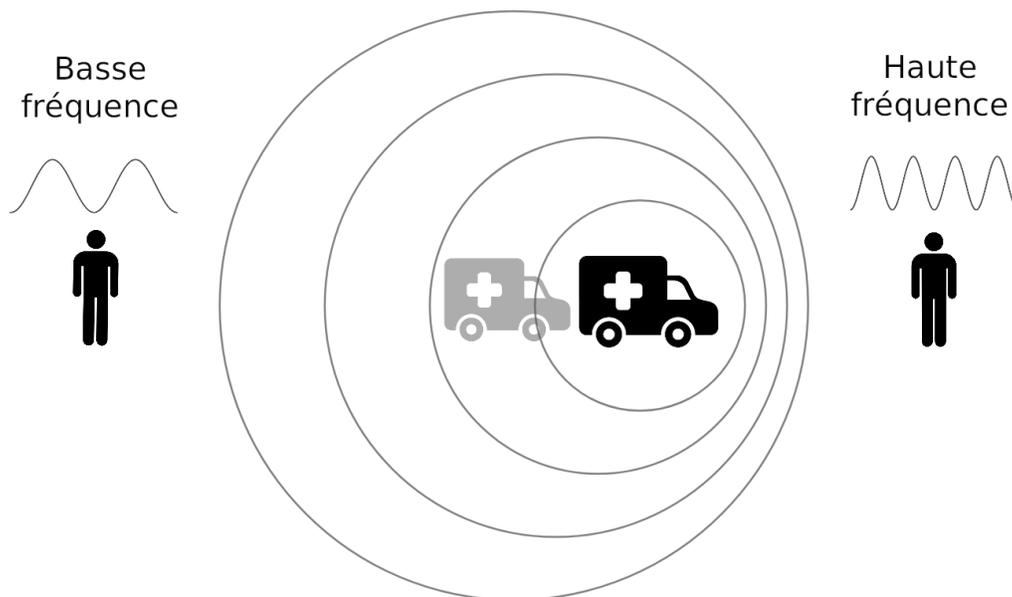


Figure 3.1 : Illustration de l'effet Doppler

LES RADARS À ONDES CONTINUES

Il est possible de diviser les systèmes radars en deux catégories : les radars à ondes continues et les radars à ondes pulsées. Les premiers se caractérisent par l'émission d'une onde radio de manière continue. La réception se fait donc de manière continue également, ainsi deux antennes sont généralement utilisées. Ce type de radar ne peut que mesurer la vitesse d'un objet via l'effet Doppler, et ne peut pas inférer la distance. En effet, il est impossible de déduire le décalage temporel entre une onde émise et une onde reçue, car les deux se font en continu. Lorsque l'onde du radar est modulée, c'est-à-dire quand sa fréquence évolue avec le temps, alors il est possible de mesurer la distance d'un objet malgré l'aspect continu de l'émission. Il suffit d'observer le décalage temporel entre la fréquence de l'onde reçue et l'émission antérieure d'une onde de même fréquence. Lors de la modulation, en revanche, on ne peut déterminer la vitesse d'un objet via l'effet Doppler. Pour obtenir des informations sur la distance et la vitesse, il faut alors alterner entre des détections à ondes non-modulées et des détections à ondes modulées. Les radars à ondes continues se distinguent par leur simplicité et leurs prix compétitifs. Leur fonctionnement permet également, pour la même puissance, de détecter des objets plus éloignés du système que les radars à ondes pulsées. Les radars à ondes continues peuvent alors être utilisés comme des alternatives économiques pour faire de la détection de distance longue portée à faible coût énergétique.

3.1.2 LES RADARS ULTRA-WIDEBAND

LES RADARS À ONDES PULSÉES

Les radars à ondes pulsées (ou radars Doppler pulsés) fonctionnent, par opposition aux radars à ondes continues, par impulsions périodiques d'ondes électromagnétiques. Ce fonctionnement permet à ces radars de pouvoir détecter à la fois la distance d'un objet par la mesure d'écart temporel, et sa composante radiale de vitesse grâce à l'effet Doppler. Le réglage de la fréquence d'émission impacte directement la portée du radar et la vitesse maximale pouvant être détectée. Ainsi, augmenter la fréquence des impulsions va augmenter la vitesse maximale détectable mais diminuer la portée, et vice-versa. Un compromis doit être trouvé au cas par cas selon l'utilisation souhaitée.

La liaison entre la fréquence des impulsions et la portée du radar s'explique assez simplement. Un objet se trouvant à une distance telle du radar qu'il faudrait, pour l'onde émise, plus de la moitié de la fréquence d'émission pour l'atteindre, entraînerait forcément une ambiguïté lors de la réception du signal. En effet, si la fréquence des impulsions est f , et qu'il faut, par exemple, $\frac{2}{3f}$ secondes pour que l'onde atteigne l'objet, alors il faudra $\frac{4}{3f}$ secondes pour que l'onde fasse l'aller-retour entre l'émetteur et le récepteur du radar. Or, entre temps, une nouvelle impulsion sera émise. Comment distinguer alors si l'onde reçue à $\frac{4}{3f}$ n'est pas en réalité la réflexion de la seconde impulsion par un objet situé de telle manière qu'il suffit de $\frac{1}{3f}$ secondes pour faire l'aller-retour ? Ainsi, plus la fréquence d'impulsion augmente, plus la portée du radar diminue.

LES RADARS ULTRA-WIDEBAND

Les radars ultra-wideband, pour ultra large bande, sont des radars à ondes pulsées, à courte portée et peu énergivores (Taylor, 1994). Ils communiquent avec une amplitude plus faible qu'un radar à ondes pulsées classique, mais sur une bande de fréquence plus large. Cette bande fréquentielle s'étend sur plus de 500MHz ou 20% de la fréquence centrale de la bande. Les impulsions émises sont de l'ordre de la nano ou de la pico seconde, et sont envoyées à des fréquences variées selon l'utilisation souhaitée. La fréquence des impulsions a tendance à être plutôt faible pour les utilisations en lien avec la détection de distance et l'imagerie. L'ordre de grandeur moyen avoisine la dizaine ou la centaine de Hertz. En revanche, pour les applications dans la communication à très grande vitesse et très petite portée, la fréquence est de l'ordre du GigaHertz. Contrairement aux émetteurs d'ondes radio traditionnels qui modulent leur signal sur la fréquence ou la phase de l'onde émise, les émetteurs ultra-wideband modulent leur signal temporellement. Cela permet de déterminer le temps de trajet de l'onde avec une très bonne résolution et donc de mesurer la distance d'un objet très précisément. De plus, la faible densité spectrale de l'onde émise rend le signal imperceptible pour les récepteurs radio classiques, et immunise partiellement les émetteurs ultra-wideband aux interférences.

Les radars UWB présentent différents avantages face aux radars classiques. En premier lieu, la détection de la distance d'une cible est plus précise. En effet, la portée du radar étant plus faible, sa résolution s'en retrouve accrue. De plus, cette résolution rend possible la détection de mouvements d'éléments considérés comme des sous-

ensembles de la cible, en plus des mouvements généraux de la cible elle-même. Par exemple, les mouvements de membres spécifiques d'une cible humaine peuvent être perçus. L'étalement du signal sur la bande fréquentielle rend les radars ultra-wideband moins vulnérables au bruit et aux interférences. Ils sont également moins impactés par les émissions de radars à bande fréquentielle courte et aux autres signaux apparentés. Enfin, la densité du signal étant plus faible, ils sont moins facilement discernables parmi le bruit pour d'autres types de radars. Ainsi, ils présentent des propriétés de camouflage intéressantes.

La technologie ultra-wideband trouve des applications dans de nombreux domaines. Parmi eux, la transmission de données sans fil à courte portée et à très haut débit. Elle peut ainsi être utilisée pour faire communiquer des périphériques informatiques nécessitant une bande passante importante comme des caméras ou des imprimantes sans connexion filaire. On retrouve également cette technologie pour le transfert de fichiers ou d'informations à très haut débit entre différents ordinateurs et micro-ordinateurs comme une alternative économe en énergie aux différents protocoles actuellement populaires. Ainsi, des émetteurs/récepteurs ultra-wideband ont récemment été ajoutés dans certains téléphones intelligents. De plus, ils sont aussi utilisés pour des applications médicales avec la surveillance de paramètres vitaux comme le pouls ou la fréquence respiratoire. Enfin, ils sont utilisés pour de la détection de cibles dans l'espace et/ou de leurs vitesses pour de nombreuses applications allant du militaire à l'aide aux personnes vulnérables, comme c'est le cas dans la problématique de ce mémoire.

3.2 APPLICATIONS EXISTANTES DES RADARS UWB

Les recherches liées aux radars ultra-wideband dans la littérature scientifique sont prolifiques et s'étendent sur de nombreux domaines. On retrouve des publications s'intéressant à leurs propriétés fondamentales et leurs structures physiques, mais aussi des articles faisant état de leurs performances pour de nombreuses applications concrètes. La section suivante fait état de l'état de l'art des radars UWB appliqués à la détection des humains et de leurs mouvements ayant un lien avec le domaine de la détection de cibles et de l'aide aux personnes. Seuls les travaux ayant été publiés entre 2015 et 2020 sont retenus, à la condition qu'ils aient une problématique liée au domaine de la santé, de l'assistance aux personnes vulnérables, ou de la reconnaissance de mouvements ou d'activités des humains. Dans un souci de synthèse, ne seront revus que les travaux les plus pertinents permettant soit de représenter le panel des applications des radars UWB, soit ayant un lien direct avec notre problématique. Ils seront divisés dans les quatre catégories suivantes en fonction de leurs thèmes : estimation du pouls et de la respiration, imagerie, détection de position ou d'occupation, et détection de chutes ou d'activités.

3.2.1 ESTIMATION DU POULS ET DE LA RESPIRATION

L'estimation du pouls de manière précise et continue est un enjeu vital du milieu hospitalier et du domaine médical en général. Il existe plusieurs techniques pour effectuer un suivi précis et fiable de cet indice, notamment l'électrocardiographie permettant de mesurer le pouls d'un être vivant via le placement d'électrodes à même le corps. Cependant, la mesure de cette valeur à distance pourrait permettre de réduire la gêne

subie par le patient causée par la présence d'électrodes sur sa peau. De plus, cette mesure peut être effectuée à travers les vêtements, et sans manipulation sur le corps du patient. L'extrêmement faible portée du mouvement d'un battement de cœur comparée à celle d'un mouvement respiratoire rends sa mesure à distance complexe. Là où les radars continus sont difficilement à même de réaliser cette tâche à cause du bruit présent, les radars UWB sont efficaces. Cependant, ils peuvent avoir des difficultés à distinguer le pouls des mouvements respiratoires avec précision. C'est pourquoi ce sont des radars UWB à bande passante relativement restreinte, à la limite de leur définition, qui sont utilisés. Ils offrent un compromis intéressant entre la précision des radars à ondes continues et la réduction de l'impact du bruit des radars UWB.

La mesure de la dérivée du pouls instantanée, proposée par Sakamoto *et al.* (2016a), utilise une approche basée sur l'extraction de caractéristiques à partir des données émises par le radar. Six caractéristiques sont isolées : les maximas locaux, les minimas locaux, les dérivées croissantes et décroissantes des maximas locaux, et les dérivées croissantes et décroissantes des minimas locaux. Une corrélation afin de trouver les images les plus proches partageant les mêmes caractéristiques permet de calculer le pouls instantané puis sa dérivée. Ces calculs sont nettement plus rapides qu'une auto-corrélation utilisant la transformation de Fourier rapide (Sakamoto *et al.*, 2016a). La publication introduit également la notion de topologie de caractéristiques, qui permet d'utiliser la série temporelle des caractéristiques extraites plutôt que la série temporelle des images du radar. Comparée avec les résultats obtenus via l'électrocardiographie, cette méthode s'est avérée efficace et viable pour la détection distante du pouls.

Naishadham *et al.* (2016) propose une approche par la modélisation de variables d'état paramétriques à partir des images renvoyées par le radar. Cette méthode a pour objectif de passer outre les problèmes d'intermodulation entre le pouls et les mouvements respiratoires qui impactent les performances d'une extraction du pouls par une transformation de Fourier rapide. Elle montre des performances intéressantes et une forte capacité à ignorer le bruit produit par les mouvements respiratoires.

L'approche de Sakamoto *et al.* (2015b) consiste à filtrer la composante liée au mouvement respiratoire du signal à l'aide d'un filtre passe-bas gaussien de fréquence limite 7,7Hz. Après soustraction de cette composante au signal brut, on obtient un signal ne possédant que les composantes hautes-fréquences et correspondant au pouls. Un algorithme basé sur la topologie du signal est ensuite appliqué pour extraire le pouls, avec une corrélation de 0,993 avec l'électrocardiogramme témoin. La même équipe de recherche étend son travail (Sakamoto *et al.*, 2015a) en utilisant la même approche mais sur un radar placé face à la plante des pieds d'un patient. En effet, les autres méthodes proposent des protocoles où le radar est placé face au torse du patient, ce qui reviendrait à placer l'équipement au plafond pour mesurer le pouls d'un patient allongé. La corrélation avec l'électrocardiogramme témoin est de 0,93, soit légèrement inférieure à la première publication. Néanmoins, cette contribution permet de montrer la faisabilité de détecter le pouls avec un radar UWB positionné dans le prolongement du corps du sujet.

Parfois gênants lorsque l'on souhaite détecter le pouls, les mouvements respiratoires sont également le sujet de publications scientifiques en lien avec les radars UWB.

Pallesen *et al.* (2018) propose une méthode de détection des réponses respiratoires dans des conditions adverses, comme celles que l'on pourrait trouver dans des ruines ou des bâtiments dégradés lors d'une situation de recherche et sauvetage en milieu urbain.

Cette détection de mouvements respiratoires peut aussi se présenter comme une alternative attrayante à la polysomnographie pour détecter les phases de sommeil et d'éveil d'un individu (Wu *et al.*, 2016). Une transformation de Fourier rapide est effectuée sur le signal, puis un algorithme de détection du sommeil est appliqué sur les motifs extraits de la transformation de Fourier afin de discriminer un individu endormi d'un individu éveillé. Les résultats semblent prometteurs pour de futures applications dans ce domaine, à la condition d'un équipement plus précis et de données empiriques plus nombreuses pour valider la méthode.

3.2.2 IMAGERIE

L'imagerie est également un domaine dans lequel les radars ultra-wideband sont étudiés, avec divers angles d'applications comme l'imagerie médicale ou l'imagerie appliquée au secteur de la sécurité. Par exemple, les radars sont largement utilisés comme une alternative rapide, efficace et économique pour la détection d'armes à feu. Le travail de la publication de Sakamoto *et al.* (2016c) permet de répondre aux limitations des radars dans ce domaine, où l'imagerie d'un individu peut être impactée par son mouvement. Afin d'éviter le flou qui pourrait apparaître, une méthode de mise au point est proposée. La vitesse de l'individu est estimée et plusieurs images sont réalisées, une par vitesse probable. Un algorithme de calcul de flou permet alors de sélectionner

l'image la plus nette. Cette méthode n'est efficace que grâce à la rapidité de l'algorithme d'imagerie utilisé précédemment. La méthode d'imagerie proposée présente des résultats très proches d'une méthode d'imagerie par migration F-K classique, avec un temps de calcul bien moins élevé (Sakamoto *et al.*, 2016c). L'estimation de la vitesse de l'individu est détectée avec une marge d'erreur de 4%, permettant de choisir efficacement l'image la plus nette de la cible.

D'autres méthodes d'imageries sont présentes dans la littérature, comme la combinaison de la migration de Kirchhoff et de la migration F-K proposée dans la contribution de Sakamoto *et al.* (2015). Cette méthode a l'avantage de produire des images à une vitesse 250 fois plus importante que la méthode "delay-and-sum" utilisée comme outil de comparaison. La netteté des images est également supérieure à cette méthode et à la migration de Kirchhoff seule.

3.2.3 DÉTECTION DE POSITION ET D'OCCUPATION

Hormis la détection du pouls ou de la fréquence respiratoire, un des enjeux les plus étudiés est celui de la détection d'occupation, de distance ou de position. En effet, la propriété passe-mur des radars ultra-wideband les rend particulièrement propices à ce genre d'exercice et peut donner lieu à des applications dans de nombreux domaines (Muqaibel *et al.*, 2005).

Par exemple, l'utilisation des radars UWB peut être conjointe à un modèle de détection d'occupation à des buts d'économie d'énergie. C'est la méthode choisie par la contribution de Brown *et al.* (2017) qui utilise cette approche en combinaison avec un

arbre de décision pour apprendre les habitudes d'occupation de pièces d'un foyer. Cela permet, *in fine*, d'optimiser l'empreinte énergétique de l'habitat. La combinaison de ces travaux avec d'autres capteurs embarqués dans l'électroménager de l'habitat laisse présager des résultats concluants pour réduire l'écart entre la consommation d'énergie optimale d'un foyer et sa consommation effective.

D'autres approches comme celle de l'analyse en composantes principales permettent de détecter avec fiabilité l'occupation d'une pièce et le nombre de personnes présentes (Baird *et al.*, 2017). Un algorithme calcule une valeur d'énergie d'une matrice correspondant à une série d'images radars. L'occupation et le nombre d'occupants sont déterminés en fonction de cette valeur énergétique comparée à différents seuils. Une précision de 88,1% est obtenue à l'issue des tests, bien que des données supplémentaires pourraient être nécessaires pour conclure de manière tranchée sur l'efficacité de la méthode. Néanmoins, elle ne permet pas de séparer deux personnes proches, et donc de détecter le bon nombre d'occupants le cas échéant.

En effet, la détection d'une multitude d'individus au sein d'un même lieu est également une problématique clé de la détection de position car la distinction de ces derniers est nécessaire pour tout foyer composé de plus d'une personne. C'est à cette problématique que répond la publication de Choi *et al.* (2017) en proposant un algorithme de détection multi-humain, là où de nombreuses publications n'étudient la détection que d'une seule personne. La complexité de ce problème repose en la discrimination des pics de détection causés par la présence d'humains à ceux causés par le bruit avoisinant ou les réflexions du signal. Le calcul du seuil d'intensité distinguant ces deux types

de pics est donc l'enjeu majeur de la solution proposée. Il est dépendant de plusieurs variables statistiques établies empiriquement au cours de l'étude. La méthode a été testée sur trois sujets en mouvement, et sur deux sujets statiques proches. Les résultats sont encourageants et montrent la possibilité de détection de plusieurs humains dans une pièce à l'aide d'un unique radar, et de les différencier.

À mi-chemin entre la détection de position et la détection de pouls, la contribution de Liang *et al.* (2017) propose une technique de détection des signes vitaux et de leurs distances à travers des obstacles. Les mouvements respiratoires et cardiaques détectés permettent de distinguer un humain parmi du bruit, par exemple dans le cadre d'une mission de recherche et de secourisme en milieu urbain. La distance de l'individu est calculée à l'aide d'une transformée de Fourier à temps court, et sa fréquence cardiaque à l'aide d'une démodulation arctangente puis d'une accumulation fréquentielle multiple. Les résultats expérimentaux et la comparaison de la méthode proposée avec d'autres méthodes standards prouvent l'efficacité de la solution et sa pertinence dans le cadre des missions de recherche et de secourisme en milieu urbain.

Enfin, la mesure de vitesse étant l'une des applications les plus connues des radars continus, on retrouve également des projets de recherche connexes utilisant les radars UWB. La contribution de Sakamoto *et al.* (2016b) s'attaque au problème des vitesses supérieures au seuil de Nyquist, c'est-à-dire deux fois la fréquence d'émission des pulsations du radar. En réponse au problème propre aux radars à ondes pulsées évoqué dans la section précédente, les auteurs proposent une méthode de traitement du signal applicable pour tout radar UWB à ondes pulsées sans altération matérielle.

Cette méthode utilise la "texture" du signal reçu, c'est-à-dire son enveloppe, pour faire disparaître les ambiguïtés présentes au-delà du seuil de Nyquist. Des tests sur des vitesses de rotation et de translation élevées, contrastés de comparaisons avec les méthodes existantes, montrent l'efficacité de la solution proposée. En revanche, elle reste assez sensible au bruit malgré tout, et laisse place à des perspectives d'améliorations.

3.2.4 DÉTECTION DE CHUTES ET D'ACTIVITÉS

La reconnaissance d'activités de la vie quotidienne et la détection de chutes font également partie des sujets de recherche mettant en scène les radars ultra-wideband. Cette branche de recherche semble assez récente et peu de chercheurs se concentrent sur la reconnaissance d'activités en tant que tel. Plusieurs contributions ont néanmoins étudié la problématique de la reconnaissance de chutes, avec plusieurs méthodes d'apprentissage machine différentes.

L'article de (Diraco *et al.*, 2018) se penche sur la question de la détection des chutes à l'aide de radars UWB en utilisant la méthode qui suit. Les données des radars passent parmi plusieurs filtres pour discriminer l'information utile du bruit. Les signaux traités sont alors projetés sur des fenêtres d'une seconde et demie. Des caractéristiques sont extraites pour mettre en évidence les variations légères entre différents mouvements qui sont distinguables grâce à l'effet Doppler détaillé dans la section précédente. La classification s'effectue à l'aide d'un vote majoritaire sur N classeurs des K plus proches voisins mono-classe, où N est le nombre de classes à discriminer. L'apprentissage pour cette méthode est donc non-paramétrique. Une comparaison de cette méthode face

à d'autres techniques d'extraction de caractéristiques et d'une classification à l'aide des machines à vecteurs de support indique des scores de sensibilité et de spécificité⁵ de 91% et 89% respectivement, face à 86% et 75% pour la méthode servant d'outil de comparaison. La méthode proposée montre ainsi des résultats intéressants et une grande propriété de généralisation en conditions réelles grâce à l'apprentissage non-paramétrique du classeur de chutes.

Les réseaux de neurones convolutifs (LeCun *et al.*, 1995) peuvent également être utilisés pour la détection de chutes à l'aide des radars UWB (Sadreazami *et al.*, 2019). Une approche temps-rapide / temps-lent est proposée via l'addition des images temps-rapide sur une fenêtre temporelle restreinte pour obtenir la composante temps-lent. Les caractéristiques ne sont pas extraites en amont via un pré-traitement mais sont directement inférées et automatiquement apprises par le réseau de neurones convolutif. Ainsi, le réseau peut détecter la dépendance d'une image en fonction de ses images voisines, ce qui introduit alors la notion de contexte temporel dans la classification. La méthode proposée a été comparée à différentes méthodes, parmi lesquelles différentes variantes du perceptron multicouche, la méthode des K voisins les plus proches, et un réseau de neurones à mémoire court-terme / long-terme (Long Short Term Memory (LSTM) (Hochreiter et Schmidhuber, 1997)). Elle affiche un gain significatif de précision, tout en possédant à la fois une sensibilité de 100% et une spécificité de 91,67%. Les propriétés de généralisation qui en découlent rendent la solution présentée particulièrement intéressante et efficace dans le contexte de la reconnaissance d'activités en conditions

5. La sensibilité est la capacité de détection (taux de vrai-positifs), et la spécificité est la capacité de discrimination (taux de vrai-négatifs)

réelles.

La structure même des réseaux de neurones à mémoire court-terme / long-terme leur permet de prendre en compte les interdépendances des séries temporelles. Ce genre de réseaux de neurones est donc particulièrement propice pour la reconnaissance d'activités et la détection de chutes. La publication de (Sadreazami *et al.*, 2018) s'attaque à ce dernier problème en proposant une méthode utilisant ces réseaux de neurones pour extraire automatiquement les caractéristiques des variations Doppler provoquées par les mouvements corporels. De la même manière que précédemment, après avoir été filtrées, une addition des images temps-rapide sur une fenêtre temporelle restreinte permet d'obtenir des composantes temps-lent. La combinaison de ces deux types d'images est alors utilisée comme entrée d'un empilement de réseaux de neurones LSTM pour une extraction automatique de caractéristiques. Cette approche permet d'obtenir des caractéristiques plus riches et complètes pour augmenter les performances du modèle. Comparée à différentes méthodes d'apprentissage machine dans lesquelles des caractéristiques standards sont manuellement passées en entrée, la méthode proposée obtient une précision de 89,80%, soit un gain de plus de 6% par rapport aux autres méthodes.⁶. Ainsi, elle se présente comme une solution particulièrement efficace et pourrait potentiellement être étendue, au-delà de la classification binaire chute / absence de chutes, à de la classification d'activités de la vie quotidienne.

6. Ici, "précision" réfère à l'indice *accuracy* en anglais.

CHAPITRE IV

ARCHITECTURE DU PROJET

Ce projet de recherche a nécessité la création d'un environnement matériel et logiciel pour utiliser les radars ultra-wideband, récupérer leurs signaux, les stocker, les traiter, et afficher des éléments de suivi statistique sur les activités effectuées. Ce chapitre vise à détailler les architectures matérielles et logicielles utilisées dans le projet, ainsi que les divers outils développés. Il sera conclu par une série de tests empiriques ayant été réalisés pour se familiariser avec les radars et la partie logicielle développée par l'École de Technologie Supérieure (ETS) de Montréal.

4.1 ARCHITECTURE MATÉRIELLE

Les éléments centraux du projet sont les trois radars ultra-wideband utilisés pour acquérir des données dans l'habitat intelligent. Ces trois radars sont des radars UWB XeThru X4M200 conçus par Novelda pour détecter des patrons de respiration et des mouvements sur des personnes n'étant pas nécessairement en train de se déplacer⁷. Ils sont décrits par le constructeur comme capables de détecter la présence d'une personne sur une distance de 5m, même pour une longue période d'immobilité. Ils permettent également de détecter la distance qui sépare la personne en question du capteur. Il est possible d'obtenir cette distance sur une portée allant jusqu'à 9,8m dans un cône d'un angle obtus avoisinant l'angle plat. Le champ de détection est représenté de manière

7. Fiche technique : https://www.codico.com/shop/media/datasheets/Novelda_X4M200_respiration_sensor_rev_c_preliminary.pdf

approximative en Figure 4.1. La portée effective mesurée lors des expérimentations est de 40cm à 9,8m. Plus de données empiriques sur le champ de détection du radar sont présentées en fin de chapitre. La fréquence des ondes émises est comprise entre 6GHz et 10,2GHz, et la fréquence d'acquisition des images est paramétrée à 50Hz. Enfin, la résolution sur la distance détectée est de 5,14cm.

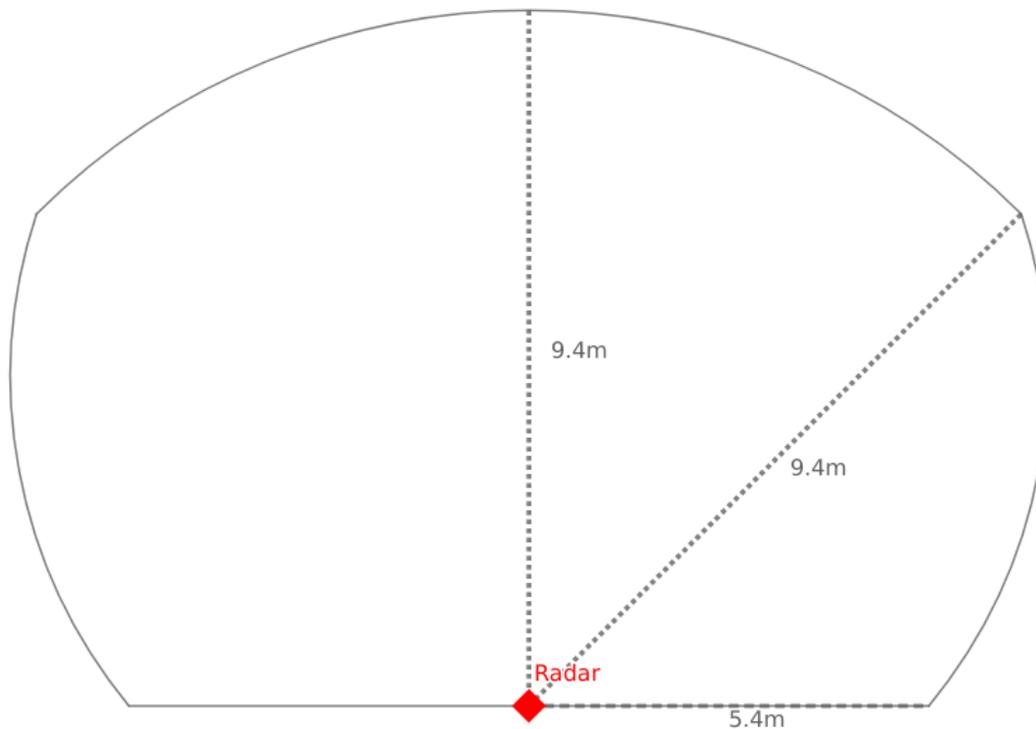


Figure 4.1 : Champ de détection frontal du radar ultra-wideband schématisé à partir des informations fournies par sa fiche technique

Les trois radars UWB sont branchés sur une Raspberry Pi 3 B+⁸ via des câbles USB. La Raspberry Pi est connectée sur le réseau privé du laboratoire auquel est

8. <https://static.raspberrypi.org/files/product-briefs/>

Raspberry-Pi-Model-Bplus-Product-Brief.pdf

également connecté le système central. Le système central est un ordinateur sur lequel fonctionne soit le logiciel de capture lors de la création des jeux de données, soit le logiciel de traitement (statistiques, pré-traitement des données pour la classification, etc.) lors de la phase de test en temps-réel. Dans ce dernier cas, un ou plusieurs appareils, présents dans un même réseau que le système central, peuvent être utilisés afin d'afficher une interface de visualisation en direct. Dans le cadre de la preuve de concept, le système central était un ordinateur portable Dell XPS 9560⁹, qui faisait également office d'outil de visualisation lors des phases de test en utilisant l'interface réseau de bouclage interne¹⁰. Cette architecture matérielle est représentée en Figure 4.2.

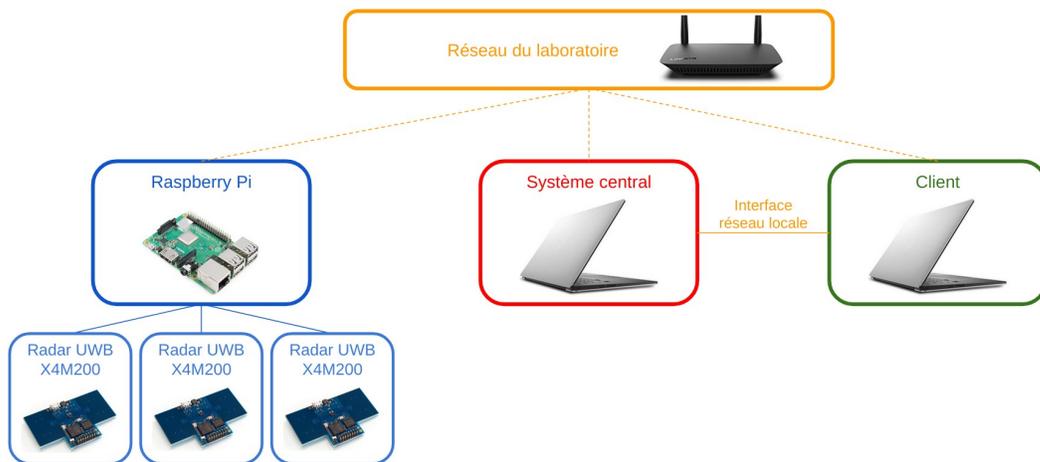


Figure 4.2 : Architecture matérielle du projet

9. https://topics-cdn.dell.com/pdf/xps-15-9560-laptop_specifications_en-us.pdf

10. Adresse 127.0.0.1 qui transmet les paquets sortants en temps que paquets entrants, utile pour tester des configurations réseaux

4.2 ARCHITECTURE LOGICIELLE

L'architecture logicielle du projet est assez semblable à la structure de l'architecture matérielle. On retrouve donc trois applications majeures distinctes, chacune présente sur l'un des appareils : le logiciel d'acquisition, le logiciel central et le logiciel de visualisation. Un schéma illustrant les différentes briques logicielles et leurs liens est présent en Figure 4.3.

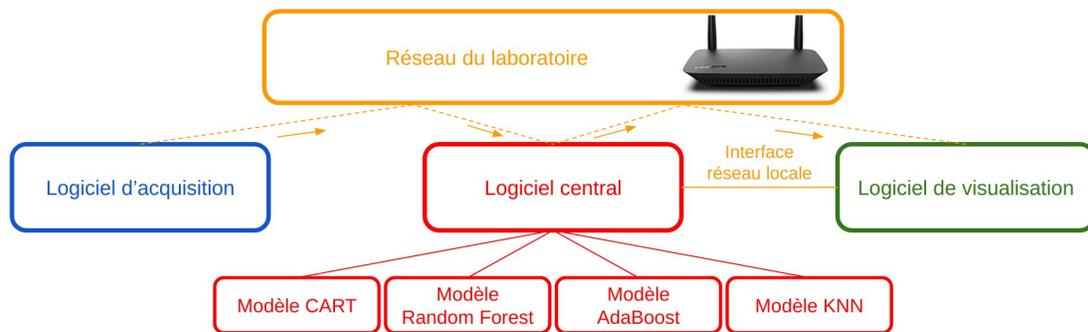


Figure 4.3 : Architecture logicielle du projet

4.2.1 LOGICIEL D'ACQUISITION

Le logiciel d'acquisition est le logiciel embarqué sur la Raspberry Pi permettant l'acquisition des données des radars, leur traitement et leur envoi sur le réseau. Ce logiciel est une extension du travail effectué par l'École de Technologie Supérieure (ETS) de Montréal dans un projet de recherche connexe. Le programme de l'ETS permet l'acquisition des données des radars et le calcul de la distance d'un objet d'intérêt pour chacun de ceux-ci. Ensuite, une trilatération (Murphy et Hereman, 1995) est réalisée à partir des

distances calculées et des positions relatives des radars dans l'espace. La position inférée est alors affichée sur un outil de visualisation représentant l'environnement couvert par les radars.

Cette application n'étant pas développée avec l'optique d'être embarquée sur une Raspberry Pi, plusieurs ajustements ont été nécessaires. La visualisation des résultats a été supprimée, car coûteuse en ressources et redondante avec l'application de visualisation telle qu'imaginée dans la structure du projet. Afin de permettre au logiciel de transmettre les données recueillies et calculées, une brique de communication réseau a été rajoutée. Elle utilise le protocole de communication "*WebSocket*" à travers la bibliothèque "*websocketpp*" développée pour le langage de programmation C++¹¹. Un fil d'exécution séparé est utilisé afin d'éviter tout blocage de l'acquisition en cas de congestion réseau. Ainsi, les données des capteurs peuvent être récupérées et stockées localement même si le fil d'exécution réseau est bloqué à cause d'une erreur. Enfin, certaines adaptations légères ont dû être apportées afin de rendre compatible l'application initiale développée pour le système d'exploitation Windows avec le système d'exploitation Linux embarqué sur la Raspberry Pi. Par exemple, de nouvelles bibliothèques ont dû être importées et les portions de codes spécifiques au système d'exploitation ont dû être généralisées ou ré-écrites pour s'adapter au système d'exploitation Linux.

11. <https://github.com/zaphoyd/websocketpp>

4.2.2 LOGICIEL CENTRAL

Le logiciel situé sur le système central est en réalité composé de plusieurs briques autonomes. Une application centrale permet de gérer les flux de données, de les ordonner, et de piloter le comportement de l'ensemble de l'architecture logicielle. Elle reçoit les données du logiciel d'acquisition, effectue une partie du traitement statistique (temps passé par activité, nombre de repas, temps de sommeil, etc.), envoie les données aux autres briques logicielles présentes sur le système central, et envoie les données à afficher sur le logiciel de visualisation. Les données des radars peuvent être récupérées en temps-réel depuis le logiciel d'acquisition, mais il est aussi possible de lire des fichiers contenant des données enregistrées au préalable depuis l'application centrale. Cela permet d'utiliser des données récoltées précédemment pour générer des statistiques et ainsi conserver une forme d'historique, mais aussi de tester les différentes briques logicielles rapidement sans nécessiter le déploiement du logiciel d'acquisition. L'application centrale est programmée en C++, à l'instar de l'application d'acquisition. Elle utilise donc la même bibliothèque que l'application d'acquisition pour implémenter les *WebSockets*.

Les autres applications présentes sur le système central sont les programmes mettant en exécution les modèles d'intelligence artificielle entraînés. C'est le système central qui choisit et commande l'exécution de ces applications externes, puis qui envoie les données à classer au modèle. Celui-ci envoie alors ses prédictions à l'application centrale. Les programmes de classification sont programmés en Python. Ainsi, elles

utilisent la bibliothèque “websockets” pour communiquer avec l’application centrale ¹².

4.2.3 LOGICIEL DE VISUALISATION

Enfin, une application web peut être déployée sur n’importe quelle machine compatible et partageant un réseau avec le système central. La compatibilité n’a pas été extensivement testée, mais l’application web n’utilisant que des outils standards, elle fonctionne sur la plupart des navigateurs web maintenus à la date de publication du présent document. Cette application sert d’outil de visualisation de la position de la personne détectée dans l’habitat intelligent. Elle affiche aussi diverses statistiques en lien avec les activités détectées, et permet ainsi de garder un suivi des cycles de sommeil, des repas, ou des événements graves comme les chutes. Une capture d’écran de cette interface est illustrée sur la Figure 4.4.

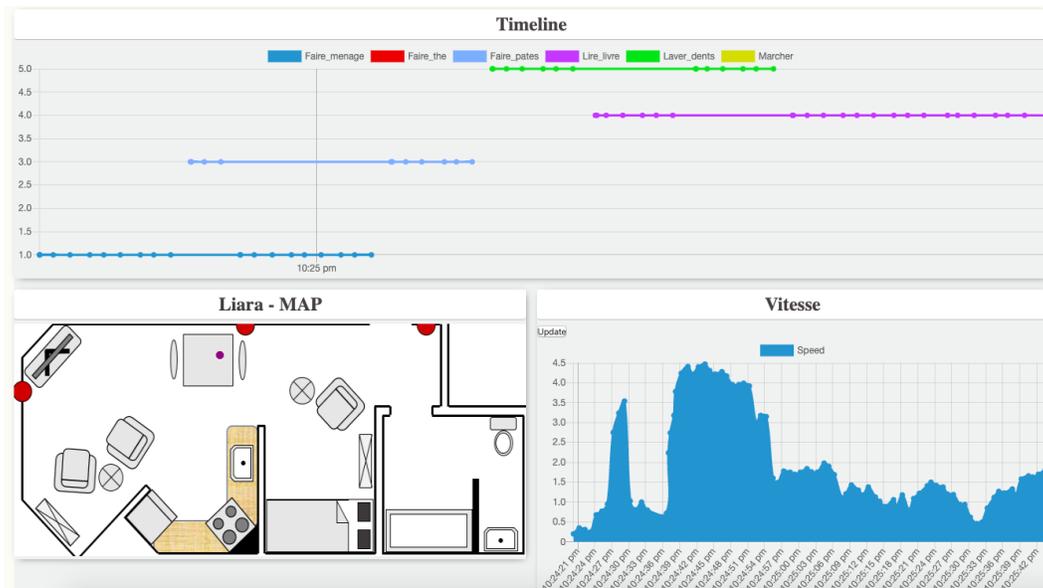


Figure 4.4 : Capture d’écran de l’interface de visualisation web

12. <https://websockets.readthedocs.io/en/stable/intro.html>

4.3 TESTS DE PRÉCISION DES CAPTEURS

Afin de se familiariser avec les radars UWB et l'application développée par l'ETS, une série de tests pour déterminer la précision et la portée effective des radars a été réalisée. L'objectif de ces tests n'est pas de déterminer rigoureusement la précision objective des radars ou la performance de l'algorithme de distance de l'ETS, mais plutôt d'obtenir une idée générale de la qualité des données qui seraient obtenues pour le jeu de données. En effet, la présence de la brique logicielle de l'ETS ne permet pas une analyse qualitative fiable des radars, et le développement continu de leur application au moment des tests rendent peu pertinente une évaluation rigoureuse de l'algorithme. Ces tests permettent néanmoins de mettre en lumière quelques limitations importantes vis-à-vis des données qui sont à prendre en compte pour estimer la qualité des jeux de données récoltés, principalement sur les distances et les positions calculées en complément des données brutes.

4.3.1 TESTS DE PRÉCISION CIRCULAIRE

Afin de déterminer si la précision du radar varie en fonction de la position angulaire de la personne par rapport à l'axe central du champ de détection du radar, des tests circulaires ont été réalisés. Des marqueurs ont été placés le long de demi-cercles de différents rayons, selon l'angle entre la droite sujet-radar et la droite centrale du champ de détection. La Figure 4.5 illustre les marqueurs en question. Le radar UWB n'étant pas conçu pour détecter efficacement des personnes à l'opposé de son champ d'émission, seul le demi-cercle face au radar fut étudié. Des demi-cercles de 0,5m ; 1m ; 1,5m ; 2m ;

2,5m ; 3m et 3,5m de rayons ont été réalisés. Sur les cinq cercles de plus petit rayons, des marqueurs aux positions d'angle -90° , -60° , -45° , -30° , 0° , 30° , 45° , 60° et 90° ont été placés. La largeur de la salle ne permettant pas de d'obtenir des demi-cercles complets de plus de trois mètres de rayons, seuls les angles -60° , -45° , -30° , 0° , 30° , 45° , 60° ont été marqués pour les deux autres demi-cercles. Pour chaque acquisition, le sujet se déplace depuis la position -90° vers la position 90° en suivant le tracé du demi-cercle de la manière la plus rigoureuse et à la vitesse la plus régulière possible. Chaque passage sur les marqueurs d'angles est ponctué d'un petit temps d'arrêt et d'un relevé temporel pour pouvoir synchroniser les données acquises avec le trajet effectué. Chaque acquisition est effectuée deux fois afin de compenser d'éventuelles erreurs d'acquisition ou de relevé temporel.

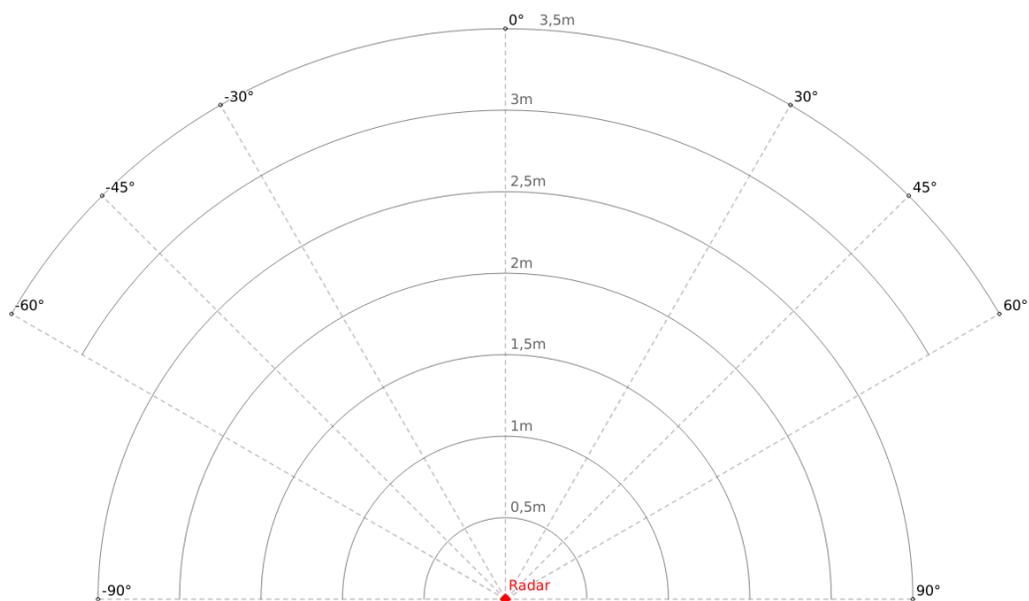


Figure 4.5 : Schéma illustrant les marqueurs des tests circulaires effectués

Les Figures 4.6, 4.7 et 4.8 montrent un aperçu des résultats obtenus. Elles représentent la distance de la personne la plus proche en fonction de son écart avec l'axe central du champ de détection. Les autres résultats des tests circulaires sont disponibles en annexe A. Bien que les mesures effectuées aux angles -90° et 90° semblent moins stables et fiables que le reste, on observe une assez forte homogénéité des mesures avec des erreurs ponctuelles ne semblant pas être dépendantes de l'angle du sujet avec la droite centrale du champ de détection. Ces résultats ne permettent pas de conclure que, pour des distances allant de 0,5m à 2,5m et pour un sujet présent dans le demi-cercle faisant face au radar, l'angle en question impacte les performances de l'algorithme de distance appliqué aux données du radar. De la même manière, une conclusion similaire peut être faite pour les demi-cercles réduits de 3m et 3,5m de rayon. Les erreurs semblent marginales et non corrélées avec la nature du test.

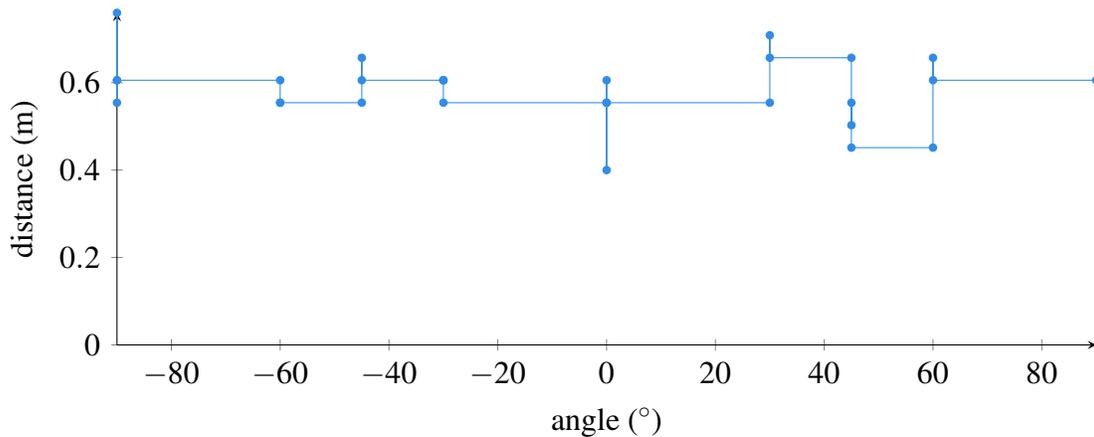


Figure 4.6 : Test de précision circulaire sur un demi-cercle de 0,5m de rayon

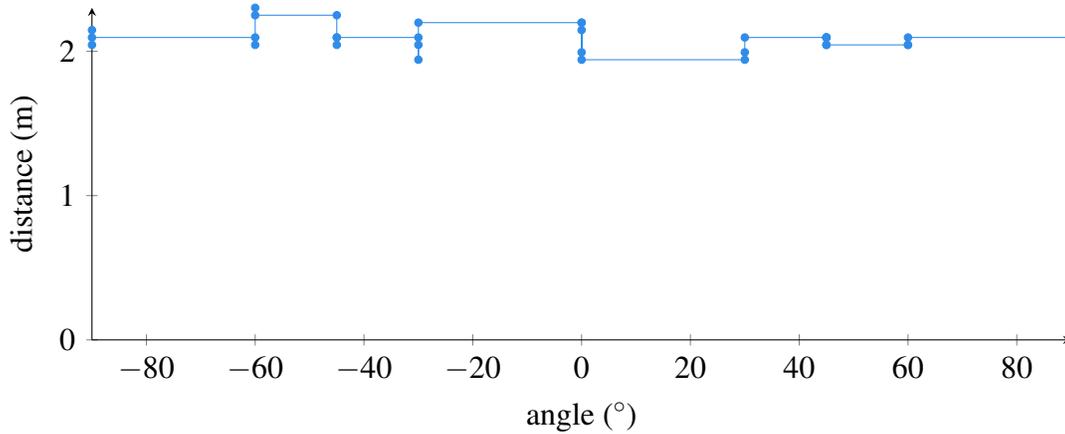


Figure 4.7 : Test de précision circulaire sur un demi-cercle de 2m de rayon

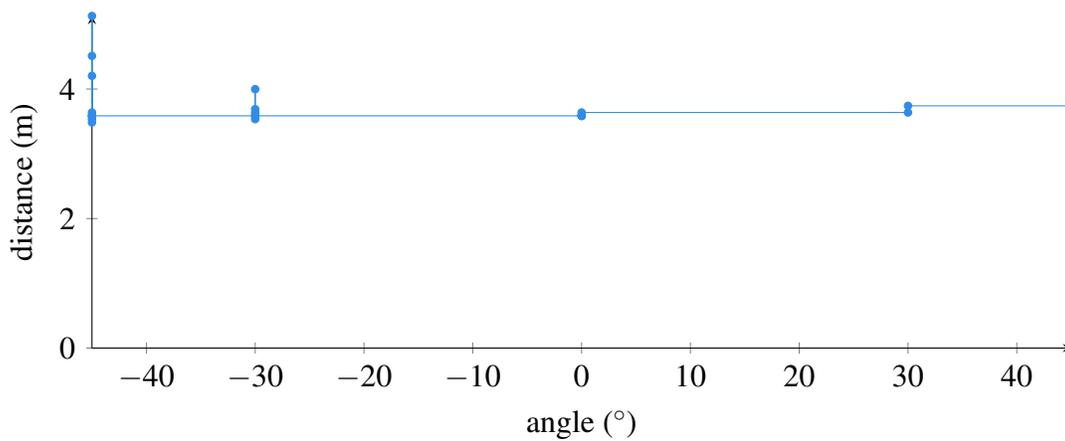


Figure 4.8 : Test de précision circulaire sur un demi-cercle de 3,5m de rayon

4.3.2 TESTS DE PRÉCISION LINÉAIRE

Des tests similaires ont été effectués pour déterminer la portée effective et la précision du radar sur la droite centrale du champ de détection et sur la droite perpendiculaire à cette dernière. La Figure 4.9 illustre les deux droites en question. La première série de tests a pour objectif de vérifier la portée maximale du radar, dont les limites théoriques semblaient indiquer un maximum de 15m. Les paramètres de configuration par défaut du radar indiquant 9,8m, ils ont donc été étendus jusqu'à une portée de 15m. Cependant, les

tests ont immédiatement montré une incapacité du radar ou de l’algorithme de distance à fonctionner au-delà de 9,8m. Ainsi, aucun graphique n’a pu être tracé à partir de ces données. Des tests visant à déterminer la précision du radar selon la distance du sujet avec celui-ci ont été réalisés avec des mesures tous les 0,25m d’intervalle sur une portée de 0,5m à 5m pour la première série puis de 5m à 10m pour la seconde.

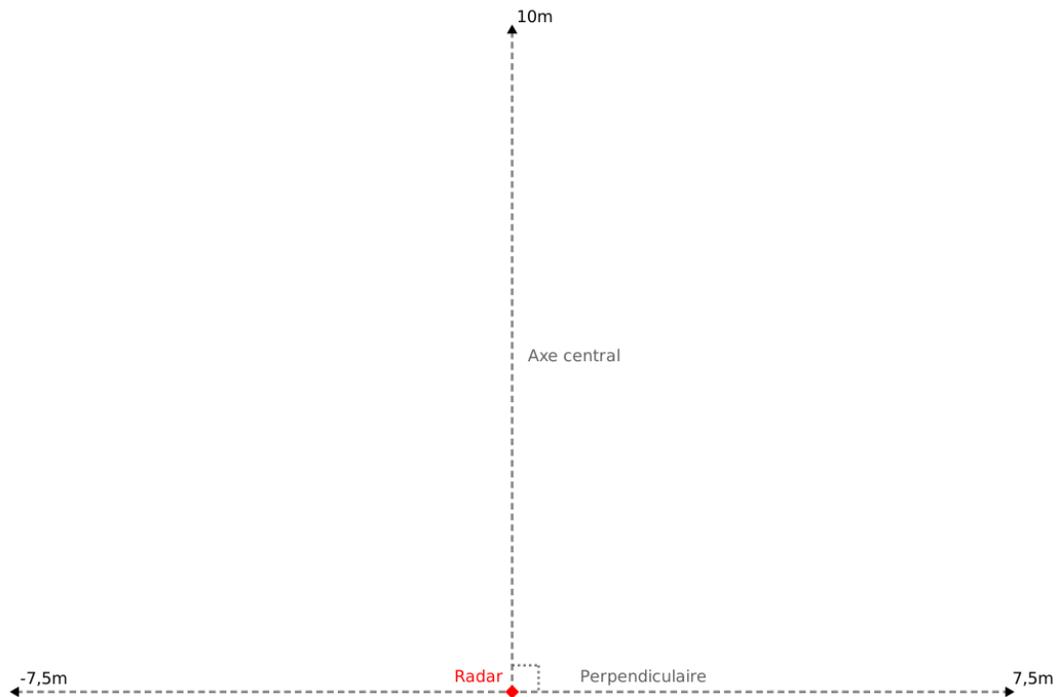


Figure 4.9 : Schéma illustrant les marqueurs des tests linéaires effectués

Les Figures 4.10 et 4.11 montrent des résultats stables et satisfaisants, indiquant une précision imparfaite, mais néanmoins toujours dans un intervalle de 0,5m centré sur la distance réelle du sujet. Les autres résultats de ces tests linéaires sont disponibles en annexe B. Un plateau semble être observé aux alentours de 9m à 9,5m, confirmant la présence d’une limite de détection à 9,8m de distance. Néanmoins, quelques difficultés

de détection sont observables dès 9m de distance.

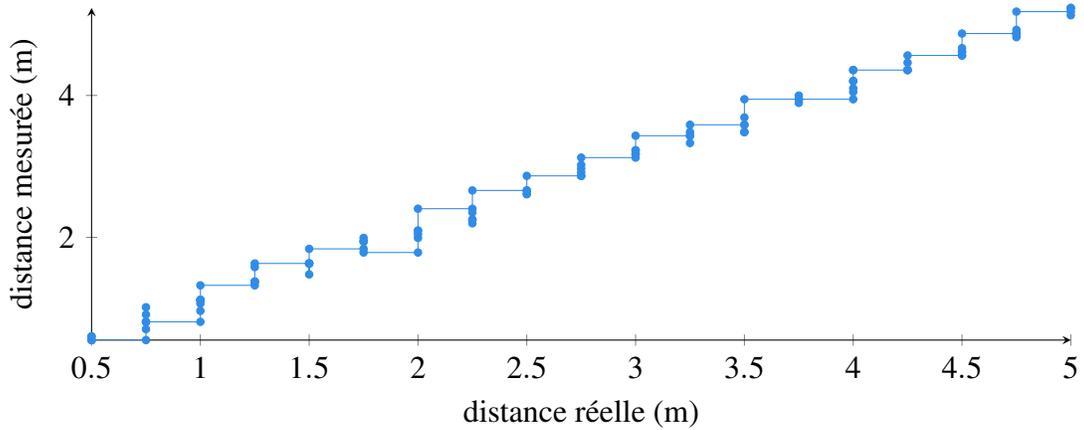


Figure 4.10 : Test de précision linéaire sur l'axe central de 0,5m à 5m

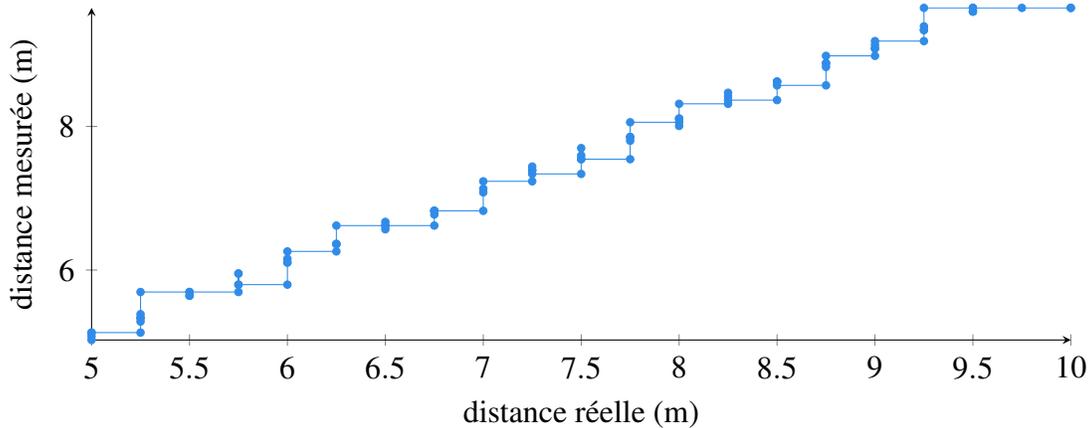


Figure 4.11 : Test de précision linéaire sur l'axe central de 5m à 10m, avec les paramètres de portée par défaut étendus

Enfin, deux séries de tests sur la droite passant par le radar et perpendiculaire à la droite centrale du champ de détection ont été effectués. La droite en question est illustrée en Figure 4.9. Le sujet s'est déplacé sur un segment de 15m de long centré sur le radar, avec des arrêts marqués tous les 0,5m. Les résultats des deux séries sont

illustrés sur les Figures 4.12 et 4.13. Elles montrent toutes deux une incapacité des radars à détecter le sujet au-delà de 6m de distance, et une très nette diminution de la fiabilité des résultats au delà de 4m. Ainsi, des difficultés pour détecter le sujet sur la droite perpendiculaire à la droite centrale de détection et passant par le radar sont mises en lumière. Cela reste cohérent avec les performances et les limites annoncées sur la fiche technique du radar¹³.

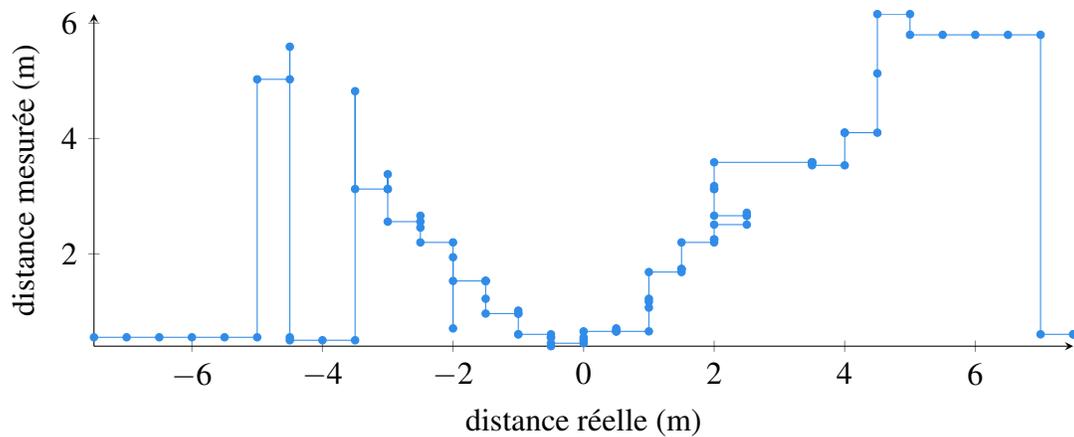


Figure 4.12 : Test de précision linéaire sur la perpendiculaire de 0m à 5m, série 1

13. Fiche technique : https://www.codico.com/shop/media/datasheets/Novelda_X4M200_respiration_sensor_rev_c_preliminary.pdf

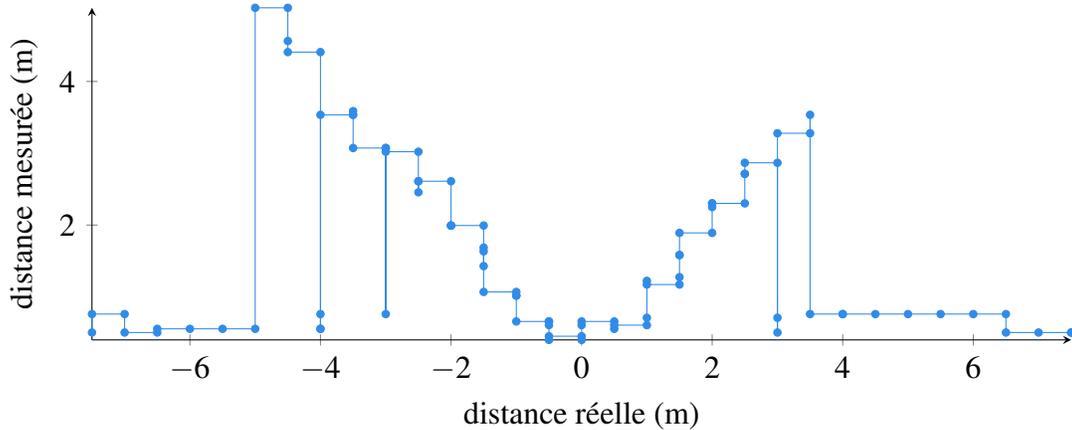


Figure 4.13 : Test de précision linéaire sur la perpendiculaire de 0m à 5m, série 2

4.3.3 CONCLUSION SUR LES TESTS

L'analyse globale de ces tests ne permet pas de conclure avec certitude sur les limites précises du champ de détection des radars, sur la précision de ces derniers, ou sur les performances de l'algorithme de distance utilisé. En effet, un protocole beaucoup plus exigeant et une plus grande exhaustivité de tests auraient été nécessaires pour tirer ce genre de conclusion. En revanche, ils permettent de mettre en lumière quelques limites des radars en combinaison avec l'algorithme de distance, notamment la portée maximale de moins de 10m, la dégradation de la fiabilité des données au-delà de 9m, ainsi que la difficulté des radars à détecter un sujet proche de la droite passant par le radar et perpendiculaire à la droite centrale du champ de détection. Ces limites ont permis d'anticiper un placement optimal des radars dans l'habitat intelligent du LIARA en s'assurant que ces derniers soient positionnés le long de murs où aucun équipement n'est présent, et de manière à compenser au mieux les limites de portée.

CHAPITRE V

JEUX DE DONNÉES

5.1 OBJECTIFS

Dans le cadre de ce projet de recherche, deux jeux de données distincts ont été récoltés puis utilisés par diverses méthodes d'apprentissage machine. Le premier d'entre eux vise à distinguer plusieurs activités de la vie quotidienne selon une classification discrète non-binaire, et le second a pour objectif de distinguer une chute de toute autre activité selon une classification binaire.

Le jeu de données des activités regroupe plusieurs activités de haut-niveau de la vie quotidienne, c'est-à-dire assez générales pour englober des sous-activités élémentaires. Par exemple, nous cherchons à reconnaître "faire des pâtes" plutôt que "remplir un récipient d'eau" puis "faire bouillir de l'eau", etc. Ces activités se veulent diverses par leur nature, par leur durée et par le lieu de leur réalisation dans l'habitat intelligent.

Le jeu de données des chutes est constitué de chutes au sol effectuées à trois endroits différents dans l'habitat intelligent, complétées par des instances témoins au cours desquelles aucune chute n'a lieu. Quatre types de chutes inspirés des travaux de Robinovitch *et al.* (2013) ont été réalisés à répartition égale afin de représenter la diversité caractéristique de ce genre d'accidents. L'objectif est de pouvoir détecter efficacement la présence de chutes parmi des activités du quotidien, afin de pouvoir agir en conséquence le cas échéant.

5.2 PROTOCOLE DE RÉCOLTE

5.2.1 L'ENVIRONNEMENT DE RÉCOLTE

Les deux jeux de données ont été récoltés dans l'habitat intelligent du LIARA, habitat conçu pour reproduire de manière fidèle et réaliste un logement pour une personne. On y retrouve deux pièces distinctes : la pièce de vie et la salle de bain. La pièce de vie est divisée en trois parties : un coin chambre, un coin cuisine et un coin salon. L'appartement est entièrement meublé et équipé pour simuler le mieux possible des conditions réelles. L'habitat est parsemé de nombreux capteurs : tactiles, infrarouges, détecteurs de luminosité, capteurs de pression, balises RFID, capteurs à ultrasons, détecteurs de fumée. Enfin, il est équipé des trois radars ultra-wideband utilisés dans le cadre de ce projet de recherche.

Les trois radars sont répartis dans l'habitat de manière à couvrir collectivement la plus grande zone possible tout en minimisant le taux d'erreur. La position la plus optimale est représentée dans le plan de l'habitat intelligent du LIARA illustré en Figure 5.1¹⁴. Ils sont tous les trois positionnés à 36cm du sol, c'est-à-dire à la hauteur des prises électriques de l'habitat. Malgré leurs capacités de détection à travers les murs et les objets, on observe quelques difficultés à détecter parfaitement et continuellement les personnes présentes dans la salle de bain. Cependant, c'est cette configuration qui a montré les résultats les plus probants lors des tests préliminaires visant à évaluer le

14. Extrait de *Recognizing Activities of Daily Living from UWB Radars and Deep Learning* de Maitre J., Bouchard K., Bertuglia C., Gaboury S., en cours d'évaluation par des pairs.

champ de détection des capteurs ¹⁵.

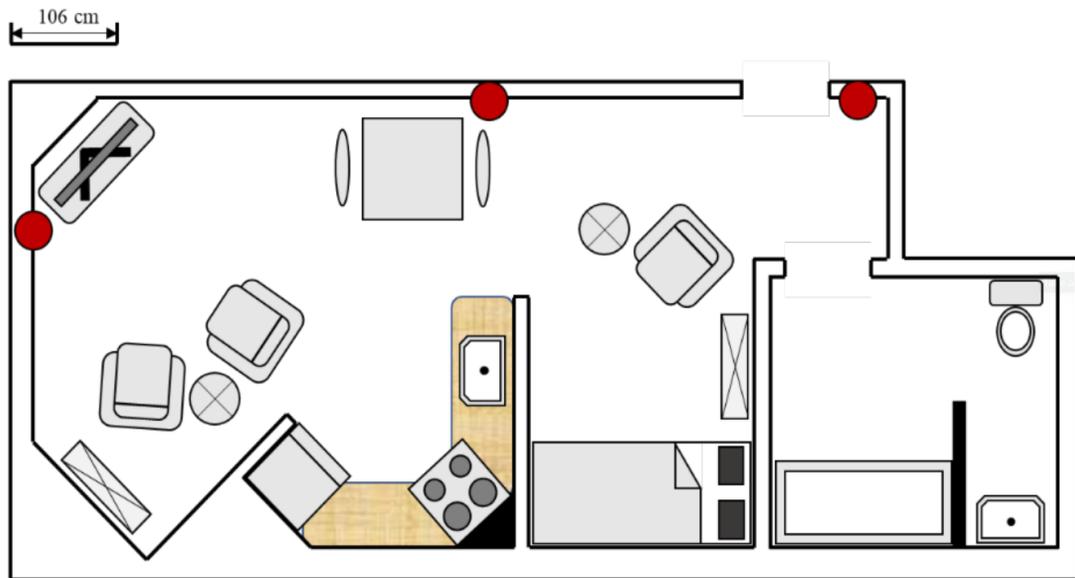


Figure 5.1 : Le plan vu du dessus de l’habitat intelligent du LIARA avec les positions des radars ultra-wideband en rouge

Pour le jeu de données des chutes, un matelas est disposé au sol. Ce matelas est présent sur le sommier du lit lors de la récolte des autres activités. C’est l’unique différence d’agencement du mobilier entre les deux jeux de données, l’objectif étant de conserver le mieux possible l’état de l’appartement dans lequel les tests ayant permis de positionner les radars UWB ont été effectués.

15. Ces tests préliminaires ont été réalisés dans l’habitat intelligent et sont basées sur les résultats des tests de précision détaillés dans le chapitre précédent

5.2.2 PROCÉDURE DE RÉCOLTE D'UNE INSTANCE

La procédure de récolte est la suivante. Le chercheur référent s'assure de l'émission continue des données par le radar. Il sort de l'enceinte de l'habitat et démarre l'application de récolte à distance commençant par un compte à rebours. À l'issue de ce dernier, un signal verbal est exprimé envers le participant, qui est amené à réaliser l'activité étape par étape selon le protocole qui lui a été communiqué auparavant. L'acquisition s'arrête automatiquement une fois que le temps prévu pour l'activité est écoulé, enregistrant ainsi les données sur le terminal distant du chercheur. Enfin, la vérification de la taille des données récoltées permet de s'assurer de l'absence de tout dysfonctionnement technique. Dans le cas de la récolte du jeu de données des chutes, une vérification du maintien de la position du matelas et des autres mesures de sécurité est effectuée avant chaque capture.

5.3 LE JEU DE DONNÉES DES ACTIVITÉS

5.3.1 DÉFINITION DU JEU DE DONNÉES

Le jeu de données des activités est composé de 15 activités, répétées chacune dix fois. Chaque répétition est effectuée par un participant différent. La liste des activités est la suivante :

— Marcher	30 secondes, 1 étape
— Dormir	120 secondes, 3 étapes
— Mettre des chaussures	45 secondes, 5 étapes
— Enfiler une veste	120 secondes, 2 étapes

— Lire un livre	120 secondes, 5 étapes
— Utiliser un ordinateur portable	120 secondes, 3 étapes
— Faire du thé	180 secondes, 6 étapes
— Faire des pâtes	300 secondes, 11 étapes
— Faire la vaisselle	180 secondes, 10 étapes
— Faire le ménage	120 secondes, 5 étapes
— Se laver les dents	180 secondes, 7 étapes
— Se laver les mains	30 secondes, 7 étapes
— Manger	30 secondes, 4 étapes
— Boire	30 secondes, 3 étapes
— Prendre des médicaments	30 secondes, 5 étapes

Chaque activité est composée de sous-activités élémentaires dont la liste est communiquée aux participants à l’avance. Seul le respect de cette liste est exigé, les autres paramètres étant laissés à la discrétion du participant. Par exemple, l’action “se laver les mains”, qu’elle soit effectuée dans la cuisine ou la salle de bain, est valide à partir du moment où les étapes élémentaires sont effectuées. Voici un exemple simple du déroulement de l’activité “se laver les mains” :

1. Ouvrir le robinet
2. S’humidifier les mains
3. Fermer le robinet
4. Se savonner les mains

5. Ouvrir le robinet
6. Se rincer les mains
7. Fermer le robinet

Le temps attribué à chaque activité n'est pas nécessairement identique. Par exemple, "se laver les mains" nécessite une acquisition de 30 secondes, là où trois minutes sont requises pour "se laver les dents". Ainsi, bien que la répartition des classes soit uniforme, la quantité brute de données acquises pour chaque classe n'est pas égale. Afin d'équilibrer les données lorsque l'apprentissage est impacté par ce phénomène, un complément composé de 65 instances d'activités a été récolté. Parmi celles-ci, on retrouve 11 fois "marcher", "enfiler une veste", "se laver les mains", "boire" et "prendre des médicaments", ainsi que dix fois "mettre des chaussures".

La récolte du jeu de données des activités respecte avec rigueur les exigences d'éthiques propres aux expérimentations impliquant des êtres humains et à reçu l'approbation du Comité d'Éthique de la Recherche (CER) de l'UQAC¹⁶. Aucune activité ne présentant de risques, la présence d'un équipement de protection ne fut pas nécessaire durant l'acquisition (Maître *et al.*, 2019b).

5.3.2 RÉSULTAT

Le jeu de données final regroupe 150 fichiers représentant un poids de 17,5 Go, complété par une extension de 65 fichiers totalisant 2,2 Go. Chacun de ces fichiers contient les données brutes envoyées par les trois radars et reçues par le logiciel d'acqui-

16. Numéro du certificat d'éthique : 2019-220, certificat disponible en annexe C

sition, étayées de l'identifiant des radars et d'une balise temporelle pour chaque image. Un calcul des distances de la personne détectée avec les radars et de ses coordonnées relatives dans l'habitat intelligent est effectué toutes les secondes. Ces données sont ajoutées en complément des données brutes lors de l'acquisition. Un extrait d'un fichier issu du jeu de données est disponible en annexe D.

5.4 LE JEU DE DONNÉES DES CHUTES

5.4.1 DÉFINITION DU JEU DE DONNÉES

Le jeu de données des chutes est composé de quatre types de chutes différentes : chute en avant, chute en arrière, chute sur le côté et chute en s'asseyant. Chaque type de chute est répété cinq fois par participant et par position dans l'habitat. La diversité des chutes est inspirée de la publication scientifique (Robinovitch *et al.*, 2013). Cette étude analyse et catégorise différentes chutes de personnages âgés filmées en centre de soin. Des extraits vidéos ont été montrés aux participants afin de les amener à imiter au mieux ces chutes spontanées et authentiques. Les chutes ont été effectuées à trois positions différentes dans l'habitat. Ces positions sont illustrées en Figure 5.2¹⁷. Les meubles entrant en conflit avec les positions ont été déplacés temporairement lors de l'acquisition des données.

Chaque chute représente une acquisition de 20 secondes durant laquelle le participant chute, puis reste au sol jusqu'à la fin du temps imparti. Il ne lui est pas interdit de

17. Extrait et modifié de *Recognizing Activities of Daily Living from UWB Radars and Deep Learning* de Maitre J., Bouchard K., Bertuglia C., Gaboury S., en cours d'évaluation par des pairs.

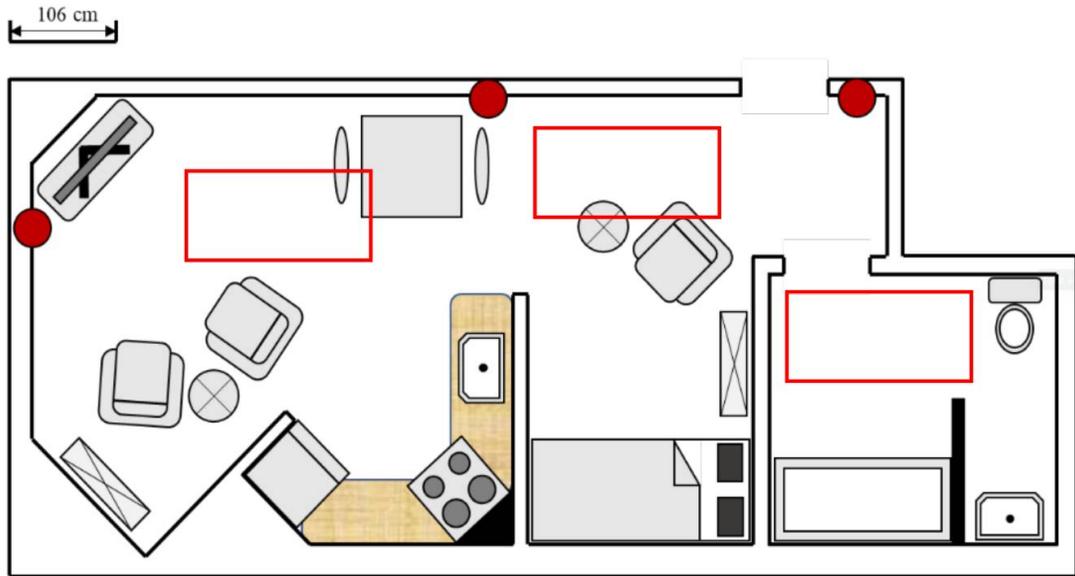


Figure 5.2 : Le plan vu du dessus de l’habitat intelligent du LIARA avec les positions des chutes effectuées désignées par les rectangles rouges.

bouger après la chute, mais il lui est demandé de ne pas se relever. Les chutes ont été effectuées à trois emplacements différents dans l’habitat intelligent, et aucun participant n’a effectué chaque type de chute plus de cinq fois par emplacement. En complément de ces chutes s’ajoute un total de 30 acquisitions témoins de deux minutes chacune, durant lesquels les participants ont été amenés à réaliser une activité de leur choix dans l’appartement, à condition qu’aucune chute ne soit effectuée.

La récolte du jeu de données des chutes respecte avec rigueur les exigences d’éthiques propres aux expérimentations impliquant des êtres humains et à reçu l’approbation du CER de l’UQAC¹⁸. Un matelas a été disposé au sol en face de chaque position prévue pour les chutes, et un équipement de protection complet incluant casque,

18. Numéro du certificat d’éthique : 2019-235, certificat disponible en annexe E

genouillères et coudières a été mis à disposition des participants. Des mesures pour assurer une prise en charge optimale des participants en cas d'accident ont également été déployées. L'intégralité des exigences de sécurité est spécifiée dans la demande d'approbation d'éthique relative à cette expérimentation (Maître *et al.*, 2019a).

5.4.2 RÉSULTAT

Le jeu de données final regroupe 630 fichiers dont 600 chutes, pour un poids total de 17,5 Go. Chaque fichier contient des données similaires à celles récoltées pour le jeu de données des activités et est structuré comme spécifié dans la section précédente.

5.5 VISUALISATION DES DONNÉES, TEMPS RAPIDE ET TEMPS LENT

5.5.1 TEMPS RAPIDE

Les données renvoyées par le radar se présentent sous la forme de nombres complexes caractérisant les réflexions captées par le radar et renseignant sur l'amplitude et la phase de ces dernières. Chaque radar a la possibilité de détecter sur 184 échantillons répartis sur une plage de 0,4m à 9,8m de distance du radar, fournissant ainsi une résolution de 5,14cm. Ces échantillons sont appelés “bins”. La concaténation des signaux reçus sur l'entièreté des bins pour une unique pulsation constitue une “image” radar, c'est-à-dire la représentation instantanée de son environnement de manière similaire à une photographie. On appelle également cette représentation “temps rapide” (fast-time en anglais). Une représentation de cette image et de la dimension du temps rapide est disponible en Figure 5.3.

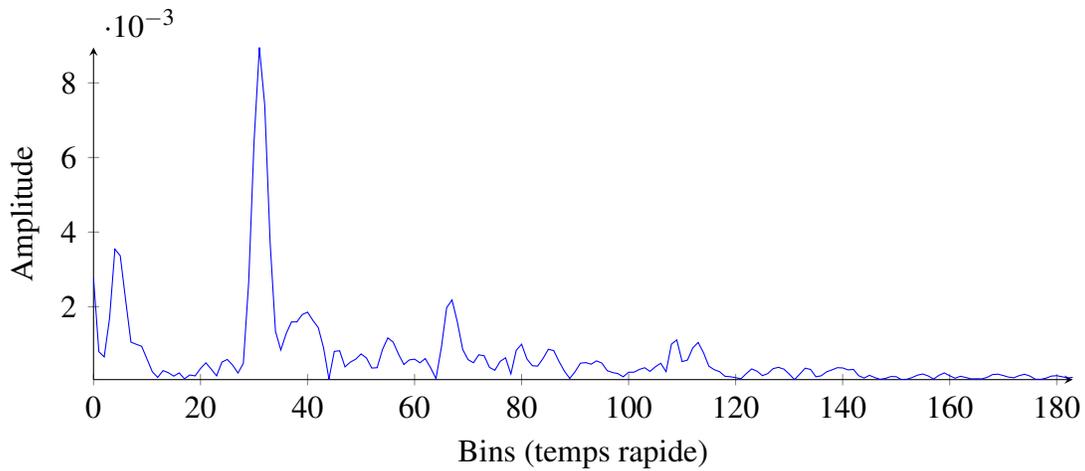


Figure 5.3 : Représentation d’une image radar extraite de l’instance d’activité “Boire_0” et visualisation de la notion de temps rapide.

5.5.2 TEMPS LENT

Les radars émettant des pulsations toutes les 20 millisecondes (50Hz), on peut regrouper successivement les images temps-rapide d’un radar pour avoir un aperçu semblable à une vidéo, où le mouvement est perceptible. Le regroupement de ces images sur plusieurs pulsations est appelé “temps long” (ou *emphslow-time* en anglais). On appelle la période durant laquelle on considère les données des différentes pulsations sur un temps long “fenêtre” temporelle (ou *window* en anglais). Là où on peut représenter une image radar correspondant à une pulsation en deux dimensions, on représente traditionnellement la succession des pulsations soit sous forme de vidéo, soit sous forme de cubes de données lorsqu’il est nécessaire de figer l’information. La Figure 5.4 représente une fenêtre temporelle d’un dixième de secondes et les dimensions temps rapide et temps lent pour un radar.

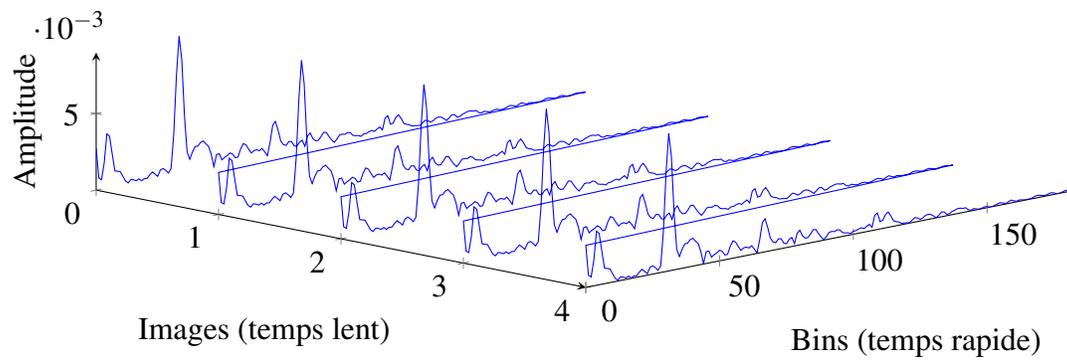


Figure 5.4 : Représentation d'une fenêtre temporelle de 100 millisecondes (5 images) et des notions de temps rapide et de temps lent.

Ces trois notions (temps rapide, temps lent et fenêtre temporelle) sont centrales pour appréhender l'extraction des caractéristiques nécessaires à l'entraînement des modèles d'apprentissage machine dont le processus est détaillé dans le Chapitre 6.

CHAPITRE VI

CONTRIBUTION

Préambule Les méthodes d'apprentissage machine évoquées dans ce chapitre ont été implémentées en Python grâce à la très populaire bibliothèque scikit-learn (Pedregosa *et al.*, 2011)¹⁹. L'intégralité du code propre aux algorithmes d'apprentissage est donc disponible sur la plateforme de développement GitHub²⁰. Scikit-learn est une bibliothèque open-source ayant pour objectif de rendre les techniques d'intelligence artificielle accessibles pour tous. Elle est notoirement très efficace pour tester rapidement et efficacement un grand panel de méthodes et leurs variantes les plus populaires. Cette bibliothèque est régulièrement maintenue à jour. Les travaux effectués dans le cadre de ce mémoire utilisent la version 0.23 de celle-ci.

INDICES D'ÉVALUATION DE PERFORMANCE

Tout au long de ce chapitre, les résultats des tests indiqueront essentiellement deux indices d'évaluation de performance : l'*accuracy* et le *Cohen kappa*. Il est important de préciser que de nombreux autres indices d'évaluation de performances ont été calculés pour ces tests (*precision*, *recall*, *f1 score*, *hamming loss*, *matthew correlation coefficient*²¹). Cependant, pour des soucis de lisibilité, seuls les indices *accuracy* et *Cohen kappa* seront affichés. Lorsqu'ils ne sont pas inclus et sauf précision du contraire,

19. <https://scikit-learn.org/stable/index.html>

20. <https://github.com/scikit-learn/scikit-learn>

21. https://scikit-learn.org/stable/modules/model_evaluation.html

les autres indices n'apportent pas d'information utile supplémentaire. De plus, pour les indices possédant les variantes "micro", "macro" et "pondéré", c'est la version pondérée qui sera présentée. Les indices d'évaluation seront désignés par leurs termes anglais pour éviter toute confusion, notamment entre les indices *accuracy* et *precision* qui peuvent tous les deux se traduire par "précision" en français. Voici la liste des indices d'évaluation utilisés dans ce rapport et leurs significations.

- *accuracy* : Cet indice représente le ratio de prédictions correctes sur le nombre d'instances présentées. Une *accuracy* de 1 signifie que chaque instance a été classée correctement. Dans le cas d'un problème multi-classe, plus le nombre de classe est grand, plus obtenir un score proche de 1 est difficile. Cet indice varie entre 0 et 1.
- *Cohen kappa* : Cet indice vise à déterminer l'écart entre les classes prédites par le modèle et les classes réelles sur le jeu de données de test. Un score proche de 1 correspond à une équivalence parfaite. Un score de 0 ou moins représente un écart fort. Cet indice varie entre -1 et 1.
- *precision* (micro, macro, pondéré) : Cet indice représente, dans le cas d'une classification binaire, la capacité à ne pas détecter positivement un cas négatif. Dans le cas d'une classification multi-classe, on utilise trois variantes. La première, "micro", calcule la précision de manière à ce que chaque instance classée pèse de manière équitable sur l'indice, sans prendre en compte la prépondérance d'une classe dans le jeu de données. Lorsque chaque classe possède la même importance et que le jeu de données n'est pas équilibré, cela peut être problématique. Dans

d'autres cas, ce biais est recherché. La deuxième variante, "macro", calcule la moyenne de la précision relative à chaque classe individuellement. Elle ne pondère pas les classes. Ainsi, une classe sous-représentée dans le jeu de données aura autant de poids qu'une classe sur-représentée. Enfin, "pondéré" effectue le même calcul que macro, mais pondère les moyennes des classes en fonction de la prépondérance de la classe dans le jeu de données. C'est un entre-deux de micro et macro.

- *recall* (micro, macro, pondéré) : Le *recall* (rappel) représente, lors d'une classification binaire, la capacité à classer positivement chaque instance qui doit l'être. Cet indice possède les mêmes variantes (micro, macro et weighted) pour s'adapter aux problèmes multi-classes.
- *f1 score* (micro, macro, pondéré) : Cet indice représente l'entre-deux de la *precision* et du *recall*. Comme ces deux indices, il varie entre 0 et 1, avec 1 le meilleur score possible.
- *hamming loss* : La perte de Hamming représente la distance de Hamming entre deux jeux de données, ici le jeu de test classé tel qu'il devrait être et le jeu de test classé par le modèle. Plus le score est faible, plus le modèle est performant.
- *matthews corrcoeff* : Le coefficient de corrélation de Matthews est initialement prévu pour la classification binaire et pour prendre en compte les différents ratios de faux-positifs, vrai-positifs, faux-négatifs et vrai-négatifs. Une variante multi-classe existe, c'est elle qui est utilisée ici. La meilleure valeur de cet indice est 1.

6.1 JEUX DE DONNÉES À SIMPLE FENÊTRE

6.1.1 INTRODUCTION AUX JEUX DE DONNÉES

Souvent plus complexe que la classification binaire, la classification multi-classe présente un défi intéressant et particulièrement pertinent pour la classification d'activités de la vie quotidienne. En effet, dans le cadre d'une application concrète, on doit faire face à une multitude d'activités parfois très diverses. Ainsi, ce chapitre se concentre exclusivement sur l'application de plusieurs techniques d'apprentissage machine sur le jeu de données des activités. C'est sa version étendue, avec 215 instances d'activités au total, qui est utilisée. Cependant, le jeu de données, sous la forme qu'il possède juste après sa récolte, n'est pas directement exploitable pour l'apprentissage machine. Sa taille et sa complexité le rendent peu propice à toute application directe. Ainsi, plusieurs étapes de pré-traitement doivent être effectuées.

La première consiste à transformer les instances d'activités en plusieurs séries temporelles selon la dimension temps lent évoquée dans le précédent chapitre. Chaque fichier est divisé en fenêtres temporelles de quelques secondes se chevauchant les unes avec les autres selon un certain ratio, pour former plusieurs séries temporelles contenant les informations essentielles sur les temps rapide et lent des données. Par défaut, on considère que la fenêtre temporelle temps lent est de 15 secondes, et que le ratio de chevauchement est de 90%. Le chevauchement signifie que 90% des données d'une fenêtre temporelle sont également utilisées par la fenêtre temporelle suivante. Les séries temporelles extraites du jeu de données brut sont alors traitées selon des techniques de traitement du signal et ses caractéristiques standards sont extraites : coefficients

minimums, maximums, moyenne, variance, degré d'asymétrie, kurtosis, écart moyen, longueur du signal, corrélation croisée, etc. Ces informations sont alors condensées sous un format standard utilisé par la bibliothèque scikit-learn dans un unique fichier faisant en moyenne entre 100Mo et 200Mo. L'implémentation de ces différentes étapes de pré-traitement est l'œuvre du travail de M. Julien Maître, co-directeur du projet de recherche dans lequel s'inscrit ce mémoire.

Afin de tester l'impact de la taille de la fenêtre et du ratio de chevauchement sur les différentes méthodes d'apprentissage machine, plusieurs variations du jeu de données pré-traité ont été réalisées. Ainsi, des tests avec des fenêtres temporelles de 5, 15 et 20 secondes combinées avec des taux de chevauchement de 0,8 et 0,9 ont été effectués. On obtient ainsi un total de 6 jeux de données différents. Il est important de noter que les jeux ayant une petite fenêtre temporelle et un grand taux de chevauchement sont plus volumineux. Par exemple la combinaison "5_0.9" fait 190Mo là où la combinaison "20_0.8" fait 73Mo. Enfin, chaque jeu de données est divisé en deux sous-ensembles : le jeu de données d'entraînement qui est utilisé pour générer les modèles, et le jeu de données de test qui est utilisé pour les valider et évaluer leurs performances. Le ratio divisant les données d'entraînement et de test utilisé est de 70/30.

6.1.2 ARBRES DE DÉCISION

ALGORITHME ET PARAMÉTRAGE

Dans la bibliothèque scikit-learn, la méthode des arbres de décision est implémentée avec l'algorithme CART. C'est un algorithme similaire à l'algorithme C4.5,

lui-même successeur de l’algorithme ID3. La version de l’algorithme CART utilisée par cette librairie est une version revendiquée comme optimisée²². Son code source est disponible sur la plateforme de développement GitHub²³. Plusieurs paramètres, parfois génériques aux arbres de décision, parfois propres à l’algorithme CART, peuvent être utilisés comme levier pour optimiser les performances de l’algorithme. Voici une courte présentation de certains de ces paramètres.

- Critère d’impureté (criterion) : C’est le choix de la fonction d’impureté utilisée lors du calcul du gain d’information. Pour rappel, celui-ci est utilisé afin de sélectionner un attribut discriminant pour chaque nœud de l’arbre de décision. Il est possible d’utiliser la fonction d’entropie (principalement utilisée par les algorithmes ID3 et C4.5), ou le critère de Gini (principalement utilisé par l’algorithme CART).
- Profondeur maximum de l’arbre (max_depth) : C’est la profondeur maximum en nombre de nœuds pour atteindre les feuilles depuis la racine de l’arbre. Autrement dit, c’est la longueur du chemin le plus long à parcourir dans l’arbre de décision. Une profondeur sans limite peut entraîner un arbre trop long et complexe, sujet au sur-apprentissage. Une profondeur trop petite transforme l’arbre en “tronc”, c’est-à-dire en arbre très basique comme ceux utilisés pour l’algorithme AdaBoost. Ce paramètre peut soit être un entier correspondant à la profondeur maximale, soit la valeur “aucune” qui ne limite donc pas la profondeur de l’arbre du tout.
- Taille minimum de l’échantillon pour une séparation (min_samples_split) : C’est le nombre minimum d’instances dans un échantillon requis pour séparer un nœud.

22. <https://scikit-learn.org/stable/modules/tree.html>

23. <https://github.com/scikit-learn/scikit-learn>

Tout échantillon de taille inférieure à ce nombre entraîne forcément une feuille.

Ce paramètre peut prendre comme valeur n'importe quel entier supérieur ou égal à 2.

- Taille minimum de l'échantillon pour une feuille (`min_samples_leaf`) : C'est le nombre minimum d'instances dans un échantillon requis pour transformer le nœud suivant en feuille. Pour une séparation, aucun attribut discriminant qui génèrerait des échantillons de taille inférieure à ce paramètre ne sera considéré. Ce paramètre peut prendre comme valeur n'importe quel entier supérieur ou égal à 1.
- Attributs maximums à considérer (`max_features`) : C'est le nombre d'attributs à considérer lors du calcul de la meilleure séparation possible pour un nœud. Il est possible de sélectionner "aucun", auquel cas ce paramètre prend comme valeur le nombre total d'attributs. Sinon, on peut choisir "sqrt" ou "log2", auquel cas c'est le résultat du nombre total d'attributs à travers les fonctions racine carrée ou log2 qui est considéré.
- Réduction minimum d'impureté (`min_impurity_decrease`) : C'est le seuil de réduction de l'impureté requis pour créer un nœud lors de la considération de la sélection d'un attribut pour une séparation. Si la réduction d'impureté est inférieure à ce seuil pour tout attribut considéré, le nœud devient une feuille.

RÉSULTATS

Afin de limiter la quantité de tests nécessaires pour définir le paramétrage optimal de l'algorithme, une première série d'apprentissage sur les différents jeux de données

avec le paramétrage par défaut a été effectuée. Cette série de tests permet de mettre en valeur le jeu de données sur lequel cette technique d'apprentissage machine est, a priori, la plus efficace. Ainsi, les tests de paramétrage ne seront faits que sur une ou deux variations du jeu données plutôt que sur les six. La configuration de l'algorithme par défaut est la suivante : `criterion = gini`, `max_depth = None`, `min_samples_split = 2`, `min_samples_leaf = 1`, `max_features = None`, `min_impurity_decrease = 0`. Le Tableau 6.1 condense les résultats principaux de ces tests. Ce tableau montre une *accuracy* et un *Cohen kappa* nettement supérieurs pour les fenêtres avec un chevauchement de 0,9. Les fenêtres de 15 secondes et de 20 secondes, toutes deux relativement équivalentes entre elles, semblent prévaloir sur les fenêtres de 5 secondes. Ainsi, les tests de paramétrages seront effectués sur le jeu de données "15_0.9" et "20_0.9", présentant les meilleurs scores dans ce tableau. Lorsque les résultats montrent une tendance claire et similaire sur les deux jeux, seuls les résultats du jeu de données "20_0.9" seront présentés.

Fenêtre	Chevauchement	Accuracy	Cohen Kappa
5	0.8	0.49	0.43
15	0.8	0.55	0.5
20	0.8	0.56	0.5
5	0.9	0.58	0.54
15	0.9	0.66	0.62
20	0.9	0.67	0.63

Tableau 6.1 : Tableau représentant l'accuracy et le Cohen kappa selon la taille des fenêtres temporelles (time window) et le chevauchement (overlap) des jeux de données. Algorithme CART.

Les premiers paramètres testés sont le critère de calcul d'impureté et la profondeur maximale de l'arbre. Pour l'indice de Gini comme pour le critère d'entropie, 14 valeurs de profondeur maximum ont été testées allant de 3 jusqu'à "sans limite". Les résultats

sont représentés par les graphiques des Figures 6.1 et 6.2. Les performances entre les deux critères étant relativement proches sur le jeu de données “20_0.9”, les résultats sur le jeu de données “15_0.9” permettent d’avoir un second aperçu sur l’impact du critère. Ici, on observe que le critère d’entropie est légèrement supérieur à l’indice de Gini dans l’essentiel des cas, particulièrement sur ceux entraînant la plus grande *accuracy*. L’augmentation de la profondeur maximale entraîne une forte hausse des performances, jusqu’à atteindre un plateau autour de la valeur 11. Cette profondeur est probablement égale ou supérieur à celle de la plupart des branches de l’arbre, même lorsque la limite est plus grande. Cependant, on observe un léger pic à la profondeur 11 précisément, qui peut s’expliquer par l’effet d’élagage. En effet, cette limite permet peut-être de réduire légèrement la spécificité du modèle et d’éviter l’écueil du sur-apprentissage. On considérera par la suite que l’entropie comme critère de gain d’information et 11 comme profondeur maximale seront les valeurs optimales.

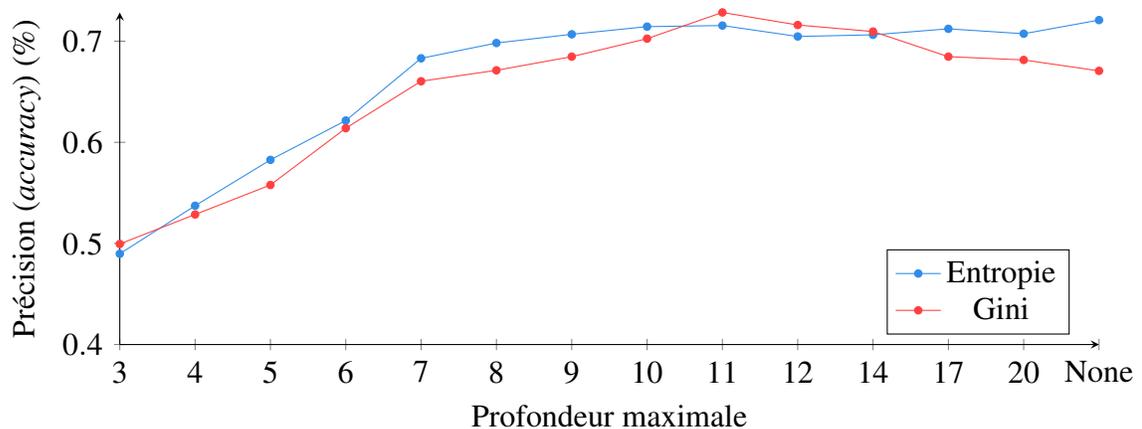


Figure 6.1 : Accuracy en fonction de la profondeur maximum de l’arbre avec en bleu l’entropie comme critère d’impureté et en rouge l’indice de Gini. Jeu de données “20_0.9”.

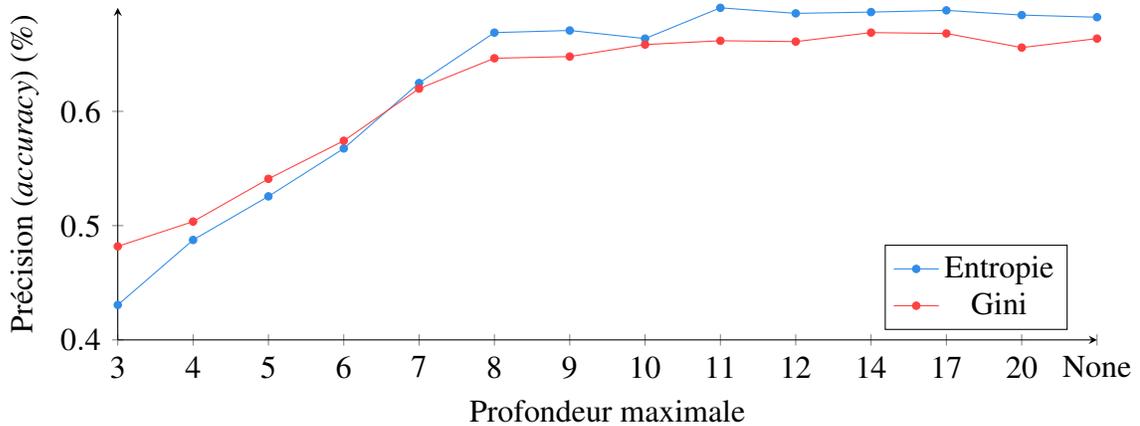


Figure 6.2 : Accuracy en fonction de la profondeur maximum de l'arbre avec en bleu l'entropie comme critère d'impureté et en rouge l'indice de Gini. Jeu de données "15_0.9".

Les paramètres de taille minimum de l'échantillon pour une séparation et de taille minimum de l'échantillon pour une feuille ont tous les deux été testés sur des combinaisons de 6 valeurs, respectivement sur les intervalles [2-10] et [1-10]. Aucune différence significative n'étant observable pour ces tests, seul un petit échantillon est représenté dans le Tableau 6.2. Il est difficile de déterminer une valeur optimale, on conservera donc les valeurs par défaut. Ainsi, on considérera par la suite que les valeurs optimales pour ces deux paramètres seront de 2 pour la taille minimum de l'échantillon pour une séparation, et de 1 pour la taille minimum de l'échantillon pour une feuille.

min_samples_split	min_samples_leaf	Accuracy	Cohen Kappa
2	1	0.72	0.68
2	3	0.71	0.68
2	5	0.73	0.69
2	10	0.72	0.68
5	1	0.72	0.68
5	3	0.72	0.68
5	5	0.72	0.69
5	10	0.71	0.67
10	1	0.71	0.67
10	3	0.72	0.68
10	5	0.72	0.68
10	10	0.71	0.67

Tableau 6.2 : Tableau représentant les indices d'évaluation *accuracy* et *Cohen kappa* selon la valeur des paramètres de taille minimale de l'échantillon pour une séparation et pour une feuille. Jeu de données "20_0.9".

Les différentes options pour limiter le nombre d'attributs à évaluer à chaque séparation montrent que, peu importe le jeu de données et les configurations des autres paramètres, ajouter une restriction réduit les performances. Ainsi, par la suite, on considère que ne pas limiter cette valeur est le paramétrage optimal. Le Tableau 6.3 représente un extrait des résultats de ces tests.

max_features	Accuracy	Cohen Kappa
log2	0.34	0.24
None	0.71	0.68
sqrt	0.43	0.35

Tableau 6.3 : Tableau représentant les indices d'évaluation *accuracy* et *Cohen kappa* selon les fonctions limitant le nombre d'attributs à évaluer à chaque séparation. Jeu de données "20_0.9".

Enfin, les variations sur le paramètre de réduction d'impureté montrent un faible impact sur les performances pour des petites valeurs, et une réduction des performances

légère pour des valeurs plus importantes. Ainsi, on estime qu'il est préférable de ne pas jouer sur ce critère pour le paramétrage. On admettra par la suite 0 comme la valeur optimale de ce paramètre. La Figure 6.3 représente un échantillon des résultats obtenus.

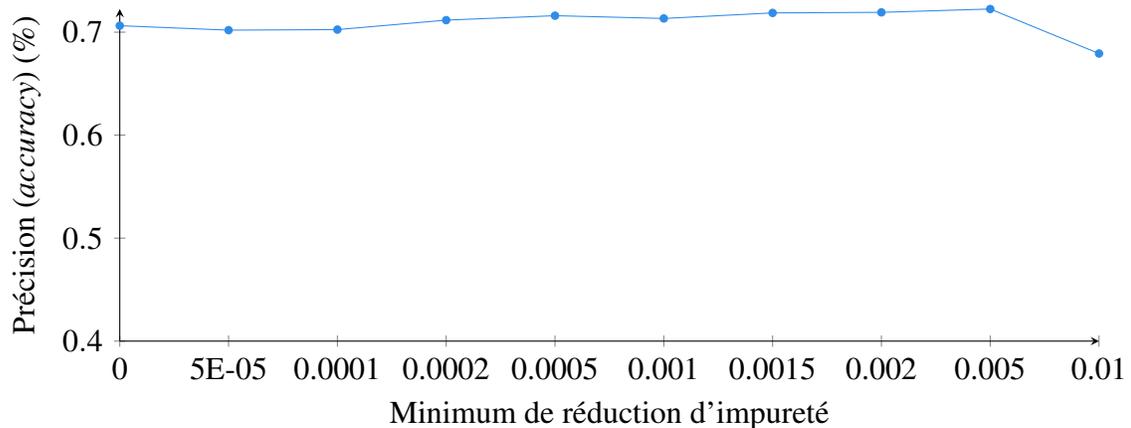


Figure 6.3 : Accuracy en fonction du seuil de réduction minimum d'impureté. Jeu de données "20_0.9".

Le paramétrage montre que la configuration optimale pour notre jeu de données est la suivante : `criterion = entropy`, `max_depth = 11`, `min_samples_split = 2`, `min_samples_leaf = 1`, `max_features = None`, `min_impurity_decrease = 0`. Afin de vérifier que cette configuration ne soit pas seulement la plus efficace sur les deux variations du jeu de données ayant servis au paramétrage, on effectue une série de tests sur toutes les variantes. Ici, la configuration optimale est comparée à la configuration par défaut évoquée précédemment, et à la configuration basique recommandée par la documentation de scikit-learn pour commencer le paramétrage²⁴. Cette configuration est la suivante : `criterion = gini`, `max_depth = 3`, `min_samples_split = 5`, `min_samples_leaf = 1`, `max_`

24. <https://scikit-learn.org/stable/modules/tree.html>

features = None, min_impurity_decrease = 0. Les résultats de ce test comparatif sont condensés dans le Tableau 6.4 avec plusieurs indices comme outil de comparaison. Ici, assez unanimement, la configuration optimale trouvée est toujours légèrement au-dessus de la configuration par défaut. Avec ce tableau, on peut également conclure que l'implémentation de l'algorithme CART avec la bibliothèque scikit-learn est la plus performante sur le jeu de données pré-traité avec une fenêtre temporelle de 20 secondes et un chevauchement de 0,9.

Configuration	Fenêtre	Chev.	Accuracy	Kappa	Precision	Recall	F1
Default	5	0.8	0.49	0.44	0.49	0.49	0.49
Optimal	5	0.8	0.51	0.46	0.5	0.51	0.5
Default	15	0.8	0.53	0.48	0.53	0.53	0.53
Optimal	15	0.8	0.59	0.54	0.58	0.59	0.58
Default	20	0.8	0.56	0.51	0.56	0.56	0.56
Optimal	20	0.8	0.56	0.51	0.56	0.56	0.56
Default	5	0.9	0.59	0.55	0.59	0.59	0.59
Optimal	5	0.9	0.61	0.57	0.61	0.61	0.6
Default	15	0.9	0.66	0.62	0.66	0.66	0.66
Optimal	15	0.9	0.69	0.65	0.69	0.69	0.68
Default	20	0.9	0.69	0.65	0.68	0.69	0.68
Optimal	20	0.9	0.71	0.67	0.7	0.71	0.7

Tableau 6.4 : Tableau comparatif de trois configurations de l'algorithme CART sur les 6 variations de pré-traitement du jeu de données.

6.1.3 FORÊT ALÉATOIRE

ALGORITHME ET PARAMÉTRAGE

Plusieurs techniques d'apprentissage d'ensemble sont implémentées dans la bibliothèque scikit-learn²⁵. Parmi elles, la méthode de la forêt aléatoire. Il existe une nuance dans la version de l'algorithme utilisée par la librairie par rapport à l'algorithme de la publication originale (Breiman, 2001). Pour chaque classe, la probabilité de détection de chaque arbre est sommée, et c'est la classe la plus probable qui l'emporte. Cela contraste avec le vote majoritaire où chaque arbre prédit une classe, et c'est la classe ayant été prédite par le plus grand nombre d'arbres qui prévaut. Plusieurs paramètres permettent d'influencer le comportement de l'algorithme, dont voici quelques exemples.

- Nombre d'arbres constituant la forêt (`n_estimators`) : Cela correspond à la taille de la forêt, c'est-à-dire au nombre d'arbres distincts la constituant.
- Critère d'impureté (`criterion`) : Identique à l'algorithme CART. C'est le choix de la fonction d'impureté utilisée lors du calcul du gain d'information. Il est possible d'utiliser la fonction d'entropie ou le critère de Gini.
- Profondeur maximale de l'arbre (`max_depth`) : Identique à l'algorithme CART. C'est la profondeur maximale, en nombre de nœuds, pour atteindre les feuilles depuis la racine des arbres. Ce paramètre peut soit être un entier correspondant à la profondeur, soit la valeur "aucune" qui ne limite donc pas la profondeur de l'arbre du tout.

25. <https://scikit-learn.org/stable/modules/ensemble.html>

- Taille minimum de l'échantillon pour une séparation (`min_samples_split`) : Identique à l'algorithme CART. C'est le nombre minimum d'instances dans un échantillon requis pour séparer un nœud. Ce paramètre peut prendre n'importe quel entier supérieur ou égal à 2 comme valeur.
- Taille minimum de l'échantillon pour une feuille (`min_samples_leaf`) : Identique à l'algorithme CART. C'est le nombre minimum d'instances dans un échantillon requis pour transformer le nœud suivant en feuille. Ce paramètre peut prendre n'importe quel entier supérieur ou égal à 1 comme valeur.
- Attributs maximums à considérer (`max_features`) : Nombre d'attributs à considérer lors du calcul de la meilleure séparation possible pour un nœud. Il est possible de sélectionner "sqrt" ou "log2" : le résultat du nombre total d'attributs à travers les fonctions racine carrée et log2 est alors considéré.
- Réduction minimum d'impureté (`min_impurity_decrease`) : Identique à l'algorithme CART. C'est le seuil minimum de réduction de l'impureté nécessaire pour créer un nœud lors de la considération de la sélection d'un attribut pour une séparation.

RÉSULTATS

De la même manière que précédemment, une première série de tests sur les différents jeux de données avec le paramétrage par défaut a été réalisée. La configuration de l'algorithme par défaut est la suivante : `n_estimators = 100`, `criterion = gini`, `max_depth = None`, `min_samples_split = 2`, `min_samples_leaf = 1`, `max_features = sqrt`,

$\text{min_impurity_decrease} = 0$. Le Tableau 6.5 condense les résultats principaux de ces tests. Ici, les résultats montrent une *accuracy* et un *Cohen kappa* nettement supérieurs pour les fenêtres avec un chevauchement de 0,9, plus particulièrement sur les fenêtres de 5 secondes et de 15 secondes. Ainsi, les tests de paramétrages seront effectués sur les jeux de données “5_0.9” et “15_0.9”, présentant les meilleurs scores dans ce tableau. Lorsque les résultats montrent une tendance claire et similaire sur les deux jeux, seuls les résultats du jeu de données “5_0.9” seront présentés.

Fenêtre	Chevauchement	Accuracy	Cohen Kappa
5	0.8	0.56	0.51
20	0.8	0.52	0.44
15	0.8	0.55	0.48
5	0.9	0.61	0.57
15	0.9	0.6	0.54
20	0.9	0.59	0.52

Tableau 6.5 : Tableau représentant les indices d'évaluation *accuracy* et *Cohen kappa* en fonction de la taille des fenêtres temporelles (time window) et du chevauchement (overlap) des jeux de données. Algorithme random forest.

L'un des paramètres principaux de cette méthode est le nombre d'arbres constituant la forêt. Plus le nombre d'arbres et leur complexité sont croissants, plus l'algorithme est long à entraîner et utiliser. La Figure 6.4 montre l'impact de ce paramètre sur les performances. On observe un palier pour 100 arbres, quoi qu'il soit possible que 75 arbres suffisent étant donné la faible différence. Par précaution, on considère que 100 arbres est la valeur optimale.

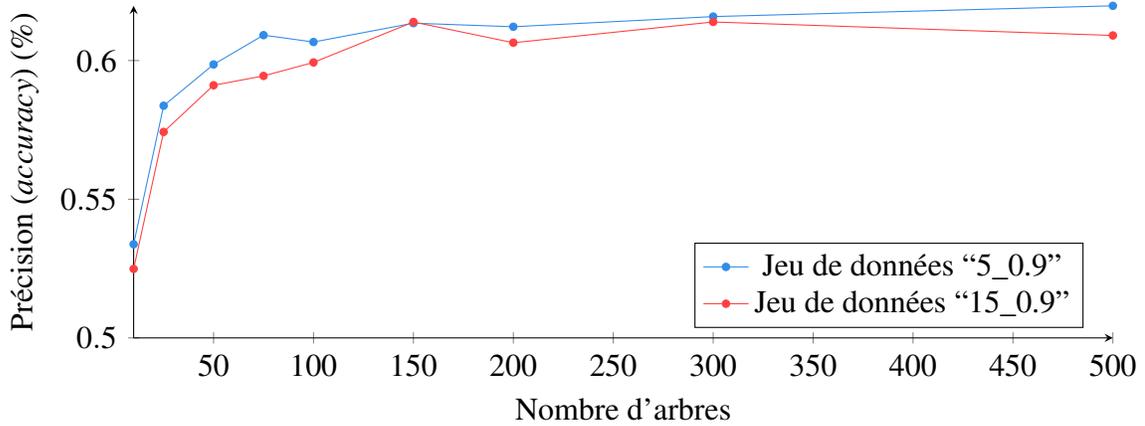


Figure 6.4 : *Accuracy* en fonction du nombre d'arbres constituant la forêt, avec en bleu le jeu de données "5_0.9" et en rouge le jeu de données "15_0.9".

Comme pour la sous-section précédente, on cherche à définir le critère de calcul de l'impureté et la profondeur de l'arbre optimaux. Les Figures 6.5 et 6.6 résument le résultat des tests effectués. Ici, l'écart entre l'entropie et l'indice de Gini est très faible. Cependant, le premier l'emporte légèrement sur le second. De plus, on ne semble pas atteindre un palier net de la croissance du critère d'*accuracy* avant une profondeur d'arbre maximale de 20. On note tout de même que l'écart est très faible entre cette profondeur et une profondeur sans limite. Étant donné que cette méthode repose sur des arbres très spécifiques, on considère optimal de n'attribuer aucune limite à la profondeur des arbres.

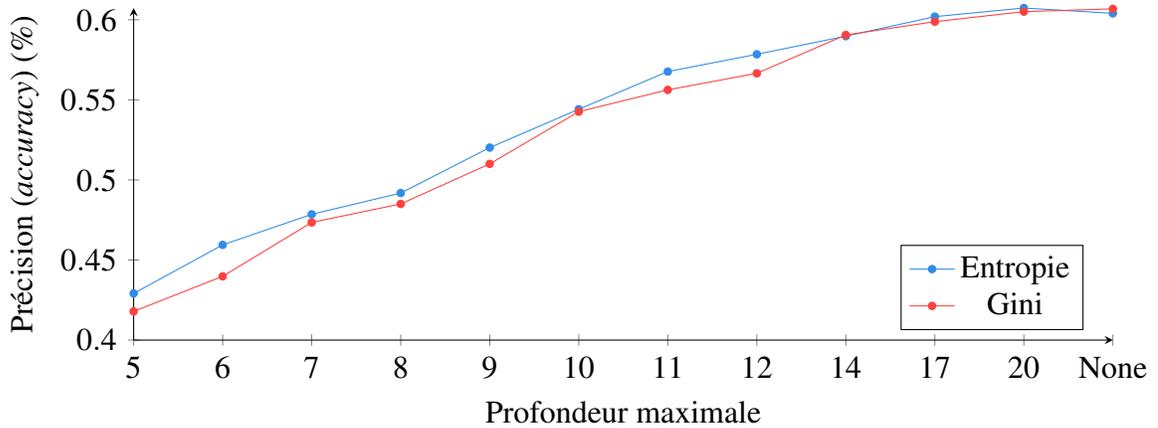


Figure 6.5 : Accuracy en fonction de la profondeur maximale avec en bleu l'entropie comme critère d'impureté et en rouge l'indice de Gini. Jeu de données "5_0.9".

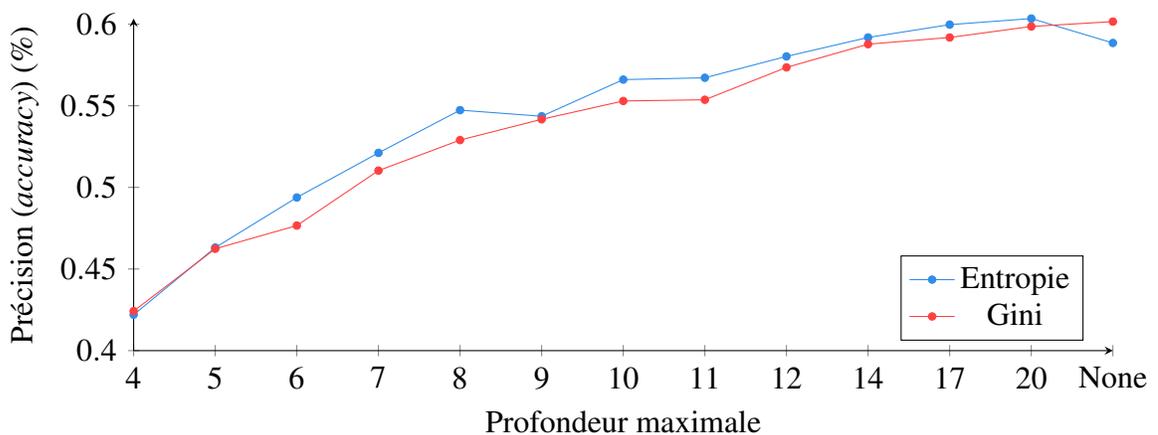


Figure 6.6 : Accuracy en fonction de la profondeur, maximale avec en bleu l'entropie comme critère d'impureté et en rouge l'indice de Gini. Jeu de données "15_0.9".

Les paramètres de taille minimum de l'échantillon pour une séparation et de taille minimum de l'échantillon pour une feuille ont tous les deux été au centre d'une série de tests. Cependant, comme pour l'arbre de décision seul, leur variation n'a montrée aucun impact significatif. Ainsi, leurs valeurs seront laissées par défaut dans la configuration

optimale des paramètres.

La limitation du nombre d'attributs à évaluer à chaque séparation est centrale au fonctionnement de la méthode de la forêt aléatoire. Ainsi, il n'est pas possible de ne pas mettre de limite, comme cela a été fait dans la sous-section précédente. Ici, deux options principales sont disponibles : la racine carrée du nombre total d'attributs ou le log2 de celui-ci. Le Tableau 6.6 montre clairement que la fonction racine carrée est plus efficace, c'est donc ce paramètre qui sera conservé dans la configuration optimale.

max_features	Fenêtre	Accuracy	Cohen Kappa
log2	5	0.53	0.47
log2	15	0.49	0.4
sqrt	5	0.61	0.57
sqrt	15	0.61	0.55

Tableau 6.6 : Tableau représentant les indices d'évaluation *accuracy* et *Cohen kappa* selon les fonctions restreignant le nombre d'attributs à évaluer à chaque séparation. Tests sur les jeux de données "5_0.9" et "15_0.9".

Enfin, le paramètre de réduction d'impureté semble encore une fois avoir peu d'impact sur les performances pour les valeurs faibles, et il semble être nocif pour des valeurs plus élevées. Ainsi, on considérera 0 en tant que valeur optimale pour ce paramètre. La Figure 6.7 représente un échantillon des résultats obtenus.

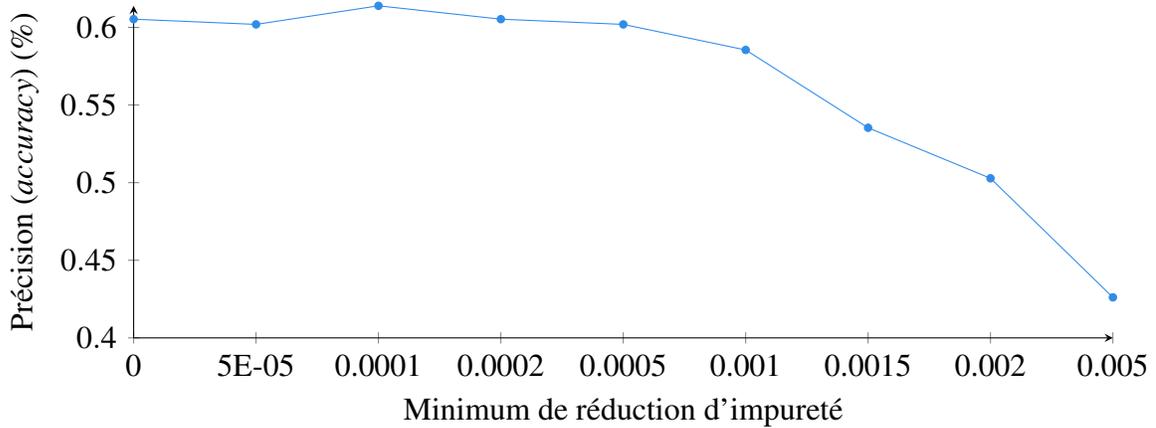


Figure 6.7 : Accuracy en fonction du seuil de réduction minimum d'impureté. Jeu de données "5_0.9"

Après plusieurs séries de tests, on obtient la configuration optimale suivante : `cn_estimators = 100`, `criterion = entropy`, `max_depth = None`, `min_samples_split = 2`, `min_samples_leaf = 1`, `max_features = sqrt`, `min_impurity_decrease = 0`. Toujours dans l'optique de vérifier que cette configuration soit efficace sur toutes les variations du jeu de données, on la compare à la configuration par défaut sur l'ensemble de celles-ci. La configuration optimale est également comparée à sa copie conforme, à l'exception de la profondeur des arbres limitée à 20. Les résultats sont représentés en Figure 6.8. On observe que, peu importe la configuration, ce sont les profils de pré-traitement "5_0.9" et "15_0.9" qui sont les plus efficaces pour cette méthode d'apprentissage. On observe aussi que la configuration par défaut est équivalente aux deux variantes de la configuration optimale. Cela s'explique par le très faible écart observé entre les critères d'impureté, seul facteur discriminant entre la configuration par défaut et la configuration optimale. De même, la profondeur maximale de l'arbre, seul paramètre variant entre les deux versions de la configuration optimale, n'a pas semblé avoir beaucoup d'impact sur

les résultats lors du paramétrage. Ces tests confirment la validité de l’hypothèse selon laquelle les deux valeurs sont équivalentes sur l’ensemble des jeux de données.

Configuration	Fenêtre	Chev.	Accuracy	Kappa	Precision	Recall	F1
Default	5	0.8	0.55	0.5	0.58	0.55	0.51
Optimal	5	0.8	0.55	0.5	0.59	0.55	0.5
Optimal 20	5	0.8	0.56	0.5	0.6	0.56	0.51
Default	15	0.8	0.52	0.45	0.49	0.52	0.46
Optimal	15	0.8	0.52	0.45	0.53	0.52	0.46
Optimal 20	15	0.8	0.53	0.46	0.49	0.53	0.47
Default	20	0.8	0.51	0.43	0.43	0.51	0.42
Optimal	20	0.8	0.51	0.43	0.56	0.51	0.43
Optimal 20	20	0.8	0.51	0.43	0.49	0.51	0.43
Default	5	0.9	0.6	0.56	0.63	0.6	0.57
Optimal	5	0.9	0.61	0.56	0.64	0.61	0.57
Optimal 20	5	0.9	0.61	0.56	0.64	0.61	0.58
Default	15	0.9	0.61	0.55	0.6	0.61	0.56
Optimal	15	0.9	0.6	0.54	0.61	0.6	0.54
Optimal 20	15	0.9	0.61	0.55	0.61	0.61	0.56
Default	20	0.9	0.6	0.54	0.63	0.6	0.55
Optimal	20	0.9	0.6	0.53	0.61	0.6	0.54
Optimal 20	20	0.9	0.59	0.52	0.6	0.59	0.52

Figure 6.8 : Tableau comparatif de trois configurations pour la méthode de forêt aléatoire sur les 6 variations de pré-traitement du jeu de données.

6.1.4 ADABOOST

ALGORITHME ET PARAMÉTRAGE

La bibliothèque scikit-learn implémente également une version de la méthode AdaBoost²⁶. Plusieurs paramètres permettent d’influencer le comportement de l’algo-

26. <https://scikit-learn.org/stable/modules/ensemble.html>

rithme. Parmi ceux-ci, on retrouve tous les paramètres propres aux arbres de décision vus précédemment. Ici, on ne considérera que le critère d'impureté et la profondeur de l'arbre, en plus des paramètres propres à AdaBoost. Voici la liste des paramètres qui ont été testés.

- Nombre d'arbres constituant le groupe (`n_estimators`) : Identique à la méthode de la forêt aléatoire. Il correspond à la taille du groupe d'arbres du modèle.
- Critère d'impureté (`criterion`) : Identique à l'algorithme CART. C'est le choix de la fonction d'impureté utilisée lors du calcul du gain d'information. Il est possible d'utiliser la fonction d'entropie ou le critère de Gini.
- Profondeur maximale de l'arbre (`max_depth`) : Identique à l'algorithme CART. C'est la profondeur maximale, en nombre de nœuds, pour atteindre les feuilles depuis la racine des arbres. Ce paramètre peut soit être un entier correspondant à la profondeur, soit la valeur "aucune" qui ne limite donc pas la profondeur de l'arbre du tout.
- Taux d'apprentissage (`learning_rate`) : C'est un nombre réel caractérisant l'impact de chaque itération dans le processus. Un taux d'apprentissage faible rend les arbres successifs peu orientés vis-à-vis de leurs prédécesseurs. Il doit ainsi être compensé par un nombre d'arbres important afin de rendre la technique efficace.

RÉSULTATS

Pour déterminer la variation du jeu de données la plus propice au paramétrage, une série de tests sur les différents profils de pré-traitement a été réalisée. La configuration

par défaut de l’algorithme est la suivante : $n_estimators = 50$, $criterion = gini$, $max_depth = 1$, $learning_rate = 1$. Le Tableau 6.7 condense les résultats de ces tests. Ici, les indices de performance montrent une *accuracy* et un *Cohen kappa* supérieurs pour les fenêtres avec un chevauchement de 0,9, et de meilleurs résultats avec une fenêtre de 5 secondes. Ainsi, les tests de paramétrages seront effectués sur le jeu de données “5_0.9”, présentant les meilleurs scores dans ce tableau. Les résultats obtenus à l’aide du jeu de données “15_0.9” seront également présentés.

Fenêtre	Chevauchement	Accuracy	Cohen Kappa
5	0.8	0.32	0.21
15	0.8	0.32	0.2
20	0.8	0.24	0.1
5	0.9	0.37	0.31
15	0.9	0.26	0.13
20	0.9	0.27	0.13

Tableau 6.7 : Tableau représentant les indices d’évaluation *accuracy* et *Cohen kappa* selon la taille des fenêtres temporelles (time window) et le chevauchement (overlap) des jeux de données. Algorithme AdaBoost.

La première série de tests pour le paramétrage vise à déterminer le taux d’apprentissage et le nombre d’arbres optimaux. Des itérations sont faites sur des valeurs allant de 0.1 à 5 pour le taux d’apprentissage, et de 10 à 100 pour le nombre d’arbres. La Figure 6.9 montre les résultats sur la variation du jeu de données “5_0.9”. Les performances sont assez médiocres en général, mais sont particulièrement mauvaises lorsque le taux d’apprentissage s’éloigne de 1. Le nombre d’arbres semble, étonnamment, avoir assez peu d’importance. La Figure 6.10 confirme ces résultats avec une ambiguïté sur le taux d’apprentissage de 1,5 qui semble être plutôt efficace. Face à l’incertitude qu’entraînent ces résultats, on considère que les valeurs par défaut sont les valeurs optimales,

c'est-à-dire 50 arbres pour la taille du groupe et 1 pour le taux d'apprentissage.

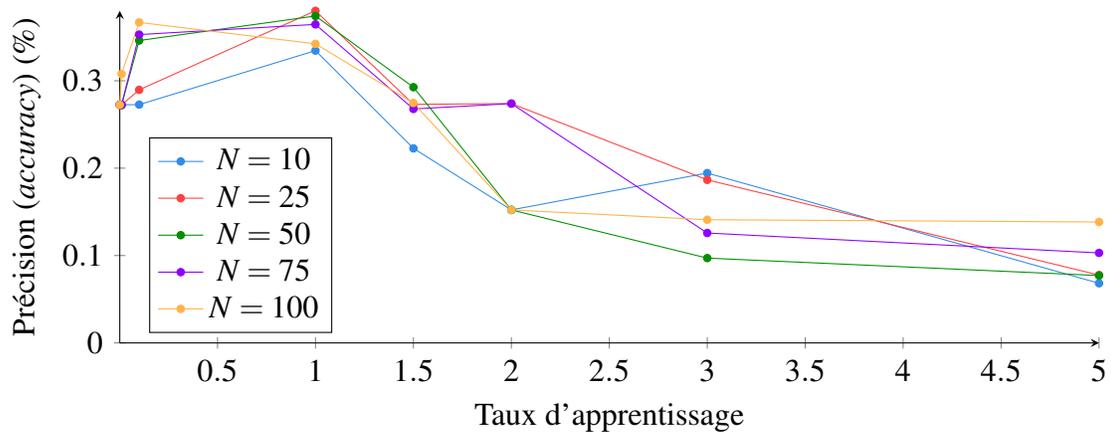


Figure 6.9 : Accuracy en fonction du taux d'apprentissage avec en bleu 10 arbres, en rouge 25 arbres, en vert 50 arbres, en violet 75 arbres et en orange 100 arbres par groupe. Jeu de données "5_0.9".

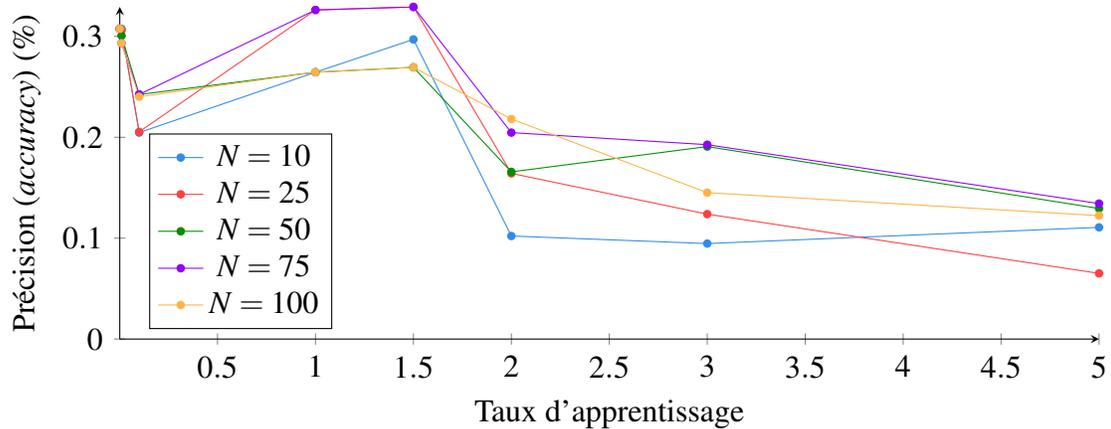


Figure 6.10 : Accuracy en fonction du taux d'apprentissage avec en bleu 10 arbres, en rouge 25 arbres, en vert 50 arbres, en violet 75 arbres et en orange 100 arbres par groupe. Jeu de données "15_0.9".

En écho aux deux sous-sections précédentes, on cherche à définir le critère de calcul de l'impureté et la profondeur de l'arbre optimums. Les résultats des tests sont

représentés sur les Figures 6.11 et 6.12. Comme pour les deux autres méthodes d'apprentissage machine, on observe une légère prévalence du critère d'entropie sur l'indice de Gini. Cependant, pour la profondeur maximale, aucun plateau n'est observé. De plus, un pic significatif d'amélioration des performances est présent lorsqu'aucune limite n'est donnée. Cela rentre en conflit avec le principe d'AdaBoost qui est normalement basé sur des arbres "troncs", c'est-à-dire très peu profonds. Cependant, il est difficile d'ignorer une telle amélioration des performances. Ainsi, les critères optimums seront l'entropie pour le critère d'impureté, et aucune limite pour la profondeur des arbres.

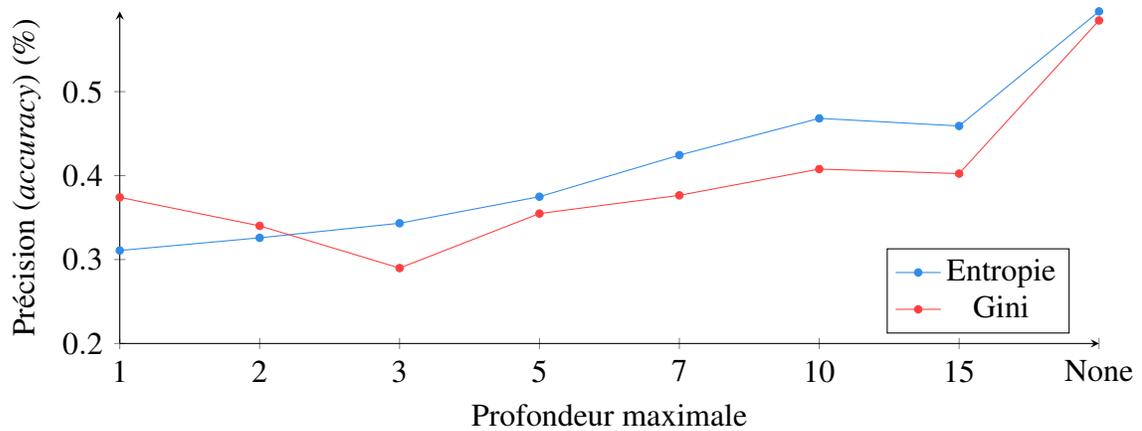


Figure 6.11 : Accuracy en fonction de la profondeur maximale, avec en bleu l'entropie comme critère d'impureté et en rouge l'indice de Gini. Jeu de données "5_0.9".

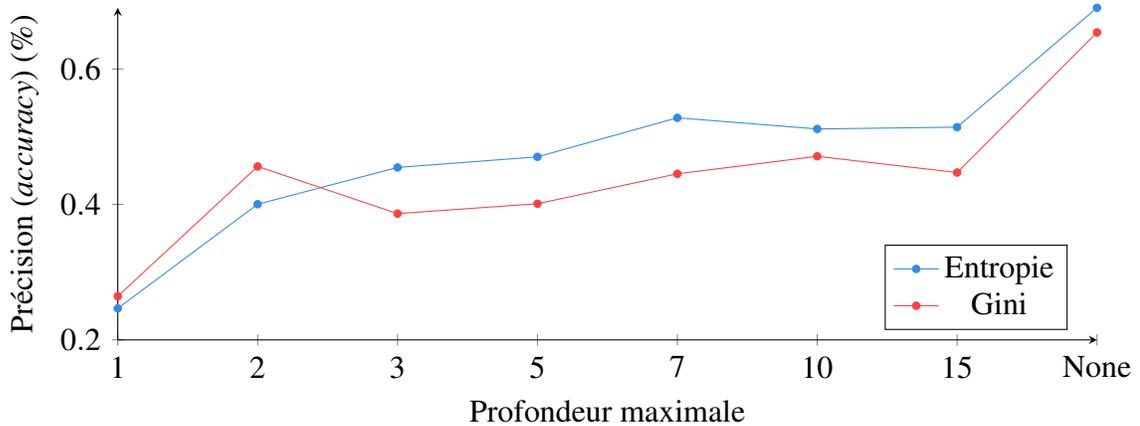


Figure 6.12 : Accuracy en fonction de la profondeur maximale, avec en bleu l’entropie comme critère d’impureté et en rouge l’indice de Gini. Jeu de données “15_0.9”.

Après la série de tests pour le paramétrage, on obtient la configuration optimale suivante : `n_estimators = 50`, `criterion = entropy`, `max_depth = None`, `learning_rate = 1`. Afin de confirmer l’optimalité de cette configuration sur toutes les variations du jeu de données, elle sera comparée à la configuration par défaut. Afin de lever le doute sur une potentielle efficacité d’un taux d’apprentissage à 1,5, une variante de la configuration optimale avec cette valeur sera également testée. Les résultats sont représentés dans le Tableau 6.8. La configuration optimale semble être largement plus performante que la version par défaut. On observe également que la version avec un taux d’apprentissage à 1,5 est sensiblement équivalente à la version optimale standard. Enfin, on note qu’avec la configuration optimale, AdaBoost est nettement plus performant sur les jeux de données “15_0.9” et “20_0.9”, là où avec la configuration par défaut c’était le jeu de données “5_0.9” qui prévalait.

Configuration	Fenêtre	Chev.	Accuracy	Kappa	Precision	Recall	F1
Default	5	0.8	0.32	0.21	0.23	0.32	0.23
Optimal	5	0.8	0.47	0.42	0.47	0.47	0.47
Optimal 1.5	5	0.8	0.47	0.42	0.47	0.47	0.47
Default	15	0.8	0.32	0.2	0.2	0.32	0.21
Optimal	15	0.8	0.57	0.52	0.58	0.57	0.57
Optimal 1.5	15	0.8	0.58	0.53	0.58	0.58	0.58
Default	20	0.8	0.25	0.11	0.16	0.25	0.11
Optimal	20	0.8	0.57	0.51	0.58	0.57	0.57
Optimal 1.5	20	0.8	0.55	0.5	0.56	0.55	0.55
Default	5	0.9	0.37	0.31	0.34	0.37	0.33
Optimal	5	0.9	0.59	0.55	0.59	0.59	0.59
Optimal 1.5	5	0.9	0.59	0.55	0.6	0.59	0.59
Default	15	0.9	0.26	0.13	0.16	0.26	0.15
Optimal	15	0.9	0.68	0.64	0.68	0.68	0.68
Optimal 1.5	15	0.9	0.69	0.65	0.68	0.69	0.68
Default	20	0.9	0.27	0.13	0.22	0.27	0.14
Optimal	20	0.9	0.71	0.67	0.71	0.71	0.71
Optimal 1.5	20	0.9	0.71	0.67	0.71	0.71	0.71

Tableau 6.8 : Tableau comparatif de trois configurations de l’algorithme AdaBoost sur les 6 variations de pré-traitement du jeu de données.

6.1.5 K PLUS PROCHE VOISINS

ALGORITHME ET PARAMÉTRAGE

Sous scikit-learn, La méthode des K plus proches voisins est implémenté sous la forme de plusieurs algorithmes différents²⁷. Afin de faciliter le paramétrage, la fonctionnalité d’attribution automatique sera utilisée. Ainsi, en analysant le format des données, la bibliothèque utilise automatiquement l’algorithme le plus pertinent, c’est-à-

27. <https://scikit-learn.org/stable/modules/neighbors.html>

dire soit Ball Tree, soit KD Tree, soit Brute Force. Le paramétrage s'effectuera sur les 4 paramètres distincts ci-dessous.

- Nombre de voisins (`n_neighbors`) : C'est le nombre de voisins à considérer pour déterminer la classe de l'instance inconnue. Ce paramètre peut prendre n'importe quelle valeur entière supérieure ou égale à 1.
- Poids (`weights`) : C'est la distribution des poids utilisés dans la prédiction. Cette distribution peut soit être uniforme, soit liée à la distance pour que les voisins les plus proches aient plus d'impact que ceux éloignés dans le cercle des K voisins.
- Taille des feuilles (`leaf_size`) : Paramètre propre aux variations Ball Tree et KD Tree de l'algorithme. Peut prendre toute valeur entière supérieure ou égale à 1.
- Paramètre de Minkowski (`p`) : Paramètre de la fonction de distance de Minkowski. Lorsque `p` est égal à 1, cela revient à la fonction de distance de Manhattan. Peut prendre toute valeur entière supérieure ou égale à 1.

RÉSULTATS

Comme précédemment, on effectue une première série de tests pour déterminer la variation du jeu de données la plus propice au paramétrage. La configuration par défaut de l'algorithme est la suivante : `n_neighbors = 5`, `weights = uniform`, `leaf_size = 30`, `p = 2`. Le Tableau 6.9 condense les résultats de ces tests. On observe une *accuracy* et un *Cohen kappa* supérieurs pour les fenêtres possédants un chevauchement de 0.9, avec une emphase pour la fenêtre de 20 secondes. Ainsi, les tests de paramétrages seront effectués sur le jeu de données "15_0.9" et "20_0.9", présentant les meilleurs scores

dans ce tableau. Lorsque les résultats sont particulièrement identiques, seuls ceux sur le jeu de données “20_0.9” seront présentés.

Fenêtre	Chevauchement	Accuracy	Cohen Kappa
5	0.8	0.49	0.43
15	0.8	0.51	0.45
20	0.8	0.54	0.48
5	0.9	0.56	0.51
15	0.9	0.59	0.54
20	0.9	0.63	0.58

Tableau 6.9 : Tableau représentant les indices d'évaluation *accuracy* et *Cohen kappa* selon la taille des fenêtres temporelles (time window) et le chevauchement (overlap) des jeux de données. Algorithme KNN.

La première série de tests pour le paramétrage vise à déterminer le nombre optimal de voisins à prendre en compte, ainsi que la meilleure méthode de répartition des poids pour ces voisins. On fait ici varier le nombre de voisins entre 1 et 20, et on utilise les deux critères disponibles pour la répartition des poids ("uniform" et "distance"). Les Figures 6.13 et 6.14 synthétisent les résultats obtenus. Le premier graphique montre des performances optimales pour un K de 1. Ainsi, la fonction de répartition des poids importe peu. Cependant, pour tout K supérieur à 1, on observe que la répartition des poids selon la fonction de distance est plus efficace. C'est donc cette méthode qui sera conservée pour la configuration optimale. Le deuxième graphique crée une ambiguïté sur le nombre de voisins optimal. En suivant la répartition selon la fonction de distance, on observe un optimum aux alentours de 5 voisins, la légère variation entre les valeurs proches n'étant pas significative. Ainsi, la valeur de K égale à 5 sera conservée pour la suite des tests, tout en considérant la potentialité qu'un K de 1 soit optimal.

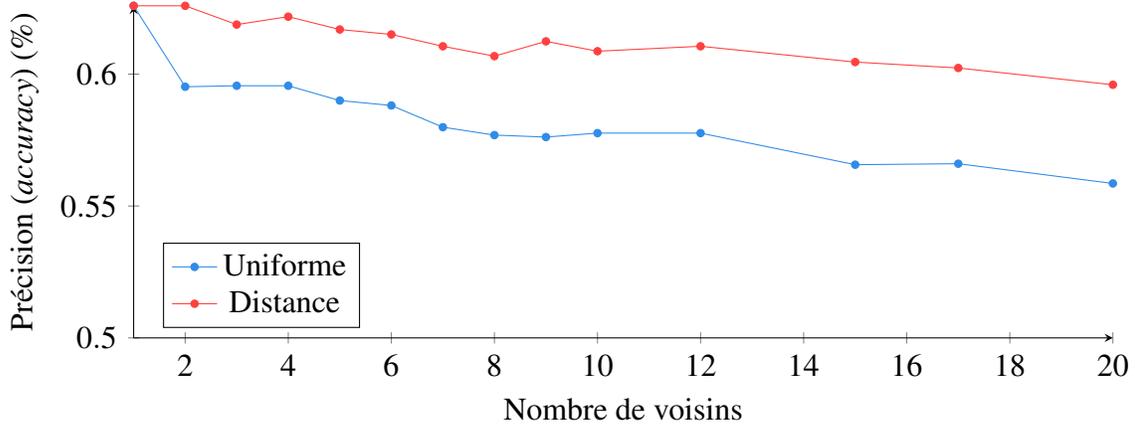


Figure 6.13 : Accuracy en fonction du nombre de voisins, avec en bleu une répartition des poids uniforme et en rouge une répartition des poids selon la fonction de distance. Jeu de données “15_0.9”.

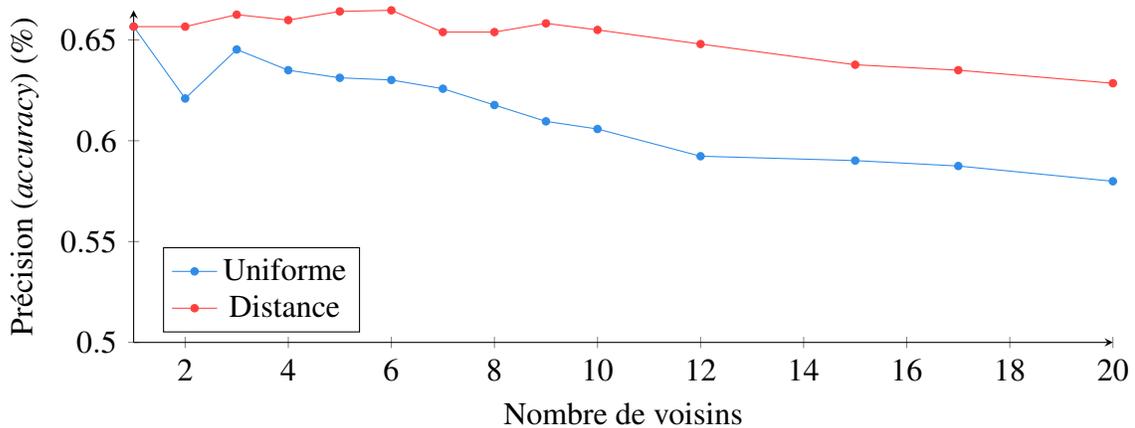


Figure 6.14 : Accuracy en fonction du nombre de voisins, avec en bleu une répartition des poids uniforme et en rouge une répartition des poids selon la fonction de distance. Jeu de données “20_0.9”.

Les tests effectués sur le paramètre de la taille des feuilles ne montrent aucune différence dans les résultats. On en conclut alors que c’est la version Brute Force de l’algorithme qui est utilisée. En revanche, la variation de la valeur p de la distance de Minkowski montre des résultats intéressants. Un graphique représentant les résultats

obtenus est disponible en Figure 6.15. Ici, c'est un p de 1 qui est optimal, ainsi la fonction de distance la plus efficace est la distance de Manhattan. C'est ce paramétrage qui sera conservé par la suite.

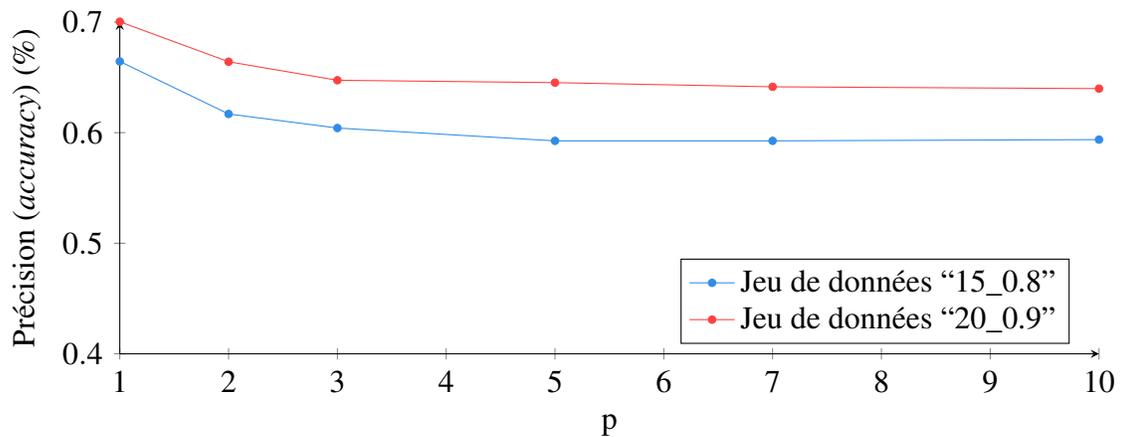


Figure 6.15 : Accuracy en fonction de la valeur de p avec le jeu de données "15_0.8" en bleu et le jeu de données "20_0.9" en rouge.

Après les tests de paramétrage, on obtient la configuration optimale suivante : $n_neighbors = 5$, $weights = distance$, $leaf_size = 30$, $p = 1$. Elle sera comparée à la configuration par défaut ainsi qu'à une variation de la configuration optimale avec un K de 1. Les résultats sont représentés dans le tableau 6.8. Peu importe le paramétrage, l'algorithme KNN semble être le plus performant sur le jeu de données "20_0.9", et, de manière générale, sur les jeux de données avec un chevauchement de 0,9. Les meilleurs scores sont obtenus avec la configuration optimale avec un K de 1, bien que la différence soit très légère avec la configuration optimale standard. De plus, cette dernière est plus efficace dans la majorité des autres jeux de données. Ainsi, les deux configurations semblent relativement équivalentes.

Configuration	Fenêtre	Chev.	Accuracy	Kappa	Precision	Recall	F1
Default	5	0.8	0.49	0.43	0.47	0.49	0.47
Optimal	5	0.8	0.52	0.46	0.51	0.52	0.5
Optimal 1	5	0.8	0.49	0.44	0.49	0.49	0.49
Default	15	0.8	0.51	0.45	0.49	0.51	0.5
Optimal	15	0.8	0.58	0.53	0.57	0.58	0.57
Optimal 1	15	0.8	0.54	0.49	0.54	0.54	0.54
Default	20	0.8	0.54	0.48	0.53	0.54	0.53
Optimal	20	0.8	0.57	0.51	0.57	0.57	0.56
Optimal 1	20	0.8	0.59	0.53	0.59	0.59	0.58
Default	5	0.9	0.56	0.51	0.55	0.56	0.55
Optimal	5	0.9	0.61	0.57	0.61	0.61	0.61
Optimal 1	5	0.9	0.62	0.58	0.62	0.62	0.62
Default	15	0.9	0.59	0.54	0.58	0.59	0.58
Optimal	15	0.9	0.66	0.62	0.66	0.66	0.66
Optimal 1	15	0.9	0.68	0.64	0.68	0.68	0.68
Default	20	0.9	0.63	0.58	0.61	0.63	0.62
Optimal	20	0.9	0.7	0.66	0.7	0.7	0.69
Optimal 1	20	0.9	0.71	0.67	0.71	0.71	0.71

Tableau 6.10 : Tableau comparatif de trois configurations de l’algorithme KNN sur les 6 variations de pré-traitement du jeu de données.

6.1.6 ANALYSE COMPARATIVE

Afin de déterminer la méthode d’apprentissage machine testée la plus efficace pour le jeu de données des activités, on compare chacune de leurs configurations optimales sur les trois profils de pré-traitement avec un chevauchement de 0,9. Le Tableau 6.11 fait état de cette comparaison. Bien que la méthode de la forêt aléatoire ait des performances équivalentes aux autres méthodes pour le jeu de données “5_0.9”, elle ne parvient pas à maintenir celles-ci pour les autres jeux de données. On observe que les algorithmes CART, AdaBoost et KNN ont tous les trois des performances équivalentes, peu importe

le jeu de données utilisé. Avec une *accuracy* d'environ 71% et un *Cohen kappa* d'environ 67% dans le meilleur cas, elles semblent assez efficaces compte tenu du fait que le jeu de données possède 15 classes distinctes, dont certaines très proches les unes des autres (boire et prendre des médicaments, faire la vaisselle et se laver les mains, ect.). Bien que les temps d'exécutions n'aient pas été inclus dans cette étude, la méthode KNN a été nettement plus rapide lors des phases d'entraînement et de tests que les autres méthodes. Ainsi, compte tenu des performances équivalentes, c'est KNN qui semble être l'algorithme le plus adapté pour notre problématique.

Méthode	Configuration	Fen. 1	Accuracy	Kappa	Precision	Recall	F1
CART	Optimal	5	0.61	0.57	0.61	0.61	0.6
Random forest	Optimal	5	0.61	0.56	0.64	0.61	0.57
AdaBoost	Optimal	5	0.59	0.55	0.59	0.59	0.59
KNN	Optimal 1	5	0.62	0.58	0.62	0.62	0.62
CART	Optimal	15	0.69	0.65	0.69	0.69	0.68
Random forest	Optimal	15	0.6	0.54	0.61	0.6	0.54
AdaBoost	Optimal	15	0.68	0.64	0.68	0.68	0.68
KNN	Optimal 1	15	0.68	0.64	0.68	0.68	0.68
CART	Optimal	20	0.71	0.67	0.7	0.71	0.7
Random forest	Optimal	20	0.6	0.53	0.61	0.6	0.54
AdaBoost	Optimal	20	0.71	0.67	0.71	0.71	0.71
KNN	Optimal 1	20	0.71	0.67	0.71	0.71	0.71

Tableau 6.11 : Tableau comparatif des quatres méthodes d'apprentissage machine testées sur 3 des 6 variations de pré-traitement du jeu de données.

6.2 JEUX DE DONNÉES À DOUBLE FENÊTRES

6.2.1 INTRODUCTION AUX JEUX DE DONNÉES

L'objectif de cette section est de définir la pertinence d'ajouter une seconde fenêtre temporelle, plus petite, pour chaque série temporelle extraite du jeu de données. Par

exemple, pour la variation du jeu de données “15_0.9”, on pourrait considérer l’ajout d’une seconde fenêtre temporelle de 5 secondes centrée sur la première. L’illustration de cet exemple est disponible en Figure 6.16. Le même traitement est appliqué identiquement pour les deux fenêtres temporelles, c’est-à-dire que les mêmes caractéristiques sont extraites sur les deux séries temporelles distinctes. Là où précédemment, on avait trois vecteurs de caractéristiques (un par radar), on en a maintenant 6. Les trois premiers correspondent aux caractéristiques des séries temporelles longues, et les trois autres correspondent aux caractéristiques des séries temporelles courtes. Cet ajout a pour objectif d’apporter un contexte plus instantané aux données. L’enrichissement des jeux de données pourrait alors se retranscrire par un gain de performance sur l’une ou plusieurs des méthodes testées auparavant. Ainsi, 6 nouvelles variantes du jeu de données ont été créées avec ce nouveau pré-traitement : “5_0.9_1”, “5_0.9_2”, “15_0.9_5”, “15_0.9_7”, “20_0.9_5” et “20_0.9_10”. Ici, le premier nombre correspond à la durée, en secondes, de la fenêtre temporelle principale. Le dernier nombre correspond à la durée de la deuxième fenêtre temporelle. On note que le poids des jeux de données pré-traités augmente en conséquence et varie entre 250Mo pour le plus léger, et 340Mo pour le plus lourd.

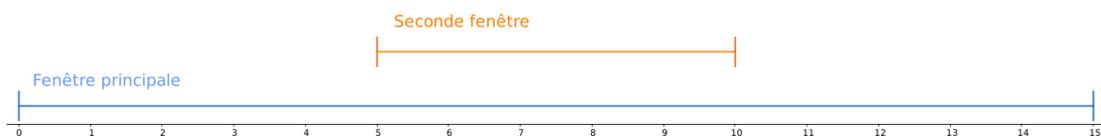


Figure 6.16 : Représentation d’une fenêtre secondaire de 5 secondes centrée sur une fenêtre principale de 15 secondes.

6.2.2 ARBRES DE DÉCISIONS

L'algorithme CART a été testé sur ces nouveaux jeux de données avec les configurations "Default" et "Optimal" décrites dans la section précédente. Les résultats sont synthétisés dans le Tableau 6.12. Cette méthode ne semble pas bénéficier d'un quelconque gain de performance quand elle est combinée avec cette nouvelle étape de pré-traitement. On observe même quelques réductions de performances pour certains cas. L'algorithme CART est donc plus performant avec le pré-traitement original.

Configuration	Fen. 1	Chev.	Fen. 2.	Accuracy	Kappa	Precision	Recall	F1
Default	5	0.8		0.49	0.44	0.49	0.49	0.49
Optimal	5	0.8		0.51	0.46	0.5	0.51	0.5
Default	5	0.9		0.59	0.55	0.59	0.59	0.59
Optimal	5	0.9		0.61	0.57	0.61	0.61	0.6
Default	5	0.9	1	0.58	0.53	0.58	0.58	0.58
Optimal	5	0.9	1	0.6	0.56	0.6	0.6	0.59
Default	5	0.9	2	0.57	0.53	0.57	0.57	0.57
Optimal	5	0.9	2	0.61	0.57	0.61	0.61	0.6
Default	15	0.8		0.53	0.48	0.53	0.53	0.53
Optimal	15	0.8		0.59	0.54	0.58	0.59	0.58
Default	15	0.9		0.66	0.62	0.66	0.66	0.66
Optimal	15	0.9		0.69	0.65	0.69	0.69	0.68
Default	15	0.9	5	0.63	0.59	0.63	0.63	0.63
Optimal	15	0.9	5	0.68	0.64	0.68	0.68	0.68
Default	15	0.9	7	0.67	0.63	0.68	0.67	0.67
Optimal	15	0.9	7	0.69	0.66	0.69	0.69	0.69
Default	20	0.8		0.56	0.51	0.56	0.56	0.56
Optimal	20	0.8		0.56	0.51	0.56	0.56	0.56
Default	20	0.9		0.69	0.65	0.68	0.69	0.68
Optimal	20	0.9		0.71	0.67	0.7	0.71	0.7
Default	20	0.9	5	0.68	0.64	0.68	0.68	0.68
Optimal	20	0.9	5	0.7	0.66	0.69	0.7	0.69
Default	20	0.9	10	0.67	0.63	0.67	0.67	0.67
Optimal	20	0.9	10	0.7	0.66	0.69	0.7	0.7

Tableau 6.12 : Tableau représentant plusieurs indices de performance selon la taille des fenêtres temporelles (time window) et le chevauchement (overlap) des jeux de données. Algorithme CART.

6.2.3 FORÊT ALÉATOIRE

L'algorithme de forêt aléatoire, quant à lui, a été testé sur ces nouveaux jeux de données avec les trois configurations établies dans la section précédente, "Default" et "Optimal" et "Optimal 20". Les résultats sont synthétisés dans le Tableau 6.13. Contrairement à CART, l'algorithme de forêt aléatoire semble bénéficier d'un léger gain de performance lorsqu'il est entraîné avec ces nouveaux jeux de données. Les deux

nouvelles variantes autour de la fenêtre temporelle de 5 secondes semblent équivalentes mais légèrement supérieures à la version simple fenêtre. La variante "15_0.9_7" semble nettement au dessus de la variante "15_0.9_5", elle-même plus efficace que la version simple fenêtre. Enfin, les deux nouvelles variantes autour de la fenêtre de 20 secondes semblent équivalentes mais montrent aussi un léger gain par rapport à la version simple fenêtre. Cette nouvelle série de tests nous permet de conclure que l'algorithme CART est plus performant de 4% selon l'indice d'*accuracy* avec la combinaison "Optimal 20" - "15_0.9_7" par rapport à la meilleur combinaison précédente, "Optimal 20" - "15_0.9".

Configuration	Fen. 1	Chev.	Fen. 2.	Accuracy	Kappa	Precision	Recall	F1
Default	5	0.8		0.55	0.5	0.58	0.55	0.51
Optimal	5	0.8		0.55	0.5	0.59	0.55	0.5
Optimal 20	5	0.8		0.56	0.5	0.6	0.56	0.51
Default	5	0.9		0.6	0.56	0.63	0.6	0.57
Optimal	5	0.9		0.61	0.56	0.64	0.61	0.57
Optimal 20	5	0.9		0.61	0.56	0.64	0.61	0.58
Default	5	0.9	1	0.61	0.56	0.64	0.61	0.58
Optimal	5	0.9	1	0.62	0.57	0.65	0.62	0.59
Optimal 20	5	0.9	1	0.62	0.57	0.65	0.62	0.59
Default	5	0.9	2	0.61	0.56	0.64	0.61	0.58
Optimal	5	0.9	2	0.62	0.57	0.65	0.62	0.58
Optimal 20	5	0.9	2	0.62	0.57	0.66	0.62	0.59
Default	15	0.8		0.52	0.45	0.49	0.52	0.46
Optimal	15	0.8		0.52	0.45	0.53	0.52	0.46
Optimal 20	15	0.8		0.53	0.46	0.49	0.53	0.47
Default	15	0.9		0.61	0.55	0.6	0.61	0.56
Optimal	15	0.9		0.6	0.54	0.61	0.6	0.54
Optimal 20	15	0.9		0.61	0.55	0.61	0.61	0.56
Default	15	0.9	5	0.62	0.56	0.63	0.62	0.58
Optimal	15	0.9	5	0.62	0.57	0.64	0.62	0.58
Optimal 20	15	0.9	5	0.62	0.56	0.62	0.62	0.58
Default	15	0.9	7	0.63	0.57	0.63	0.63	0.59
Optimal	15	0.9	7	0.63	0.57	0.63	0.63	0.59
Optimal 20	15	0.9	7	0.65	0.59	0.63	0.65	0.61
Default	20	0.8		0.51	0.43	0.43	0.51	0.42
Optimal	20	0.8		0.51	0.43	0.56	0.51	0.43
Optimal 20	20	0.8		0.51	0.43	0.49	0.51	0.43
Default	20	0.9		0.6	0.54	0.63	0.6	0.55
Optimal	20	0.9		0.6	0.53	0.61	0.6	0.54
Optimal 20	20	0.9		0.59	0.52	0.6	0.59	0.52
Default	20	0.9	5	0.61	0.55	0.63	0.61	0.57
Optimal	20	0.9	5	0.61	0.55	0.64	0.61	0.56
Optimal 20	20	0.9	5	0.62	0.55	0.63	0.62	0.56
Default	20	0.9	10	0.62	0.56	0.65	0.62	0.57
Optimal	20	0.9	10	0.61	0.54	0.64	0.61	0.55
Optimal 20	20	0.9	10	0.62	0.56	0.63	0.62	0.57

Tableau 6.13 : Tableau représentant plusieurs indices de performance selon la taille des fenêtres temporelles (time window) et le chevauchement (overlap) des jeux de données. Algorithme random forest.

6.2.4 ADABOOST

L'algorithme AdaBoost a été testé selon les deux configurations "Optimal" et "Optimal 1.5" évoquées dans la section précédente. Les résultats sont synthétisés dans le Tableau 6.14. Cette méthode semble bénéficier d'un léger gain de performance sur les variations autour de la fenêtre temporelle de 5 secondes, mais une baisse de celle-ci sur toutes les autres variations. Or, étant donné que cet algorithme performe le mieux sur la fenêtre large de 20 secondes, il semblerait qu'il soit plus efficace avec le pré-traitement original.

Configuration	Fen. 1	Chev.	Fen. 2.	Accuracy	Kappa	Precision	Recall	F1
Optimal	5	0.8		0.47	0.42	0.47	0.47	0.47
Optimal 1.5	5	0.8		0.47	0.42	0.47	0.47	0.47
Optimal	5	0.9		0.59	0.55	0.59	0.59	0.59
Optimal 1.5	5	0.9		0.59	0.55	0.6	0.59	0.59
Optimal	5	0.9	1	0.59	0.55	0.59	0.59	0.59
Optimal 1.5	5	0.9	1	0.59	0.55	0.59	0.59	0.59
Optimal	5	0.9	2	0.6	0.55	0.59	0.6	0.59
Optimal 1.5	5	0.9	2	0.6	0.56	0.6	0.6	0.6
Optimal	15	0.8		0.57	0.52	0.58	0.57	0.57
Optimal 1.5	15	0.8		0.58	0.53	0.58	0.58	0.58
Optimal	15	0.9		0.68	0.64	0.68	0.68	0.68
Optimal 1.5	15	0.9		0.69	0.65	0.68	0.69	0.68
Optimal	15	0.9	5	0.67	0.63	0.67	0.67	0.67
Optimal 1.5	15	0.9	5	0.67	0.63	0.67	0.67	0.67
Optimal	15	0.9	7	0.68	0.64	0.69	0.68	0.68
Optimal 1.5	15	0.9	7	0.68	0.64	0.68	0.68	0.68
Optimal	20	0.8		0.57	0.51	0.58	0.57	0.57
Optimal 1.5	20	0.8		0.55	0.5	0.56	0.55	0.55
Optimal	20	0.9		0.71	0.67	0.71	0.71	0.71
Optimal 1.5	20	0.9		0.71	0.67	0.71	0.71	0.71
Optimal	20	0.9	5	0.68	0.64	0.68	0.68	0.68
Optimal 1.5	20	0.9	5	0.68	0.64	0.68	0.68	0.68
Optimal	20	0.9	10	0.68	0.64	0.68	0.68	0.68
Optimal 1.5	20	0.9	10	0.68	0.64	0.68	0.68	0.68

Tableau 6.14 : Tableau représentant plusieurs indices de performance selon la taille des fenêtres temporelles (time window) et le chevauchement (overlap) des jeux de données. Algorithme AdaBoost.

6.2.5 K PLUS PROCHE VOISINS

Enfin, la méthode des K plus proches voisins a été testée sur les configurations "Optimal" et "Optimal 1" définies plus haut. Les résultats sont présentés dans le Tableau 6.15. Ici, les performances des nouvelles variations sont toutes supérieures à leurs équivalents originels. Parmi les nouvelles variations, ce sont celles avec les plus petites secondes fenêtres temporelles qui sont les plus efficaces. Ainsi, ce sont les versions

”5_0.9_1“, ”15_0.9_5“ et ”20_0.9_5“ qui sont les plus performantes. Au total, on a un gain de 2% sur l’indice d’*accuracy* entre la meilleure combinaison précédente (”Optimal 1“ - ”20_0.9“) et la nouvelle meilleure combinaison (”Optimal 1“ - ’20_0.9_5“).

Configuration	Fen. 1	Chev.	Fen. 2.	Accuracy	Kappa	Precision	Recall	F1
Optimal	5	0.8		0.52	0.46	0.51	0.52	0.5
Optimal 1	5	0.8		0.49	0.44	0.49	0.49	0.49
Optimal	5	0.9		0.61	0.57	0.61	0.61	0.61
Optimal 1	5	0.9		0.62	0.58	0.62	0.62	0.62
Optimal	5	0.9	1	0.63	0.59	0.63	0.63	0.62
Optimal 1	5	0.9	1	0.63	0.59	0.63	0.63	0.63
Optimal	5	0.9	2	0.62	0.58	0.62	0.62	0.62
Optimal 1	5	0.9	2	0.63	0.59	0.63	0.63	0.63
Optimal	15	0.8		0.58	0.53	0.57	0.58	0.57
Optimal 1	15	0.8		0.54	0.49	0.54	0.54	0.54
Optimal	15	0.9		0.66	0.62	0.66	0.66	0.66
Optimal 1	15	0.9		0.68	0.64	0.68	0.68	0.68
Optimal	15	0.9	5	0.71	0.67	0.71	0.71	0.7
Optimal 1	15	0.9	5	0.73	0.7	0.73	0.73	0.73
Optimal	15	0.9	7	0.69	0.65	0.69	0.69	0.69
Optimal 1	15	0.9	7	0.71	0.67	0.72	0.71	0.71
Optimal	20	0.8		0.57	0.51	0.57	0.57	0.56
Optimal 1	20	0.8		0.59	0.53	0.59	0.59	0.58
Optimal	20	0.9		0.7	0.66	0.7	0.7	0.69
Optimal 1	20	0.9		0.71	0.67	0.71	0.71	0.71
Optimal	20	0.9	5	0.72	0.69	0.72	0.72	0.72
Optimal 1	20	0.9	5	0.73	0.7	0.73	0.73	0.73
Optimal	20	0.9	10	0.71	0.67	0.71	0.71	0.71
Optimal 1	20	0.9	10	0.72	0.68	0.72	0.72	0.72

Tableau 6.15 : Tableau représentant plusieurs indices de performance selon la taille des fenêtres temporelles (time window) et le chevauchement (overlap) des jeux de données. Algorithme KNN.

6.2.6 ANALYSE COMPARATIVE

Le Tableau 6.16 compare les algorithmes sur toutes les nouvelles variations du jeu de données. On observe assez peu de changement par rapport au tableau comparatif original, hormis la méthode de la forêt aléatoire qui surpasse CART et AdaBoost sur les variantes autour de la fenêtre large de 5 secondes. Ici, KNN est plus performant que les autres méthodes sur toutes les nouvelles variantes. En revanche, certains algorithmes performant moins bien avec celles-ci qu'avec les variantes originales. Ainsi, le Tableau 6.17 condense les meilleures performances relevées pour chaque algorithme durant les séries de tests de ce chapitre, avec la configuration utilisée et la variante du jeu de données sur lequel le modèle a été entraîné. Tous les indices concordent pour déclarer que l'algorithme KNN est la méthode testée la plus efficace pour répondre à notre problématique, avec une *accuracy* de 73,3% et *Cohen kappa* de 0,698.

Méthode	Configuration	Fen. 1	Fen. 2.	Accuracy	Kappa	Precision	Recall	F1
CART	Optimal	5	1	0.6	0.56	0.6	0.6	0.59
Random forest	Optimal 20	5	1	0.62	0.57	0.65	0.62	0.59
AdaBoost	Optimal	5	1	0.59	0.55	0.59	0.59	0.59
KNN	Optimal 1	5	1	0.63	0.59	0.63	0.63	0.63
CART	Optimal	5	2	0.61	0.57	0.61	0.61	0.6
Random forest	Optimal 20	5	2	0.62	0.57	0.66	0.62	0.59
AdaBoost	Optimal	5	2	0.6	0.55	0.59	0.6	0.59
KNN	Optimal 1	5	2	0.63	0.59	0.63	0.63	0.63
CART	Optimal	15	5	0.68	0.64	0.68	0.68	0.68
Random forest	Optimal 20	15	5	0.62	0.56	0.62	0.62	0.58
AdaBoost	Optimal	15	5	0.67	0.63	0.67	0.67	0.67
KNN	Optimal 1	15	5	0.73	0.7	0.73	0.73	0.73
CART	Optimal	15	7	0.69	0.66	0.69	0.69	0.69
Random forest	Optimal 20	15	7	0.65	0.59	0.63	0.65	0.61
AdaBoost	Optimal	15	7	0.68	0.64	0.69	0.68	0.68
KNN	Optimal 1	15	7	0.71	0.67	0.72	0.71	0.71
CART	Optimal	20	5	0.7	0.66	0.69	0.7	0.69
Random forest	Optimal 20	20	5	0.62	0.55	0.63	0.62	0.56
AdaBoost	Optimal	20	5	0.68	0.64	0.68	0.68	0.68
KNN	Optimal 1	20	5	0.73	0.7	0.73	0.73	0.73
CART	Optimal	20	10	0.7	0.66	0.69	0.7	0.7
Random forest	Optimal 20	20	10	0.62	0.56	0.63	0.62	0.57
AdaBoost	Optimal	20	10	0.68	0.64	0.68	0.68	0.68
KNN	Optimal 1	20	10	0.72	0.68	0.72	0.72	0.72

Tableau 6.16 : Tableau comparatif des quatres méthodes d'apprentissage machine testées sur les 6 nouvelles variations de pré-traitement du jeu de données.

Méthode	Configuration	Fen. 1	Fen. 2.	Accuracy	Kappa	Precision	Recall	F1
Random forest	Optimal 20	15	7	0.65	0.59	0.63	0.65	0.61
CART	Optimal	20		0.71	0.67	0.7	0.71	0.7
AdaBoost	Optimal	20		0.71	0.67	0.71	0.71	0.71
KNN	Optimal 1	20	5	0.73	0.7	0.73	0.73	0.73

Tableau 6.17 : Tableau comparatif final regroupant les meilleures performances de chaque algorithme testé dans ce chapitre.

CONCLUSION

Dans un contexte où l'aide aux personnes âgées est un enjeu sociétal, social, économique et éthique majeur, les nouvelles technologies se présentent comme une partie de la solution. Les habitats intelligents et l'équipement qui les composent sont des outils précieux, qui, combinés avec des techniques d'intelligence artificielle, peuvent permettre de répondre à ce défi. La jonction entre ces enjeux et les radars UWB n'est pas la source de nombreux travaux scientifiques. Ainsi, l'objectif de ce mémoire était de répondre à la problématique suivante. Comment effectuer de la reconnaissance d'activités dans un habitat intelligent à l'aide de radars UWB ? Nous avons répondu à cette question de trois manières distinctes : par une contribution logicielle, par la création de deux jeux de données, et par une contribution sur l'apprentissage machine.

Dans un premier temps, une preuve de concept sous la forme d'un ensemble de briques logicielles a été réalisée. La chaîne que forment ces briques permet, de bout en bout, de récolter les données, d'effectuer des calculs pour les contextualiser, de les formater, de les pré-traiter, de les envoyer à des modèles d'intelligence artificielle, et de les retransmettre sur une interface utilisateur distante. Dans un deuxième temps, deux jeux de données ont été créés et utilisés. Ils ont donné lieu à plusieurs contributions scientifiques (Bouchard *et al.*, 2020), auxquelles s'ajoutent ce document et les futures contributions à venir. Ces jeux de données sont un premier pas dans la création d'outils et de données pour stimuler la recherche dans ce domaine. Enfin, dans un troisième temps, des tests avec différentes techniques d'apprentissage machine ont permis d'évaluer les performances pouvant être obtenues grâce aux jeux de données récoltés. Avec des

scores de précision atteignant jusqu'à 73% pour un problème multi-classe, ils ont ainsi démontrés la faisabilité d'utiliser des radars UWB pour faire de la reconnaissance d'activités.

Ce mémoire laisse place à diverses opportunités de contributions futures. De nombreuses améliorations de la preuve de concept peuvent être envisagées, à la fois sur la brique d'acquisition des données, sur la brique centralisant les divers flux de données, et sur la brique d'affichage de l'interface utilisateur. Le code source gagnerait en clarté à être unifié et retravaillé. Des nouvelles fonctionnalités pour faciliter les phases de test peuvent être ajoutées, et un travail sur l'ergonomie et l'optimisation algorithmique peut également être envisagé.

Le jeu de données des activités pourrait bénéficier d'une nouvelle extension afin de rendre la distribution des classes uniforme pendant la phase d'apprentissage machine. Un ajout de nouvelles instances pour les activités les moins longues peut ainsi être réalisé. La création d'un jeu de données dans des conditions encore plus proches de la réalité serait également intéressante, par exemple, en embarquant les capteurs dans de réelles habitations occupées quotidiennement par des volontaires.

Enfin, les tests d'apprentissage machine peuvent être étendus pour parcourir l'espace de paramétrage plus exhaustivement, avec des machines plus puissantes. D'autres méthodes d'apprentissages pourraient également s'avérer efficaces, par exemple les techniques d'apprentissage profond évoquées dans le chapitre 2. Enfin, l'extension des tests sur le jeu de données des chutes et sur une combinaison des deux jeux de données pourrait ajouter une nouvelle perspective aux travaux de ce mémoire.

BIBLIOGRAPHIE

Alam, M. R., Reaz, M. B. I. et Ali, M. A. M. (2012). A review of smart homes—past, present, and future. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(6), 1190–1203.

Alberdi, A., Weakley, A., Schmitter-Edgecombe, M., Cook, D. J., Aztiria, A., Basarab, A. et Barrenechea, M. (2018). Smart home-based prediction of multidomain symptoms related to alzheimer’s disease. *IEEE Journal of Biomedical and Health Informatics*, 22(6), 1720–1731.

Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3), 175–185.

Baird, Z., Gunasekara, I., Bolic, M. et Rajan, S. (2017). Principal component analysis-based occupancy detection with ultra wideband radar. *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*, 1573–1576.

Bhattacharya, S. et Lane, N. D. (2016). From smart to deep : Robust activity recognition on smartwatches using deep learning. Dans *2016 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, 1–6.

Bouchard, K., Maitre, J., Bertuglia, C. et Gaboury, S. (2020). Activity recognition in smart homes using uwb radars. *Procedia Computer Science*, 170, 10–17.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1), 5–32.

Breiman, L., Friedman, J., Olshen, R. et Stone, C. J. (1984). *Classification and regression trees*. Chapman and Hall/CRC.

Brown, R., Ghavami, N., Siddiqui, H.-U.-R., Adjrad, M., Ghavami, M. et Dudley, S. (2017). Occupancy based household energy disaggregation using ultra wideband radar and electrical signature profiles. *Energy and Buildings*, 141, 134 – 141.

Chan, M., Campo, E., Estève, D. et Fourniols, J.-Y. (2009). Smart homes — current features and future perspectives. *Maturitas*, 64(2), 90 – 97.

Chappell, N. L., Dlitt, B. H., Hollander, M. J., Miller, J. A. et McWilliam, C. (2004). Comparative costs of home care and residential care. *The Gerontologist*, 44(3),

389–400.

Chapron, K., Bouchard, K., Duchesne, E. et Gaboury, S. (2017). Transportable and scalable system for activities and exercises recognition in real-time. Dans *2017 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computed, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCOM/IOP/SCI)*, 1–7.

Cheng, J., Sundholm, M., Zhou, B., Hirsch, M. et Lukowicz, P. (2016). Smart-surface : Large scale textile pressure sensors arrays for activity recognition. *Pervasive and Mobile Computing*, 30, 97 – 112.

Choi, J. W., Nam, S. et Cho, S. H. (2017). Multi-human detection algorithm based on an impulse radio ultra-wideband radar system. *IEEE Access*, PP, 1–1.

Cousyn, C., Bouchard, K., Gaboury, S. et Bouchard, B. (2020). Towards using scientific publications to automatically extract information on rare diseases. *Mobile Networks and Applications*, 25, 953–960.

Cover, T. et Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1), 21–27.

Department of Economic and Social Affairs, United Nations (2017). World population prospects, the 2017 revision.

Department of Economic and Social Affairs, United Nations (2019). World population prospects, the 2019 revision.

Diraco, G., Leone, A. et Siciliano, P. (2018). A fall detector based on ultra-wideband radar sensing. *Sensors*, 373–382.

Doppler, C. (1842). *Über das farbige Licht der Doppelsterne und einiger anderer Gestirne des Himmels*. Wiley Online Library.

Dudani, S. A. (1976). The distance-weighted k-nearest-neighbor rule. *IEEE Transactions on Systems, Man, and Cybernetics*, (4), 325–327.

Fortin-Simard, D., Bilodeau, J., Bouchard, K., Gaboury, S., Bouchard, B. et Bouzouane, A. (2015). Exploiting passive rfid technology for activity recognition in smart homes. *IEEE Intelligent Systems*, 30(4), 7–15.

Freund, Y. et Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1), 119 – 139.

Garcia-Ceja, E., Galván-Tejada, C. E. et Brena, R. (2018). Multi-view stacking for activity recognition with sound and accelerometer data. *Information Fusion*, 40, 45 – 56.

Goodfellow, I., Bengio, Y. et Courville, A. (2016). *Deep Learning*. MIT Press.

Gravelle, M. (2019). *La création d'un milieu de vie en CHSLD par la participation sociale des résidents*. (Mémoire de maîtrise). University of Ottawa.

Hammerla, N. Y., Halloran, S. et Ploetz, T. (2016). Deep, convolutional, and recurrent models for human activity recognition using wearables.

He, K., Zhang, X., Ren, S. et Sun, J. (2016). Deep residual learning for image recognition. Dans *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hevesi, P., Wille, S., Pirkl, G., Wehn, N. et Lukowicz, P. (2014). Monitoring household activities and user location with a cheap, unobtrusive thermal sensor array. Dans *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing, UbiComp '14*, p. 141–145., New York, NY, USA. Association for Computing Machinery.

Ho, T. K. (1995). Random decision forests. Dans *Proceedings of 3rd International Conference on Document Analysis and Recognition*, volume 1, 278–282.

Hochreiter, S. et Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735–1780.

Hosmer Jr, D. W., Lemeshow, S. et Sturdivant, R. X. (2013). *Applied logistic regression*, volume 398. John Wiley & Sons.

Institut de la Statistique du Québec (2019). Perspectives démographiques du Québec et des régions, 2016-2066, édition 2019.

Karlik, B. et Olgac, A. V. (2011). Performance analysis of various activation functions in generalized mlp architectures of neural networks. *International Journal of Artificial Intelligence and Expert Systems*, 1(4), 111–122.

Kleinbaum, D. G., Dietz, K., Gail, M., Klein, M. et Klein, M. (2002). *Logistic regression*. Springer.

Krizhevsky, A., Sutskever, I. et Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Dans *Advances in neural information processing systems*, 1097–1105.

LeCun, Y., Bengio, Y. *et al.* (1995). Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10), 1995.

Leon, J. et Moyer, D. (1999). Potential cost savings in residential care for Alzheimer's disease patients for Alzheimer's disease patients. *The Gerontologist*, 39(4), 440–449.

Li, X., Zhang, Y., Marsic, I., Sarcevic, A. et Burd, R. S. (2016). Deep learning for rfid-based activity recognition. Dans *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM, SenSys '16*, p. 164–175., New York, NY, USA. Association for Computing Machinery.

Liang, X., Zhang, H., Fang, G., Ye, S. et Gulliver, T. A. (2017). An improved algorithm for through-wall target detection using ultra-wideband impulse radar. *IEEE Access*, 5, 22101–22118.

Maitre, J., Rendu, C., Bouchard, K., Bouchard, B. et Gaboury, S. (2019). Basic daily activity recognition with a data glove. *Procedia Computer Science*, 151, 108 – 115.

Maître, J., Bouchard, K., Gaboury, S. et Bertuglia, C. (2019a). Détection des chutes à partir d'un capteur uwb, demande d'approbation éthique pour un projet de recherche impliquant des êtres humains.

Maître, J., Bouchard, K., Gaboury, S. et Bertuglia, C. (2019b). Reconnaissance d'activités dans un habitat intelligent, demande d'approbation éthique pour un projet de recherche impliquant des êtres humains.

Muja, M. et Lowe, D. G. (2009). Fast approximate nearest neighbors with automatic algorithm configuration. Dans *In VISAPP International Conference on Computer Vision Theory and Applications*, 331–340.

Muqaibel, A., Safaai-Jazi, A., Bayram, A., Attiya, A. et Riad, S. (2005). Ultrawideband through-the-wall propagation. *IEE Proceedings-Microwaves, Antennas and Propagation*, 152(6), 581–588.

Murphy, W. et Hereman, W. (1995). Determination of a position in three dimensions using trilateration and approximate distances. *Department of Mathematical and Computer Sciences, Colorado School of Mines, Golden, Colorado, MCS-95*, 7, 19.

Nair, V. et Hinton, G. E. (2010a). Rectified linear units improve restricted boltzmann machines. Dans *Proceedings of the 27th International Conference on International Conference on Machine Learning, ICML'10*, p. 807–814., Madison, WI, USA. Omnipress.

Nair, V. et Hinton, G. E. (2010b). Rectified linear units improve restricted boltzmann machines. Dans *ICML*, 807–814.

Naishadham, K., Piou, J. E., Ren, L. et Fathy, A. E. (2016). Estimation of cardiopulmonary parameters from ultra wideband radar measurements using the state space method. *IEEE Transactions on Biomedical Circuits and Systems*, 10(6), 1037–1046.

Nilsson, N. J. (1965). *Learning machines*. New York.

Organisation Mondiale de la Santé (2006). Neurological disorders : public health challenges.

Orpwood, R., Gibbs, C., Adlam, T., Faulkner, R. et Meegahawatte, D. (2005). The design of smart homes for people with dementia—user-interface aspects. *Universal Access in the Information Society*, 4(2), 156–164.

Pallesen, S., Grønli, J., Myhre, K., Moen, F., Bjorvatn, B., Hanssen, I. et Heglum, H. S. (2018). A pilot study of impulse radio ultra wideband radar technology as a new tool for sleep assessment. *J Clin Sleep Med.*, 14(7), 1249–1254.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A.,

Cournapeau, D., Brucher, M., Perrot, M. et Duchesnay, E. (2011). Scikit-learn : Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.

Planinc, R. et Kampel, M. (2013). Introducing the use of depth data for fall detection. *Personal and Ubiquitous Computing*, 17(6), 1063–1072.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1(4), 81–106.

Quinlan, J. R. (1993). *C4.5 : Programs for Machine Learning*. San Francisco, CA, USA : Morgan Kaufmann Publishers Inc.

Riendeau, Y. (2006). *La qualité des soins offerts aux personnes âgées en CHSLD : l'opinion des préposé(e)s aux bénéficiaires*. (Mémoire de maîtrise). Université du Québec à Montréal.

Robinovitch, S. N., Feldman, F., Yang, Y., Schonnop, R., Leung, P. M., Sarraf, T., Sims-Gould, J. et Loughin, M. (2013). Video capture of the circumstances of falls in elderly people residing in long-term care : an observational study. *Lancet*, 381(9860), 47–54.

Rosenblatt, F. (1958). The perceptron : A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65–386.

Rumelhart, D. E., Durbin, R., Golden, R. et Chauvin, Y. (1995). Backpropagation : The basic theory. *Backpropagation : Theory, architectures and applications*, 1–34.

Russell, S. et Norvig, P. (2020). *Artificial intelligence : a modern approach* (4 éd.). Prentice Hall.

Sadrezami, H., Bolic, M. et Rajan, S. (2018). On the use of ultra wideband radar and stacked lstm-rnn for at home fall detection. Dans *2018 IEEE Life Sciences Conference (LSC)*, 255–258.

Sadrezami, H., Bolic, M. et Rajan, S. (2019). Fall detection using standoff radar-based sensing and deep convolutional neural network. *IEEE Transactions on Circuits and Systems II : Express Briefs*, 1–1.

Sakamoto, T., Imasaka, R., Taki, H., Sato, T., Yoshioka, M., Inoue, K., Fukuda, T. et

Sakai, H. (2015a). Accurate heartbeat monitoring using ultra-wideband radar. *IEICE Electronics Express*, 12.

Sakamoto, T., Imasaka, R., Taki, H., Sato, T., Yoshioka, M., Inoue, K., Fukuda, T. et Sakai, H. (2016a). Feature-based correlation and topological similarity for interbeat interval estimation using ultrawideband radar. *IEEE Transactions on Biomedical Engineering*, 63(4), 747–757.

Sakamoto, T., Matsuoka, A. et Yomo, H. (2016b). Estimation of doppler velocities from sub-nyquist ultra-wideband radar measurements. *IEEE Sensors Journal*, 16(23), 8557–8565.

Sakamoto, T., Okumura, S., Imanishi, R., Taki, H., Sato, T., Yoshioka, M., Inoue, K., Fukuda, T. et Sakai, H. (2015b). Remote heartbeat monitoring from human soles using 60-ghz ultra-wideband radar. *IEICE Electronics Express*, 12.

Sakamoto, T., Sato, T., Aubry, P. et Yarovoy, A. (2016c). Fast imaging method for security systems using ultrawideband radar. *IEEE Transactions on Aerospace and Electronic Systems*, 52(2), 658–670.

Sakamoto, T., Sato, T., Aubry, P. J. et Yarovoy, A. G. (2015). Ultra-wideband radar imaging using a hybrid of kirchhoff migration and stolt f-k migration with an inverse boundary scattering transform. *IEEE Transactions on Antennas and Propagation*, 63(8), 3502–3512.

Skolnik, M. (2001). *Introduction to Radar Systems* (illustrated 3 éd.). Electrical engineering series. McGraw-Hill.

Stager, M., Lukowicz, P. et Troster, G. (2004). Implementation and evaluation of a low-power sound-based user activity recognition system. Dans *Eighth International Symposium on Wearable Computers*, volume 1, 138–141.

Su, X., Tong, H. et Ji, P. (2014). Activity recognition with smartphone sensors. *Tsinghua Science and Technology*, 19(3), 235–249.

Sutton, R. S. et Barto, A. G. (2018). *Reinforcement learning : An introduction*. MIT press.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Van-

houcke, V. et Rabinovich, A. (2015). Going deeper with convolutions. Dans *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1–9.

Taylor, J. D. (1994). *Introduction to Ultra-Wideband Radar Systems* (1 éd.). CRC Press.

Thullier, F., Plantevin, V., Bouzouane, A., Halle, S. et Gaboury, S. (2017). A position-independent method for soil types recognition using inertial data from a wearable device. Dans *2017 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computed, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCOM/IOP/SCI)*, 1–10.

Widrow, B. et Hoff, M. E. (1960). Adaptive switching circuits. Dans *1960 IRE WESCON Convention Record, Part 4*, 96–104. IRE.

Wu, S., Tan, K., Xia, Z., Chen, J., Meng, S. et Guangyou, F. (2016). Improved human respiration detection method via ultra-wideband radar in through-wall or other similar conditions. *IET Radar, Sonar Navigation*, 10(3), 468–476.

Xia, L. et Aggarwal, J. (2013). Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera. Dans *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Xu, B., Wang, N., Chen, T. et Li, M. (2015). Empirical evaluation of rectified activations in convolutional network.

ANNEXE A

RÉSULTATS DES TESTS DE PRÉCISION CIRCULAIRE

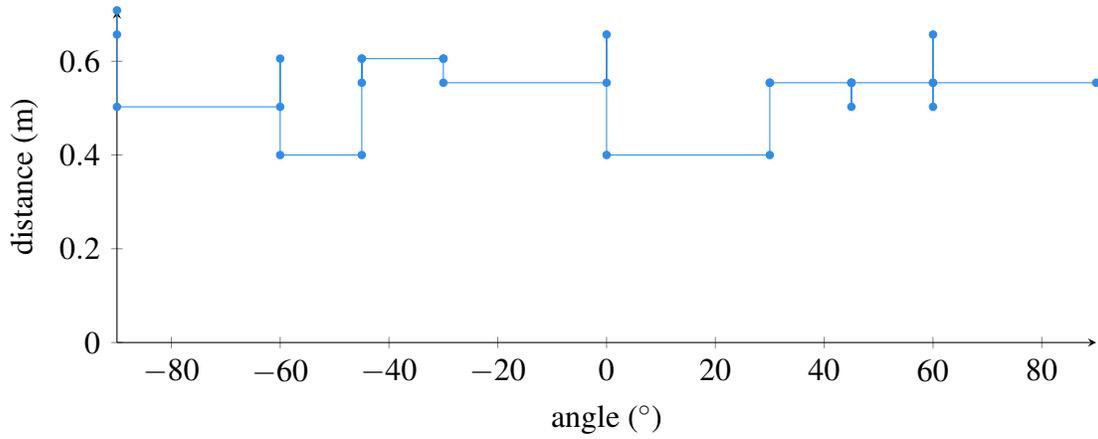


Figure A.1 : Test de précision circulaire sur un demi-cercle de 0,5m de rayon - 01

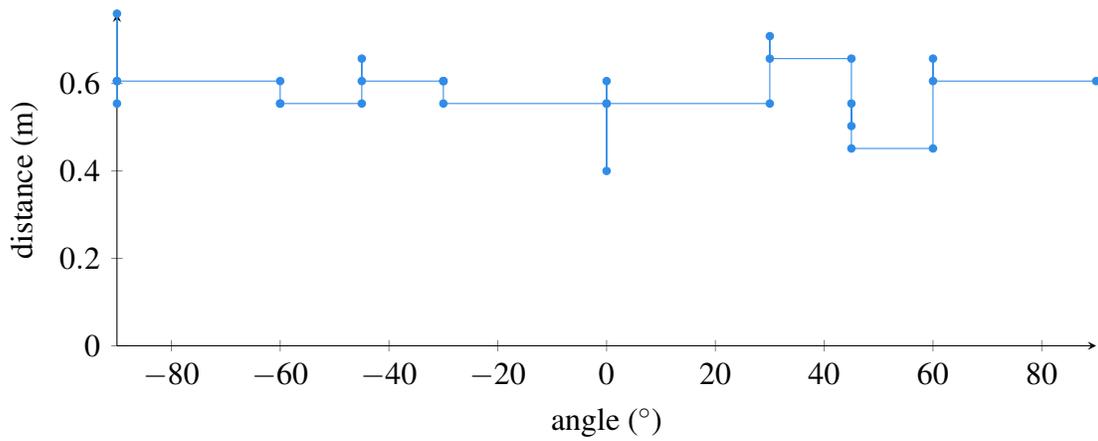


Figure A.2 : Test de précision circulaire sur un demi-cercle de 0,5m de rayon - 02

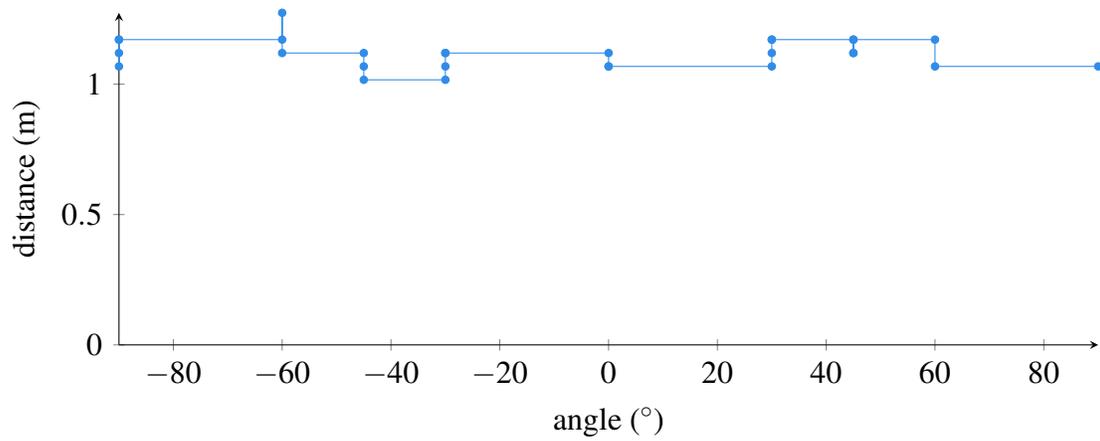


Figure A.3 : Test de précision circulaire sur un demi-cercle de 1m de rayon - 01

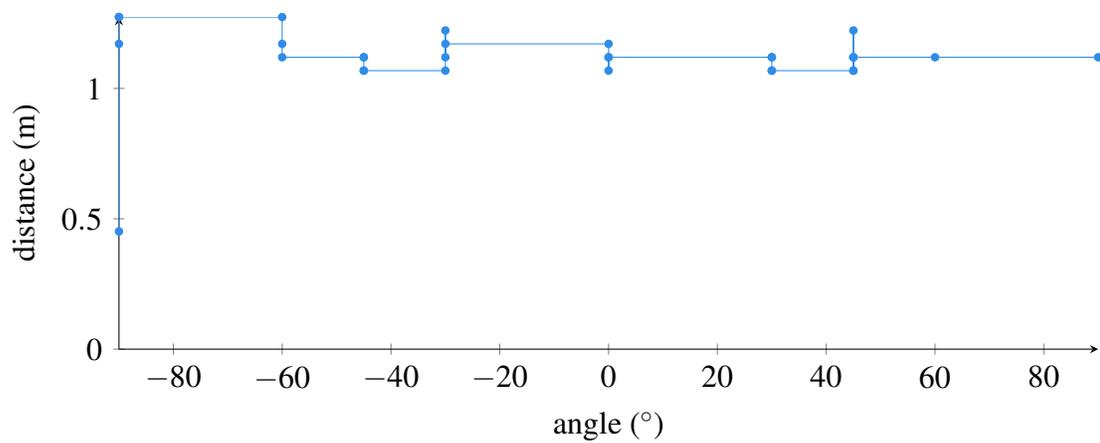


Figure A.4 : Test de précision circulaire sur un demi-cercle de 1m de rayon - 02

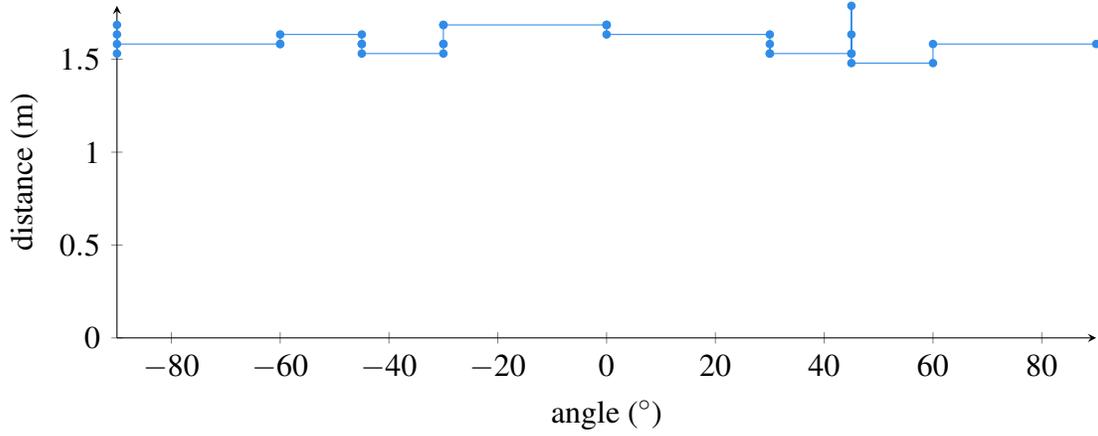


Figure A.5 : Test de précision circulaire sur un demi-cercle de 1,5m de rayon - 01

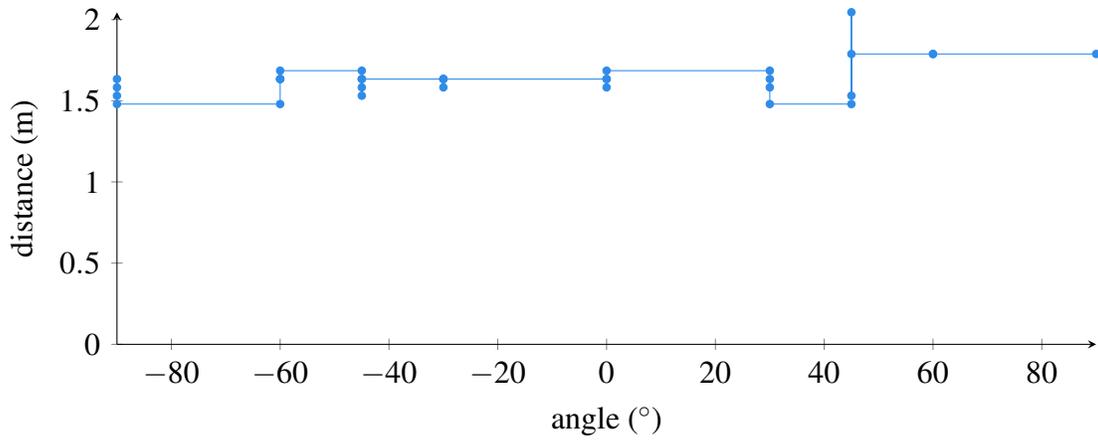


Figure A.6 : Test de précision circulaire sur un demi-cercle de 1,5m de rayon - 02

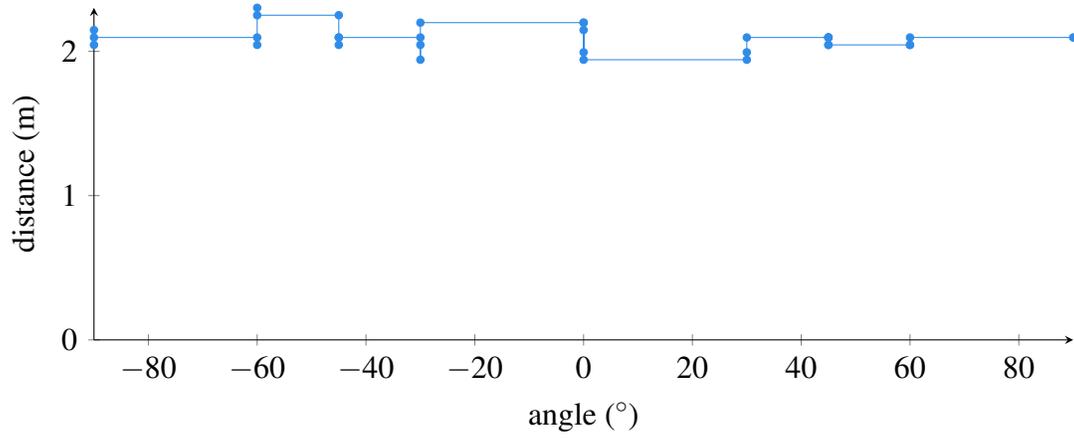


Figure A.7 : Test de précision circulaire sur un demi-cercle de 2m de rayon - 01

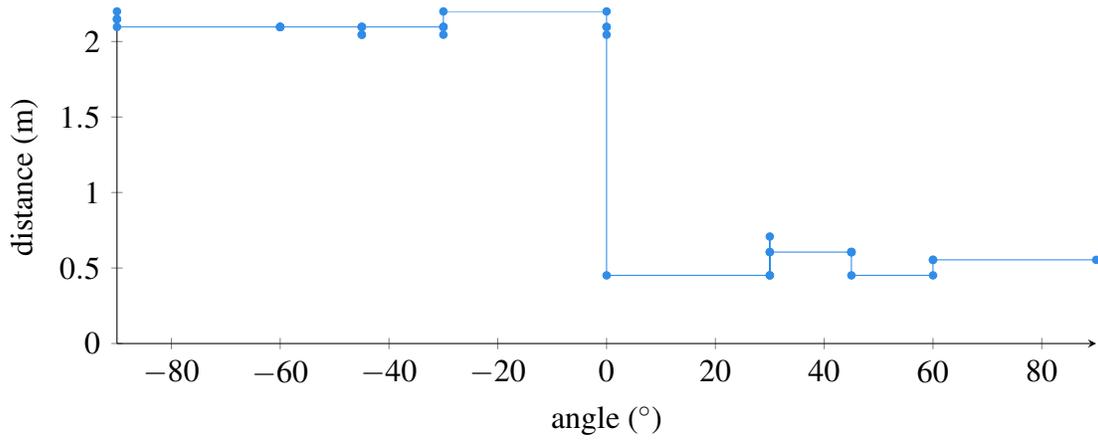


Figure A.8 : Test de précision circulaire sur un demi-cercle de 2m de rayon - 02

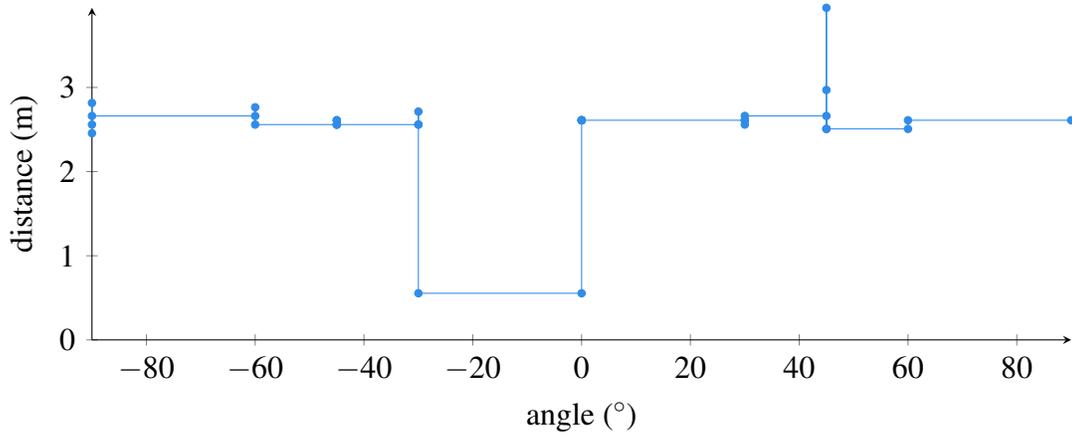


Figure A.9 : Test de précision circulaire sur un demi-cercle de 2.5m de rayon - 01

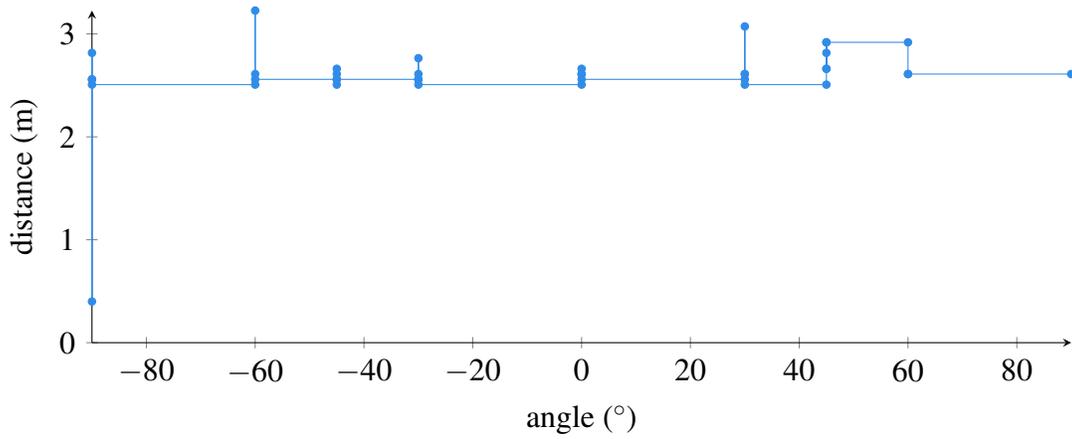


Figure A.10 : Test de précision circulaire sur un demi-cercle de 2.5m de rayon - 02

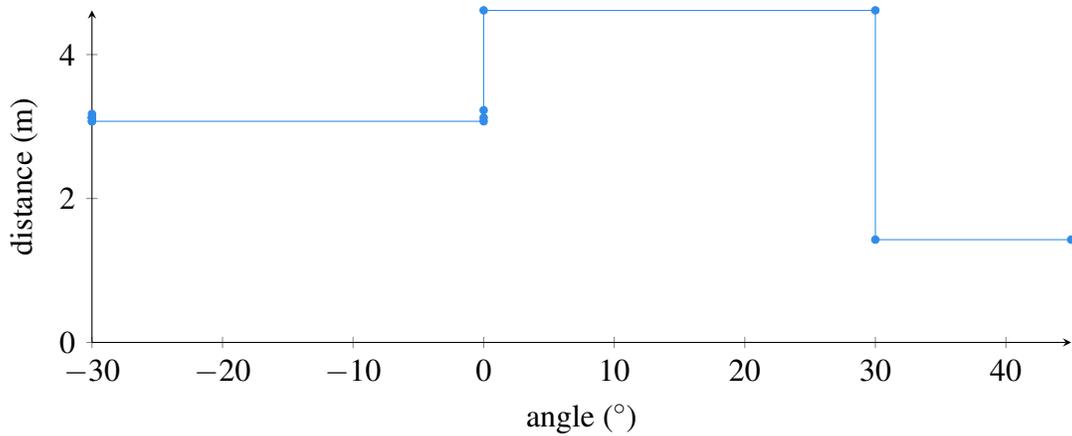


Figure A.11 : Test de précision circulaire sur un demi-cercle de 3m de rayon - 01

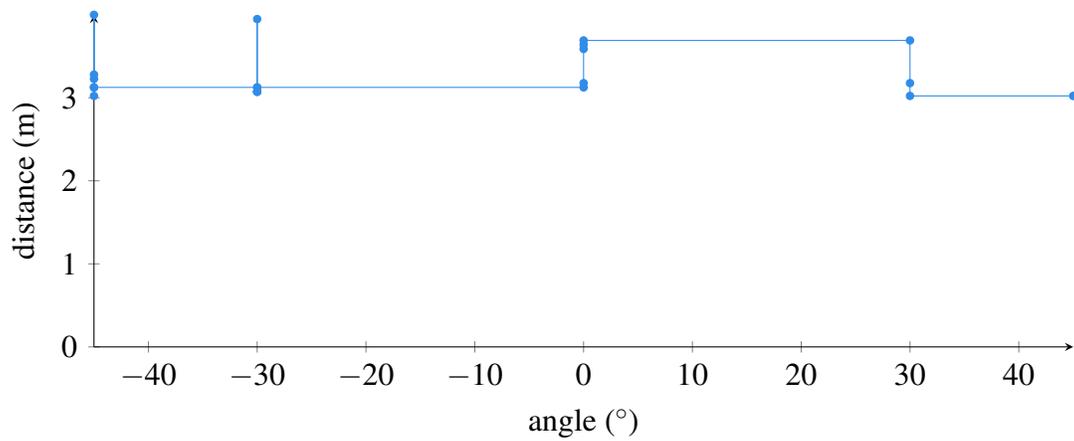


Figure A.12 : Test de précision circulaire sur un demi-cercle de 3m de rayon - 02

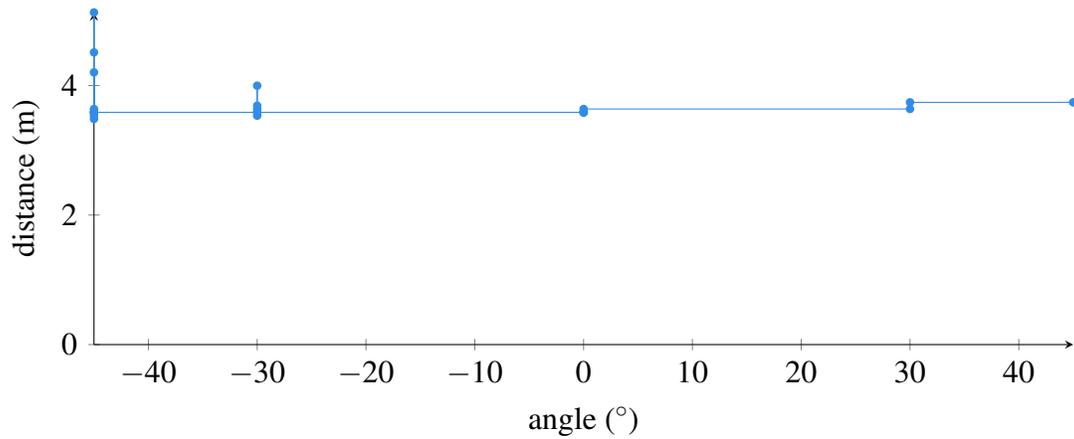


Figure A.13 : Test de précision circulaire sur un demi-cercle de 3.5m de rayon - 01

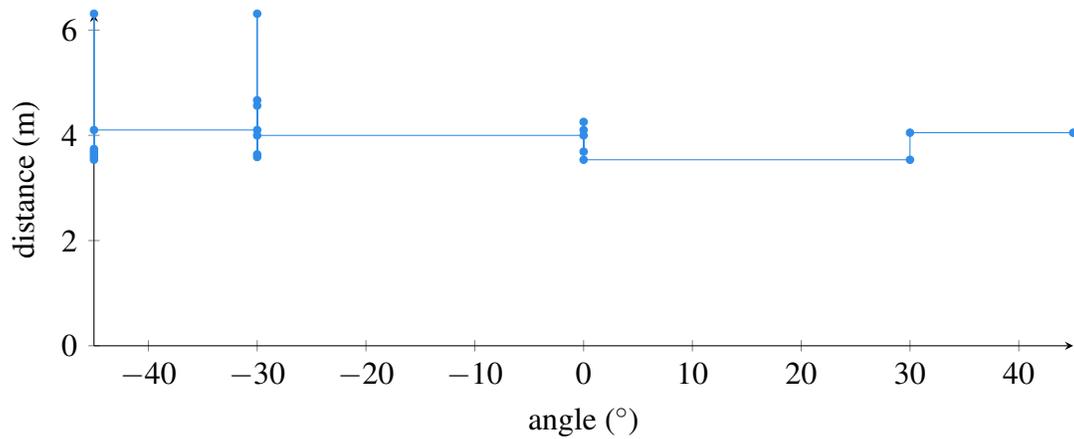


Figure A.14 : Test de précision circulaire sur un demi-cercle de 3.5m de rayon - 02

ANNEXE B

RÉSULTATS DES TESTS DE PRÉCISION LINÉAIRE

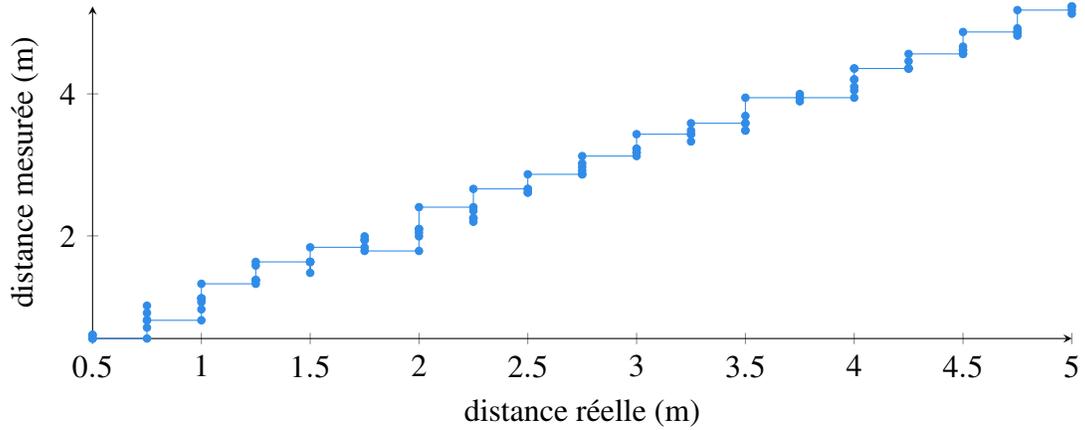


Figure B.1 : Test de précision linéaire sur l'axe central de 0,5m à 5m

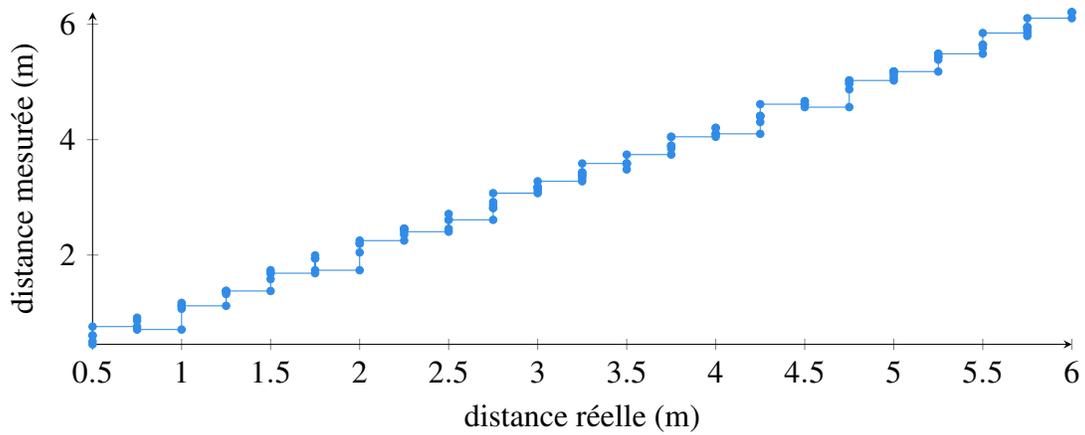


Figure B.2 : Test de précision linéaire sur l'axe central de 0,5m à 6m

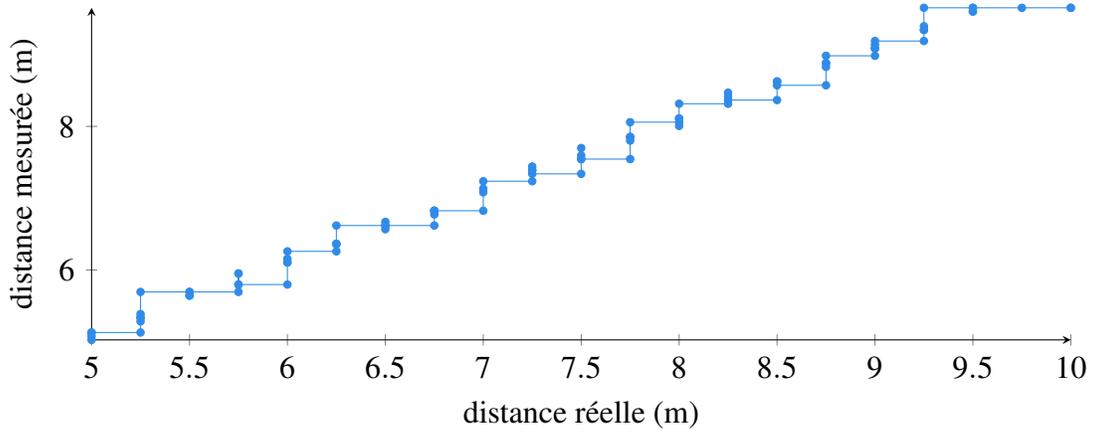


Figure B.3 : Test de précision linéaire sur l'axe central de 5m à 10m, avec les paramètres de portée par défaut étendus

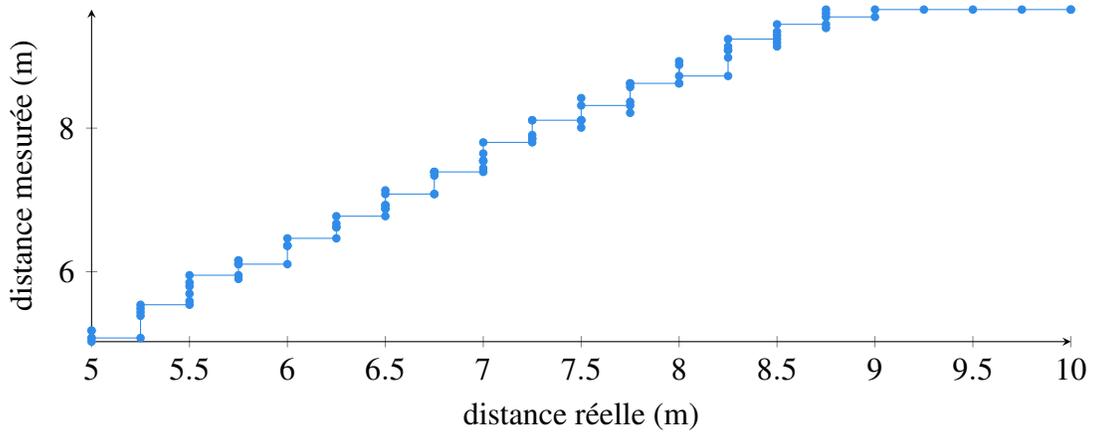


Figure B.4 : Test de précision linéaire sur l'axe central de 5m à 10m, avec les paramètres de portée par défaut maintenus

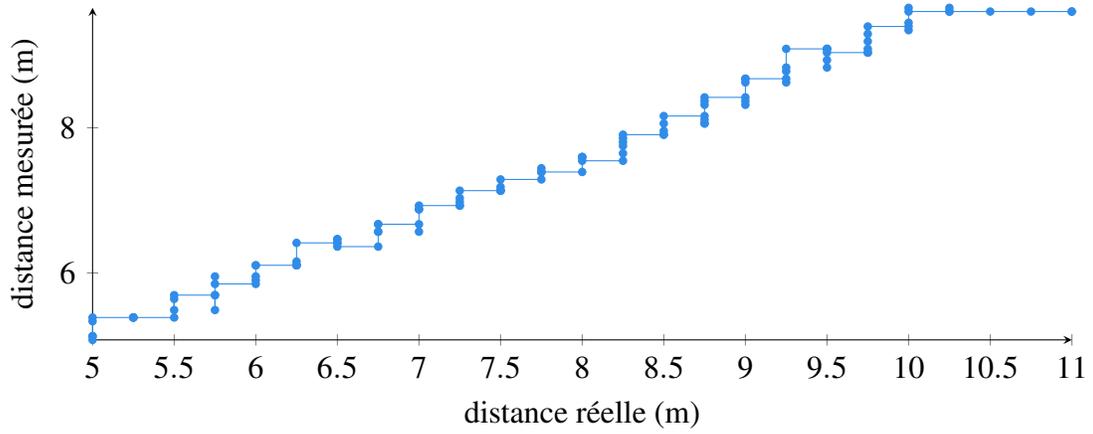


Figure B.5 : Test de précision linéaire sur l'axe central de 5m à 11m, avec les paramètres de portée par défaut maintenus

ANNEXE C

CERTIFICAT D'ÉTHIQUE POUR LE JEU DE DONNÉES DES ACTIVITÉS



CERTIFICAT D'APPROBATION ÉTHIQUE

La présente atteste que le projet de recherche décrit ci-dessous a fait l'objet d'une évaluation en matière d'éthique de la recherche avec des êtres humains et qu'il satisfait aux exigences de la politique de l'UQAC en cette matière.

Les membres jugent que ce projet rencontre les critères d'une recherche à risque minimal.

Projet # : 2019-220

Titre du projet de recherche : Reconnaissance d'activités dans un habitat intelligent

Chercheur principal :

Camille Bertudlia, Étudiant
Maîtrise en informatique
Département d'informatique et de mathématique

Directeur/codirecteur :

En provenance de l'UQAC:
Kevin Bouchard, Professeur
Département d'informatique et de mathématique
Sébastien Gaboury, Professeur
Département d'informatique et de mathématique
Julien Maitre, Chargé de cours
Département d'informatique et de mathématique

Financement : Programme de soutien à la valorisation et au transfert (PSVT) du ministère de l'Économie et de l'Innovation (MESI) (financement de Kevin Bouchard)

Date d'approbation du projet : 21 mai 2019

Date d'entrée en vigueur du certificat : 21 mai 2019

Date d'échéance du certificat : 21 mai 2020

A handwritten signature in blue ink that reads 'Tommy Chevette'.

Tommy Chevette

ANNEXE E

CERTIFICAT D'ÉTHIQUE POUR LE JEU DE DONNÉES DES CHUTES



CERTIFICAT D'APPROBATION ÉTHIQUE

La présente atteste que le projet de recherche décrit ci-dessous a fait l'objet d'une évaluation en matière d'éthique de la recherche avec des êtres humains et qu'il satisfait aux exigences de la politique de l'UQAC en cette matière.

Les membres jugent que ce projet rencontre les critères d'une recherche à risque minimal.

Projet # : 2019-235

Titre du projet de recherche : Détection des chutes à partir d'un capteur UWB

Chercheur principal :

Camille Bertuglia, Étudiant (Maîtrise en informatique)
Département d'informatique et de mathématique

Directeur/codirecteur :

Kévin Bouchard, Professeur
Département d'informatique et de mathématique

Sébastien Gaboury, Professeur
Département d'informatique et de mathématique

Julien Maitre, Professeur
Département d'informatique et de mathématique

Financement : PSVT Maturation (Kévin Bouchard)

Date d'approbation du projet : 09 juillet 2019

Date d'entrée en vigueur du certificat : 09 juillet 2019

Date d'échéance du certificat : 09 juillet 2020

A handwritten signature in blue ink that reads 'Tommy Chevette'. The signature is written in a cursive style with a horizontal line above the name.

Tommy Chevette