

**REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA
RECHERCHE SCIENTIFIQUE
UNIVERSITE BLIDA 1**

**FACULTE DES SCIENCES
DEPARTEMENT D'INFORMATIQUE**



Présenté en vue de l'obtention du Diplôme de
MASTER EN INFORMATIQUE

Option : Systèmes Informatiques et Réseaux

THEME

**PROPOSITION ET DEVELOPPEMENT
D'UN OBJET INTELLIGENT POUR LA PREVENTION
CONTRE LES ACCIDENTS DE PLONGEE EN APNEE**

Réalisé par :

BOUCHENAFI Rayane

SAKHRI Farid

Encadré par :

Dr DOUGA Yassine

Juin 2023

DEDICACES

A mes chers parents qui m'ont donné la vie !

A ma famille.

A mes amis.

*Avec toute ma reconnaissance
pour votre amour et votre
patience.*

BOUCHENAF A Rayane

DEDICACES

A mes chers parents,

Je tiens à exprimer ma profonde gratitude pour votre soutien indéfectible tout au long de mon parcours universitaire. Votre amour, votre encouragement et votre dévouement ont été des piliers essentiels de ma réussite.

Que ce message de reconnaissance exprime mon immense gratitude envers mes parents, mes frères. Vous êtes les piliers de ma vie et vos encouragements ont été ma source d'inspiration constante.

À mes chers frères, Mahdi et Walid, je vous suis infiniment reconnaissant d'avoir été présents à mes côtés pendant cette aventure académique. Votre soutien inconditionnel, vos encouragements constants et vos précieux conseils ont fait toute la différence. Vous avez été mes plus grands supporters et mes confidents les plus précieux.

Et à mes amis,

Avec tout mon amour et ma reconnaissance,

SAKHRI Farid

REMERCIEMENTS

Nous commençons d'abord par remercier Dieu le tout puissant pour la réalisation de ce travail.

Nous adressons nos remerciements les plus sincères au Docteur DOUGA Yassine, qui nous a fait le grand honneur de diriger ce mémoire. Qu'il nous soit permis de lui témoigner toute notre gratitude pour l'aide, les conseils et le soutien qu'il nous a témoigné ainsi que pour la disponibilité qu'il nous a accordé. Son savoir, son savoir faire, son savoir être et sa bienveillance nous ont été d'un grand apport.

Nous tenons à exprimer nos remerciements aux membres du jury, qui ont bien voulu juger ce mémoire.

Qu'il nous soit permis de remercier également tous les enseignants du département d'informatique qui ont contribué à notre formation.

Enfin que tous ceux qui ont participé de près ou de loin à l'aboutissement de notre parcours universitaire trouvent ici l'expression de notre profonde gratitude.

Résumé

Ce projet de fin d'études a pour objectif de développer pour la première fois, un système intelligent novateur pour la détection de la syncope et la prévention des dangers de la plongée en apnée. En utilisant la méthode d'apprentissage supervisé Random Forest, nous avons entraîné un modèle capable de prédire les risques de syncope en se basant sur les données environnementales et physiologiques des plongeurs telles que la profondeur, l'accélération du mouvement et le temps. Ce système est conçu pour émettre des alertes précoces, contribuant ainsi à augmenter la sécurité lors de la plongée en apnée. En outre, ce travail comprend une revue approfondie des méthodes d'apprentissage automatique comme l'algorithme du Random Forest de l'apprentissage supervisé ainsi que les outils de développement. Enfin, quatre scénarios probables sont envisagés, discutés et présentés par émulation.

Mots clés : Apnée, syncope, machine learning, détection, randomforest, émulation

Abstract

This project aims to develop for the first time, an innovative intelligent system for the detection of syncope and the prevention of the dangers of freediving. Using the Random Forest supervised learning method, we trained a model capable of predicting the risk of syncope based on the environmental and physiological data of divers such as depth, acceleration of movement. This system is designed to issue early warnings, helping to increase safety when snorkeling. Furthermore, this work includes an in-depth review of machine learning methods such as the Random Forest algorithm of supervised learning as well as development tools. Finally, four probable scenarios are considered, discussed and presented by emulation.

Keywords: Apnea, syncope, machine learning, detection, random forest, emulation

ملخص

يهدف مشروع نهاية الدراسات هذا إلى تطوير نظام ذكي مبتكر لأول مرة لاكتشاف الإغماء والوقاية من مخاطر الغوص الحر. باستخدام طريقة التعلم تحت إشراف Forest Random، قمنا بتدريب نموذج قادر على التنبؤ بخطر الإغماء بناءً على البيانات البيئية والفسولوجية للغواصين مثل العمق وتسريع الحركة. تم تصميم هذا النظام لإصدار تحذيرات مبكرة، مما يساعد على زيادة السلامة عند الغطس وكذلك أدوات التطوير. وأخيرًا، يتم النظر في أربعة سيناريوهات محتملة ومناقشتها وتقديمها عن طريق المحاكاة. علاوة على ذلك، يتضمن هذا العمل مراجعة متعمقة لطرق التعلم الآلي مثل خوارزمية الغابات العشوائية للتعلم تحت الإشراف.

الكلمات المفتاحية: انقطاع النفس، الإغماء، التعلم الآلي، الكشف، الغابة العشوائية، المحاكاة

Sommaire :

Introduction générale	1
CHAPITRE I: NOTIONS DE BASE : Introduction	3
I.1 Internet des objets (IoT)	3
I.2. L'e-Santé	5
I.3. Plongée sous-marine en apnée	7
I.4. Dangers de la plongée sous-marine	8
Conclusion	8
CHAPITRE II: ETAT DE L'ART : Introduction	9
II.1 Types de Machine Learning	9
II.1.1 Apprentissage supervisé	10
II.1.1.1. Régression logistique	11
II.1.1.2. Machines à vecteurs de support (SVM)	11
II.1.1.3. Arbres de décision	11
II.1.1.4. Forêts aléatoires	12
II.1.2. Apprentissage non supervisé	13
II.1.2.1. Clustering	13
II.1.2.2. Réduction de dimensionnalité	14
II.1.3. Apprentissage par Renforcement	14
II.2. Raisons de choix la méthode de Random Forest	14
II.2.1. Supervision et étiquetage des données	15
II.2.2. Apprentissage direct et ciblé	15
II.2.3. Performance et précision	15
II.2.4. Adaptabilité et évolutivité	16
Conclusion	18
CHAPITRE III: SOLUTIONS PROPOSEES : Introduction	19
III.1. Description générale	19
III.2. Détection de danger en profondeur	23
III.3. Détection de danger en ascension	25
III.4. Algorithme de régulation de la durée de plongée	27
III.5. Algorithme de calibration	29
Conclusion	32

CHAPITRE IV: EXPERIMENTATIONS ET RESULTATS : Introduction	33
IV.1 Matériel utilisé	33
IV.1.1. Accéléromètre MPU 6050	33
IV.1.2. Arduino UNO et Arduino NANO	34
IV.1.3. NRF24L01 Wireless Transmitter	35
IV.1.4. Capteur de pression d'eau	36
IV.2. Présentation des résultats et émulation	37
IV.2.1. Mise en oeuvre de l'objet connecté:	37
IV.2.2. Résultats et intégration du modèle Machine Learning sur Arduino	38
IV.2.2.1. Résultats et évaluation du modèle Machine Learning	39
IV.2.2.2. Intégration avec Arduino en utilisant micromlgen	39
IV.2.2.3. Implémentation et utilisation	39
IV.2.2. Emulation	40
IV.2.2.1 Premier Scénario : Plongée régulière sans danger	40
IV.2.2.2 Deuxième Scénario : Danger dû à un arrêt prolongé en profondeur	41
IV.2.2.3 Troisième Scénario : Danger potentiel lors de la remontée	42
IV.2.2.4 Quatrième Scénario : Danger à la surface	43
Conclusion	44
Conclusion générale	45

ACRONYMES

IoT : Internet of Things

LED : Light Emitting Diode

IDE : Integrated Development Environment

4G : Fourth-generation

5G : Fifth-generation

IEEE : Institute of Electrical and Electronics Engineers

IETF : Internet Engineering Task Force

WPAN : Wireless Personal Area Network

WLAN : Wireless Local Area Network

LISTE DES FIGURES

CHAPITRE I : Notions de Base	
Figure I.1: Différents domaines d'applications de l'IoT	4
Figure I.2 : Applications de l'e-Santé.	6
CHAPITRE II : État de l'Art	
Figure II.1 : Types de machine Learning	10
Figure II.2: Schéma de l'apprentissage supervisé	10
Figure II.3: Exemple d'un arbre de décision d'attribution des mentions en fonction des notes	12
Figure II.4: Représentation des forêts aléatoires	13
CHAPITRE III : Solutions proposées	
Figure III.1 : Organigramme général	20
Figure III.2 : Organigramme de la détection du danger en profondeur.	24
Figure III.3 : Organigramme de la détection du danger en ascension.	26
Figure III.4 : Organigramme de régulation de la durée de plongée.	28
Figure III.5 : Organigramme de calibration.	31
CHAPITRE IV : Expérimentation et Résultats	
Figure IV.1 : Image d'un accéléromètre MPU6050.	34
Figure IV.2 : Photographie des Arduino Nano et Uno.	35
Figure IV.3 : Image du NRF24L01 Wireless Transmitter.	36
Figure IV.4 : Image d'un capteur de pression SEN0257.	37
Figure IV.5 : Système Emetteur de l'objet connecté.	38
Figure IV.6 : Système Receveur de l'objet connecté.	38
Figure IV.7 : Scénario 1 de plongée en apnée.	41
Figure IV.8 : Scénario 2 de plongée en apnée.	42
Figure IV.9 : Scénario 3 de plongée en apnée.	43
Figure IV.10 : Scénario 4 de plongée en apnée.	44

LISTE DES TABLEAUX

CHAPITRE II : État de l'Art	
Tableau II.1 : Comparaison entre l'apprentissage supervisé, non supervisé, et renforcé pour la détection des dangers en apnée.	17
Tableau II.2 : Comparaison entre les algorithmes d'apprentissage supervisé pour la détection des dangers en apnée.	18

INTRODUCTION GENERALE

Introduction générale

La plongée en apnée, également connue sous le nom de plongée libre, est une activité pratiquée dans le monde entier pour le sport, la détente et la découverte du monde sous-marin. Toutefois, cette pratique comporte des risques, avec environ 50 décès chaque année liés aux accidents de plongée en apnée [1]. La syncope, une perte de conscience causée par une privation d'oxygène dans le cerveau, est l'un des dangers les plus mortels associés à cette activité [2]. Malheureusement à ce jour, et malgré l'ampleur de ce problème, aucun travail de recherche n'a été consacré à la détection précoce de la syncope en plongée sous marine et à la prévention des dangers associés. Ainsi, il est essentiel de mettre en place des mesures de prévention et de sécurité pour protéger les plongeurs et réduire le nombre d'accidents et c'est bien dans ce contexte que notre travail s'inscrit.

L'Internet des objets (IoT) et le Machine Learning sont deux technologies qui ont révolutionné divers domaines de la société numérique, y compris la prévention des risques et l'e-santé [3]. Dans ce mémoire, nous explorons comment ces technologies pourraient être utilisées pour développer un objet connecté intelligent pour la prévention des accidents de plongée en apnée, en particulier ceux liés à la syncope.

La problématique abordée dans ce travail est la suivante : comment concevoir et développer un objet connecté novateur capable de détecter les accidents de syncope lors d'une séance de plongée en apnée et de prévenir la personne la plus proche du plongeur pour une éventuelle intervention de sauvetage. Nous pensons qu'un tel objet, intégrant un système de détection de syncope basé sur le Machine Learning, pourrait optimiser les résultats obtenus et offrir de meilleures expériences pour les plongeurs.

Les objectifs principaux de ce projet sont les suivants :

- ✓ Développer un prototype d'objet connecté pour détecter les syncopes lors de la plongée en apnée.
- ✓ Utiliser le Machine Learning pour optimiser la détection et offrir une meilleure expérience.
- ✓ Réduire le nombre d'accidents et de décès liés à la plongée en apnée.

Pour atteindre ces objectifs, nous utiliserons une approche technique combinant l'Arduino, des capteurs de pression et des accéléromètres, ainsi que des émetteurs sans fil et des lampes témoins électroluminescente (LED). Le développement logiciel sera effectué à l'aide de d'un environnement à développement intégré (IDE Arduino), et la méthode de Machine Learning choisie pour ce projet est la technique des Random Forest, dont les avantages et les inconvénients seront discutés en détail dans les chapitres suivants.

Ce mémoire est structuré en quatre chapitres principaux. Le premier chapitre présente les notions de base sur l'IoT, l'e Santé, les dangers de la plongée sous-marine et la syncope. Le deuxième chapitre aborde l'état de l'art, en examinant diverses méthodes de Machine Learning et en justifiant le choix de la méthode Random Forest. Le troisième chapitre décrit en détail les solutions proposées utiles au développement d'un objet connecté. Quant au quatrième chapitre, il traite du matériel utilisé et la discussion des résultats de l'émulation. Le mémoire se terminera par une conclusion générale et des perspectives.

Chapitre I :

NOTIONS DE BASE

Introduction:

Ce chapitre sert d'introduction aux principaux concepts nécessaires à la compréhension de ce travail. Il abordera les notions de base de l'Internet des objets, de l'e-Santé, de la plongée en apnée, ainsi que les dangers associés à cette dernière.

I.1 Internet des objets (IoT) :

L'Internet des objets est un concept qui fait référence à la connexion d'objets physiques à Internet, permettant ainsi une communication bidirectionnelle entre ces objets et d'autres dispositifs ou systèmes [4]. Les objets connectés sont souvent équipés de capteurs et d'actuateurs qui leur permettent de collecter des données et d'interagir avec leur environnement. Les données collectées sont ensuite transmises à des plateformes cloud ou à des systèmes de traitement pour être analysées et utilisées dans diverses applications [5]. L'évolution rapide de l'IoT a été soutenue par des avancées dans les domaines des réseaux sans fil, de l'informatique en nuage, des capteurs et de l'intelligence artificielle [6].

Les applications de l'IoT sont nombreuses et variées, touchant divers secteurs tels que l'industrie, la santé, l'agriculture, les transports et la gestion de l'énergie. Par exemple, dans le secteur industriel, l'IoT est utilisé pour la maintenance prédictive, le suivi de la production en temps réel et l'automatisation des processus de fabrication [7]. Dans le domaine de la santé, l'IoT permet la télésurveillance des patients, la gestion des maladies chroniques et la détection précoce des problèmes de santé [8]. Dans l'agriculture, l'IoT est utilisé pour la gestion des cultures et de l'élevage, ainsi que pour l'optimisation de l'utilisation des ressources en eau et en énergie [9]. Enfin, dans le secteur des transports, l'IoT est utilisé pour le suivi en temps réel des véhicules, la gestion du trafic et la sécurité routière [10] comme le montre la figure I.1.

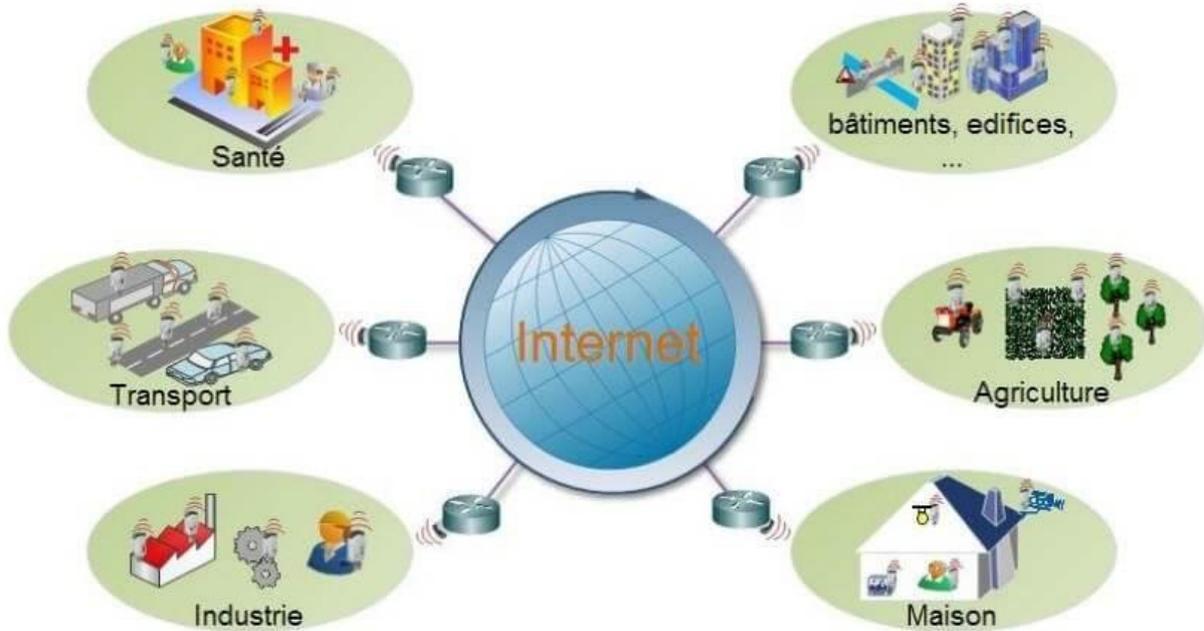


Figure I.1. Différents domaines d'applications de l'IoT [11]

Les technologies clés qui soutiennent l'IoT comprennent les réseaux de communication sans fil, les capteurs et les dispositifs de traitement embarqués. Les réseaux sans fil, tels que les réseaux cellulaires (4G, 5G), les réseaux locaux sans fil (Wi-Fi) et les réseaux basse consommation et longue portée (LoRa, Sigfox), permettent aux objets connectés de communiquer entre eux et avec les systèmes de traitement des données [12].

Les capteurs sont des dispositifs qui mesurent des paramètres physiques ou chimiques, tels que la température, l'humidité, la pression, la lumière, les gaz et les vibrations, et convertissent ces mesures en signaux électriques pouvant être traités par les systèmes informatiques [13]. Les dispositifs de traitement embarqués, tels que les microcontrôleurs et les microprocesseurs, permettent aux objets connectés de traiter les données collectées par les capteurs et d'exécuter des algorithmes pour prendre des décisions ou commander des actionneurs [14].

La sécurité et la protection de la vie privée sont des préoccupations majeures dans le développement et le déploiement de l'IoT. Les objets connectés sont souvent exposés à des cyberattaques, telles que les attaques par déni de service, l'usurpation d'identité et l'accès non autorisé aux données [15]. Pour assurer la sécurité et la protection de la vie privée des utilisateurs, les solutions IoT doivent intégrer des mécanismes de sécurité robustes, tels que le

chiffrement des données, l'authentification des utilisateurs et des dispositifs, la détection des intrusions et les mises à jour de sécurité régulières [16]. De plus, les lois et les réglementations sur la protection des données, comme le Règlement général sur la protection des données (RGPD) en Europe, imposent aux concepteurs et aux opérateurs de systèmes IoT de prendre des mesures pour garantir la confidentialité et la sécurité des données personnelles des utilisateurs [17].

Un autre défi important pour le développement et l'adoption de l'IoT est l'interopérabilité entre les différents dispositifs, protocoles et normes [18]. Les objets connectés proviennent de différents fabricants et utilisent souvent des protocoles de communication et des formats de données propriétaires. Afin de faciliter l'intégration et la communication entre ces dispositifs, il est nécessaire d'établir des normes ouvertes et des protocoles communs qui permettent l'échange de données et la collaboration entre les différents systèmes IoT [19].

Plusieurs organisations, telles que l'IEEE, l'IETF et l'Open Connectivity Foundation, travaillent sur l'élaboration de normes et de protocoles pour l'IoT afin de promouvoir l'interopérabilité et la scalabilité des solutions IoT [20].

I.2. L'e-Santé :

La notion d'E-santé, désignée également par E-health, télésanté ou cyber santé, est utilisée pour la première fois en 1999 par Jhon Michelle [21].

La définition la plus connue et la plus utilisée est celle proposée par Eysenbach [22] : « E-health est un domaine émergent créé par l'intersection entre les domaines de l'informatique médicale, la santé publique et le secteur privé. Ce domaine fait référence au service de santé et les informations en matière de santé à travers Internet ou d'autres technologies connexes...

L'intégration des nouvelles technologies dans le domaine de la santé publique est la clé qui a permis d'ouvrir une nouvelle époque d'évolution au niveau matériel afin de fournir une multitude d'informations qui n'étaient pas disponibles par le passé.

Par ailleurs, l'e-santé, est un domaine en pleine croissance qui utilise les technologies de l'information et de la communication pour améliorer la qualité des soins de santé, la prévention des maladies et la recherche médicale [23]. Les applications de l'e-santé sont variées et incluent

la télémédecine, les dossiers médicaux électroniques, les dispositifs médicaux connectés et les systèmes de surveillance à distance [24] comme le montre la figure I.2 ci-dessous.

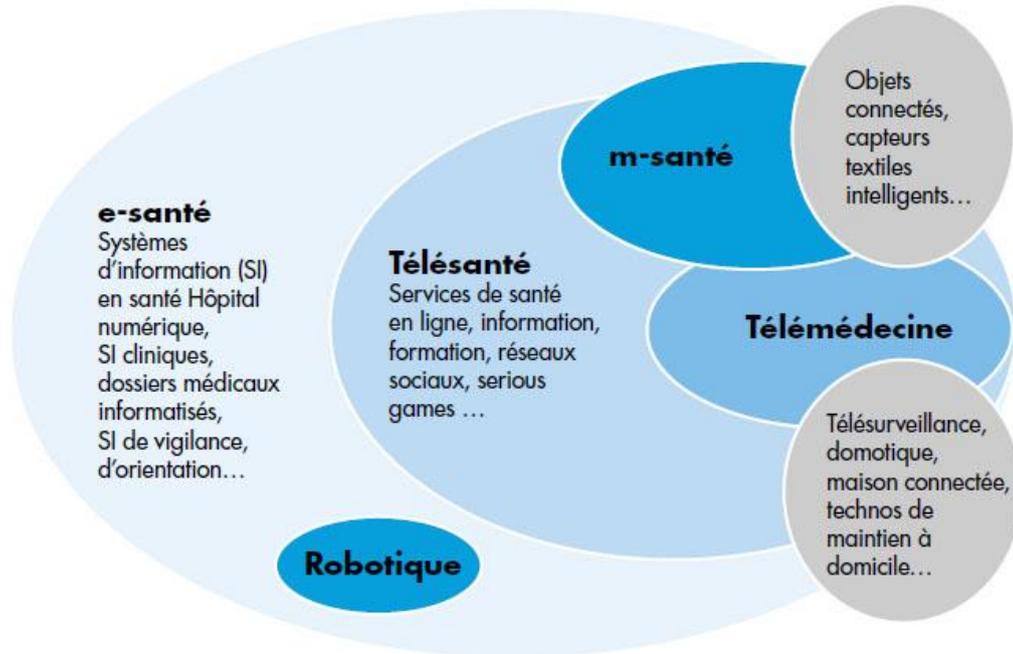


Figure I.2. Applications de l'e-Santé [25].

La télémédecine est une application clé de l'e-santé qui permet aux professionnels de la santé de consulter leurs patients à distance via des plateformes de vidéoconférence ou d'autres outils de communication en ligne [26]. Cela est particulièrement utile dans les zones rurales ou éloignées où l'accès aux soins médicaux est limité et peut également réduire les coûts associés aux déplacements et à l'infrastructure médicale [27].

Les objets connectés pour la santé (m santé), tels que les montres intelligentes, les bracelets de fitness et les glucomètres, peuvent aider les individus à suivre leurs activités physiques, leur sommeil, leur glycémie et d'autres paramètres de santé, encourageant ainsi un mode de vie plus sain [28]. La détection des situations anormales se fait donc d'une manière instantanée et l'intervention humaine ne s'exécute qu'en cas de besoin. Différentes applications sont proposées dans ce cadre et peuvent être réparties en trois classes [29] : Soins à domicile, Soins à la résidence spécialisée, Soins intensifs. L'analyse des données collectées par ces dispositifs peut contribuer à la recherche médicale en fournissant de précieuses informations sur les tendances et les corrélations entre les facteurs de risque et les maladies [30].

En outre, l'e-santé englobe également l'utilisation de l'intelligence artificielle (IA) et du Big Data pour analyser et exploiter de vastes ensembles de données médicales, ce qui peut améliorer la détection précoce des maladies, la personnalisation des traitements et la prédiction des résultats pour les patients [31].

Enfin, Il devient alors nécessaire de mettre en évidence le problème de la différenciation des services relatifs à ces informations. Ainsi, le nombre de WPAN (Wireless Personal Area Networks) et WLAN (Wireless Local Area Networks) utilisés dans les hôpitaux, maisons de retraite ou à domicile est en constante augmentation.

La qualité et l'importance des données collectées demandent un certain niveau d'assurance en termes de fiabilité du réseau : l'information doit arriver au récepteur ou au point de collecte d'information de façon certaine, avec une latence maîtrisée et en temps réels avec différents niveaux de priorité. Les services e-santé doivent alors être totalement fiables et efficaces.

I.3. Plongée sous-marine en apnée :

La plongée sous-marine en apnée, également connue sous le nom de plongée libre, est une activité passionnante et exigeante qui permet aux plongeurs de découvrir les merveilles de la vie sous-marine sans l'utilisation d'un équipement de plongée autonome. Cette forme de plongée exige une grande maîtrise de la respiration, de la relaxation et de la technique de descente pour permettre aux plongeurs de descendre en profondeur sans l'aide d'une source d'air supplémentaire. Cependant, la plongée libre peut être dangereuse si elle n'est pas pratiquée avec une formation appropriée, une technique adéquate et un équipement de sécurité adapté.

Il existe différents types d'apnée :

- L'apnée dynamique qui consiste à parcourir la plus grande distance possible sous l'eau sans reprendre sa respiration,
- L'apnée statique consiste à retenir sa respiration le plus longtemps possible en état immobile,
- L'apnée en profondeur se pratique le long d'un câble vertical, qui va servir de guide pour la descente et la remontée.

I.4. Dangers de la plongée sous-marine :

La plongée en apnée présente plusieurs dangers et risques pour la santé de plongeurs [30]. On cite les accidents de décompression [33] et le barotraumatisme qui peut non seulement affecter les oreilles et les sinus (barotraumatisme de l'oreille moyenne et barosinusite), mais aussi les poumons (barotraumatisme pulmonaire), les dents (barotraumatisme dentaire) et les yeux (barotraumatisme oculaire) [34]. Cependant, le risque majeur en plongée sous-marine en apnée est la syncope ou “la samba”. Il s’agit d’une perte de connaissance violente qui s’accompagne souvent de tremblements, et qui se suit d’une noyade si rien n’est fait à temps pour dégager les voies aériennes hors de l’eau.

Dans le cas de l’apnée statique ou dynamique, la syncope est fréquente car certains apnéistes n’hésitent pas à repousser leurs limites jusqu’au point de rupture, notamment lors de records du monde. Tandis que pour la plongée profonde, la syncope survient généralement dans la zone entre 0 et 10 mètres autrement appelée “rendez-vous syncopal des 7 mètres” car les variations de pressions sont les plus importantes à ce niveau. La pression en oxygène dans le sang baisse subitement et le cerveau décide alors de couper son activité pour privilégier les organes vitaux qui maintiennent le corps en activité minimale. En revanche, il est arrivé que des syncopes surviennent bien plus profond donnant lieu à des disparitions brutales.

Il arrive que l’apnéiste soit victime d’une syncope même en surface. En effet, la sensation de détente dans les profondeurs conduit l’apnéiste à séjourner plus longtemps au fond car il ne ressent pas encore le besoin de remonter. Il utilise plus complètement ses réserves d'O₂. Lorsqu’il ressent trop tardivement ce besoin de respirer, il déclenche sa remontée. La consommation importante d'O₂ durant le séjour au fond, couplée avec la chute de la pression absolue, aboutit à une hypoxie sévère à la surface, qui entraînera une perte de connaissance.

Conclusion:

Elle présente une introduction complète à divers concepts fondamentaux, y compris l'IoT, l'e-Santé, et les divers dangers liés à la plongée en apnée. Ces connaissances serviront de base pour le prochain chapitre, où nous examinerons l'état de l'art des techniques de machine learning applicables à notre système de détection de danger.

Chapitre II :

ETAT DE L'ART

Introduction:

Cette partie de notre travail se focalise sur l'état actuel des techniques de Machine Learning, mettant l'accent sur pourquoi l'algorithme Random Forest a été choisi pour notre système de détection de danger. Une comparaison entre les différents algorithmes de Machine Learning sera également présentée.

II.1 Types de Machine Learning

Le Machine Learning est une branche de l'intelligence artificielle qui se concentre sur le développement d'algorithmes capables d'apprendre à partir de données et d'ajuster leurs comportements en conséquence [35].

L'apprentissage automatique, ou Machine Learning, se réfère à l'exploitation scientifique d'algorithmes et de modèles statistiques que les systèmes informatiques utilisent pour accomplir des tâches spécifiques sans être explicitement programmés [36]. Ces algorithmes sont largement utilisés dans des applications de tous les jours, comme les véhicules autonomes, la reconnaissance d'images, ou encore les moteurs de recherche sur internet [37]. Par exemple, lorsque nous utilisons un moteur de recherche comme Google, il fournit des résultats pertinents en partie grâce à des algorithmes d'apprentissage qui ont été formés pour classer efficacement les pages web [38].

L'exploration de données, le traitement d'images, l'analyse prédictive sont quelques-uns des nombreux domaines où ces algorithmes sont mis en œuvre [39]. L'un des principaux avantages de l'apprentissage automatique est qu'une fois qu'un algorithme a appris à traiter certaines données, il peut faire son travail de manière autonome [40].

Il existe principalement trois types de Machine Learning: l'apprentissage supervisé, l'apprentissage non supervisé et l'apprentissage par renforcement, comme illustré à la figure II.1.

Types of Machine Learning

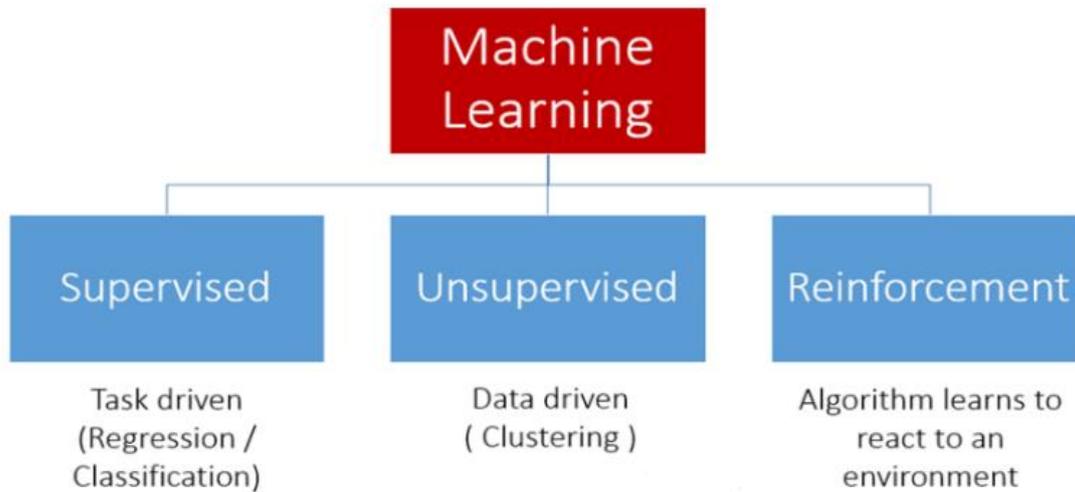


Figure II.1. Types de machines learning [41]

I.1.1 Apprentissage supervisé

L'apprentissage supervisé est la méthode la plus répandue dans le domaine du Machine Learning. Cette approche est dirigée par une connaissance préexistante des résultats attendus du modèle. Les algorithmes sont entraînés sur un ensemble de données labellisées, ce qui signifie que chaque exemple d'entraînement est accompagné de la réponse appropriée, ou "cible" [42]. Les algorithmes d'apprentissage supervisé utilisent ces données pour apprendre à effectuer des prédictions ou des classifications sur de nouvelles données non labellisées [43]. La figure II.2 ci dessous schématise ce type d'apprentissage.

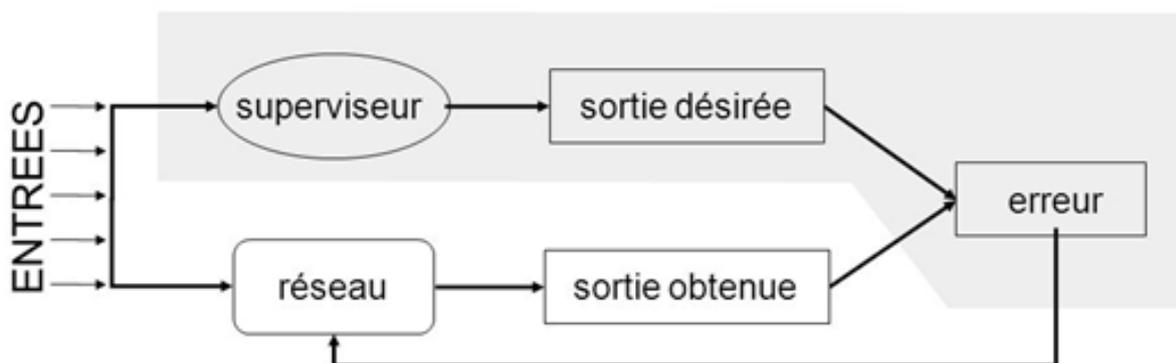


Figure II.2. Schéma de l'apprentissage supervisé [44]

Les problèmes résolus par l'apprentissage supervisé sont principalement classifiés en deux catégories: la classification et la régression. Dans les problèmes de classification, l'objectif est de prédire la catégorie ou la classe d'un nouvel exemple, sur la base d'un ensemble de caractéristiques. Ces problèmes peuvent être binaires, où il n'y a que deux classes possibles, ou multiclasse, avec plusieurs catégories à prédire. Divers algorithmes de classification sont couramment utilisés, chacun ayant ses propres avantages et inconvénients.

II.1.1.1. Régression logistique:

La régression logistique est un algorithme de classification linéaire bien adapté aux problèmes de classification binaire. Il modèle la probabilité qu'un exemple appartienne à une certaine classe en utilisant une fonction logistique. La régression logistique est simple et rapide à implémenter, mais elle peut être moins efficace lorsque les relations entre les caractéristiques et les classes ne sont pas linéaires [45].

II.1.1.2. Machines à vecteurs de support (SVM):

Les SVM sont des algorithmes de classification qui cherchent à trouver l'hyperplan de séparation optimal entre les classes en maximisant la marge entre les exemples les plus proches de cet hyperplan. Les SVM peuvent être utilisés pour les problèmes de classification binaire et multiclasse et sont efficaces même avec un petit nombre d'exemples d'entraînement. Cependant, leur complexité de calcul peut augmenter rapidement avec la taille de l'ensemble de données [46].

II.1.1.3. Arbres de décision:

Les arbres de décision sont des modèles de classification hiérarchique qui fonctionnent en partitionnant l'espace des caractéristiques de manière récursive. À chaque nœud de l'arbre, une caractéristique est choisie pour diviser les données en sous-ensembles, et cette procédure est répétée jusqu'à ce que les sous-ensembles soient suffisamment homogènes, comme le montre la figure II.3. Les arbres de décision sont faciles à interpréter et peuvent capturer des relations non linéaires, mais ils sont sensibles au bruit et peuvent être sujets au surapprentissage [47].

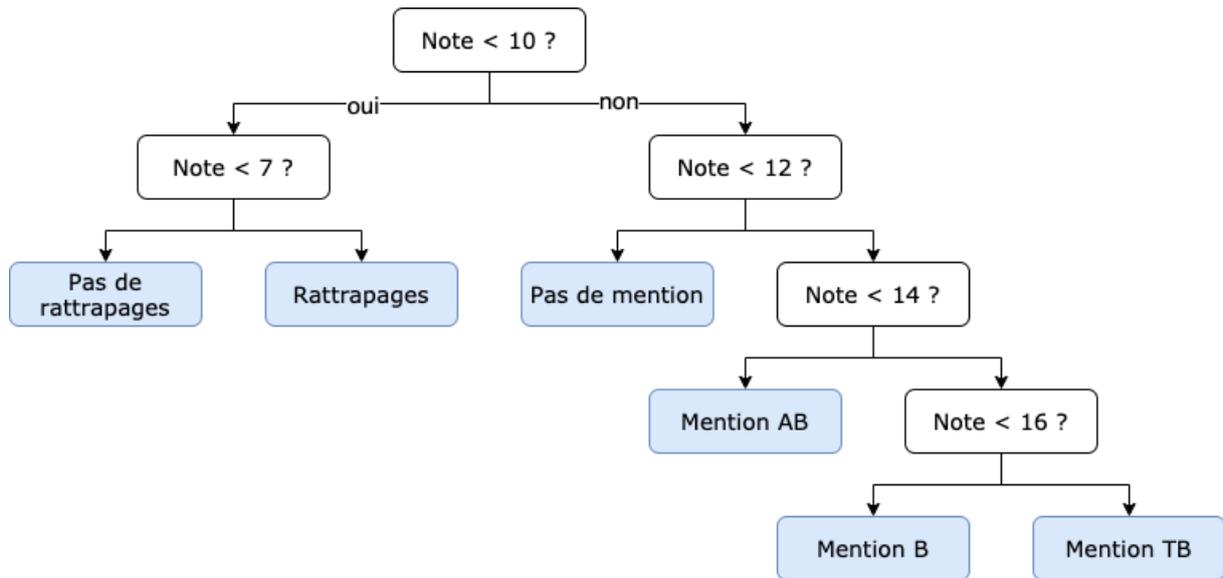


Figure II.3. Exemple d'un arbre de décision d'attribution des mentions en fonction des notes [48]

II.1.1.4. Forêts aléatoires:

Les forêts aléatoires sont des ensembles d'arbres de décision, où chaque arbre est construit en utilisant un sous-ensemble aléatoire de caractéristiques et d'exemples d'entraînement.

La prédiction finale est obtenue en combinant les prédictions de tous les arbres par un vote majoritaire. Les forêts aléatoires (figure II.4) sont robustes face au bruit, capables de gérer des relations non linéaires et généralement moins sujettes au surapprentissage que les arbres de décision individuels [49].

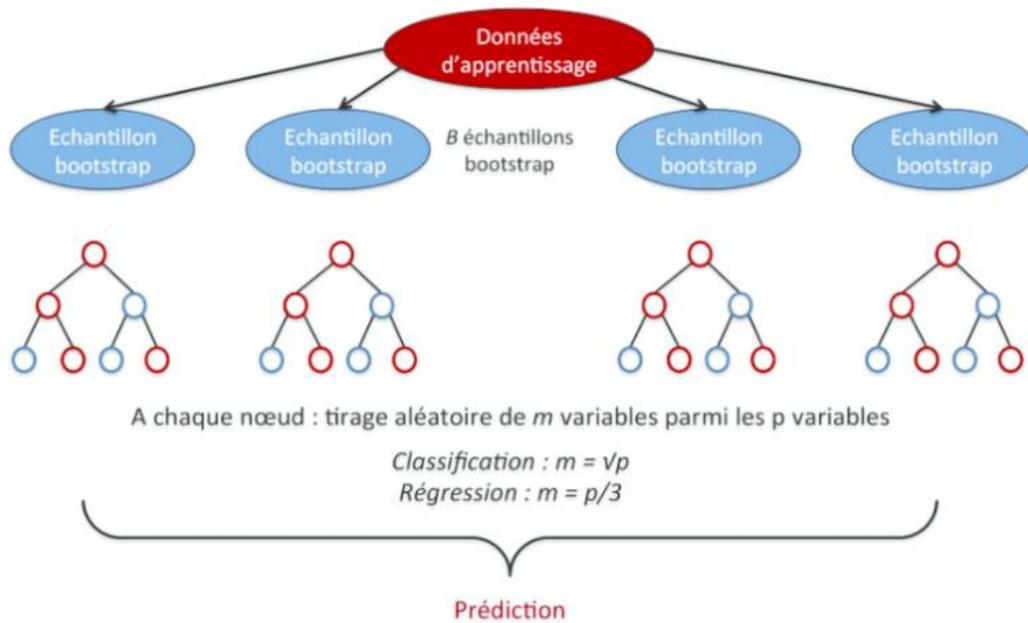


Figure II.4. Représentation des forêts aléatoires [50]

II.1.2. Apprentissage non supervisé

L'apprentissage non supervisé est un aspect du Machine Learning qui s'occupe des problèmes pour lesquels les données d'entraînement n'incluent pas de cibles ou d'étiquettes. L'objectif principal de l'apprentissage non supervisé est de découvrir les structures cachées, les motifs ou les relations dans les données sans utiliser de données étiquetées pour guider le processus d'apprentissage. Les algorithmes d'apprentissage non supervisé sont généralement classés en deux catégories principales à savoir, le clustering et la réduction de dimensionnalité.

II.1.2.1. Clustering :

Le clustering est une technique d'apprentissage non supervisé qui cherche à partitionner un ensemble de données en sous-groupes homogènes, appelés clusters, sur la base de la similarité entre les exemples. Les algorithmes de clustering cherchent à maximiser la similarité intra-cluster et à minimiser la similarité inter-cluster.

Parmi les algorithmes de clustering couramment utilisés, on cite K-means [51], DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [52] et le Clustering hiérarchique [53].

II.1.2.2. Réduction de dimensionnalité:

La réduction de dimensionnalité est une autre technique d'apprentissage non supervisé qui vise à transformer les données de haute dimension en données de plus faible dimension, tout en préservant le plus possible les relations et les structures importantes dans les données.

La réduction de dimensionnalité est souvent utilisée pour faciliter la visualisation des données, améliorer les performances des algorithmes d'apprentissage et réduire la complexité computationnelle. Parmi les méthodes de réduction de dimensionnalité les plus courantes, on trouve l'analyse en composante principale (PCA) [54], t-SNE (t-distributed Stochastic Neighbor Embedding) [55] et les auto-encodeurs [56].

II.1.3. Apprentissage par Renforcement

L'Apprentissage par Renforcement (Reinforcement Learning, RL) constitue une autre approche dans le domaine de l'apprentissage automatique. Il permet à un agent d'apprendre à travers ses interactions avec son environnement en prenant des décisions qui maximisent une récompense cumulée sur le long terme [57]. Par opposition à l'apprentissage supervisé et non supervisé, l'accent est mis sur la prise de décision et la résolution de problèmes séquentiels plutôt que sur la prédiction ou la détection de structures dans les données [58].

Un problème d'apprentissage par renforcement [59] peut se décomposer en plusieurs éléments clés tels que l'Agent, l'Environnement, l'État, l'Action et la Récompense.

De nombreux algorithmes d'apprentissage par renforcement existent, chacun se distinguant par la manière dont il estime les valeurs d'actions, représente et met à jour les connaissances de l'agent, et explore et exploite l'environnement.

II.2. Raisons de choix la méthode de Random Forest

Plusieurs facteurs ont motivé notre choix de la méthode Random Forest pour le développement de notre système intelligent de détection de syncope et de prévention des risques de la plongée en apnée.

Ce choix se distingue par rapport aux autres approches d'apprentissage supervisé, non supervisé

et par renforcement. Les raisons principales de ce choix sont:

II.2.1. Supervision et étiquetage des données:

À l'opposé de l'apprentissage non supervisé, l'apprentissage supervisé, comme la méthode Random Forest, nécessite des données étiquetées pour l'entraînement.

Dans le contexte de la détection de syncope et de la prévention des risques de la plongée en apnée, des données historiques étiquetées sur les incidents de syncope et les conditions environnementales associées sont disponibles. Ces données étiquetées permettent au modèle d'apprendre à identifier les situations à risque et de fournir des alertes appropriées, ce qui est plus difficile à réaliser avec les méthodes non supervisées [60].

II.2.2. Apprentissage direct et ciblé:

Contrairement à l'apprentissage par renforcement qui requiert une exploration et une interaction avec l'environnement pour apprendre les meilleures actions à entreprendre, l'apprentissage supervisé permet un apprentissage direct et ciblé à partir des données historiques disponibles.

Dans le contexte de la détection de syncope et de la prévention des risques de la plongée en apnée, l'apprentissage supervisé est préféré car il permet au modèle d'apprendre rapidement à partir des données existantes sans avoir à explorer et interagir avec l'environnement sous-marin, ce qui pourrait être difficile et coûteux [61].

II.2.3. Performance et précision:

La méthode Random Forest est reconnue pour sa performance et sa précision dans les tâches de classification et de régression, la rendant particulièrement adaptée à la détection de syncope et à la prévention des risques de la plongée en apnée.

En comparaison avec d'autres méthodes d'apprentissage supervisé, comme les machines à vecteurs de support (SVM) ou les réseaux de neurones artificiels, les forêts aléatoires offrent un bon équilibre entre la complexité du modèle, la capacité de généralisation et la facilité d'interprétation [62].

II.2.4. Adaptabilité et évolutivité:

La méthode Random Forest est adaptable et évolutive, ce qui signifie qu'elle peut être ajustée et améliorée au fil du temps à mesure que de nouvelles données sont collectées et que les connaissances sur les risques de syncope et les dangers de la plongée en apnée évoluent.

Les forêts aléatoires permettent également d'intégrer de nouvelles caractéristiques ou de modifier les paramètres du modèle en fonction des besoins spécifiques des plongeurs et des situations de plongée. Cette adaptabilité et évolutivité sont essentielles pour assurer la pertinence et l'efficacité de notre système intelligent à long terme [63].

Nous présenterons dans le tableau II.1 les avantages et les inconvénients des machines learning. En réalité, chaque méthode est plus ou moins adaptée au problème considéré et à la nature des données. Il faut juste connaître les avantages et les inconvénients de chaque méthode et surtout les expérimenter pour trouver la méthode la plus pertinente. De ce fait, l'estimation de la qualité des méthodes est centrale, mais pas toujours évidente, car dans notre cas par exemple, il nous a été impossible de tester la méthode sur des apnéistes en situation dangereuse. Toutefois, il ressort de la comparaison des méthodes de machine learning, que pour la détection des accidents en plongée sous marine, l'apprentissage supervisé est le plus adapté.

Tableau II.1: Comparaison entre l'apprentissage supervisé, non supervisé, et renforcé pour la détection des dangers en apnée

Méthode d'apprentissage	Avantages	Inconvénients
Supervisée	<ol style="list-style-type: none"> 1. Permet de prévoir les dangers en se basant sur des exemples précédents. 2. Fournit une performance mesurable et peut s'adapter à l'évolution des comportements de plongée. 	<ol style="list-style-type: none"> 1. Nécessite des données étiquetées, c'est-à-dire des scénarios de plongée avec des dangers identifiés. 2. Moins adapté à la découverte de dangers inconnus ou nouveaux
Non Supervisée	<ol style="list-style-type: none"> 1. Peut découvrir des dangers ou des comportements de plongée anormaux non connus à l'avance. 2. Utile quand il n'y a pas suffisamment de données étiquetées sur les dangers 	<ol style="list-style-type: none"> 1. Les résultats peuvent être moins précis et plus difficiles à interpréter. 2. Moins adapté à la prévention précise et rapide des dangers.
Renforcée	<ol style="list-style-type: none"> 1. Peut s'adapter en temps réel et apprendre de nouvelles stratégies pour la détection des dangers. 2. Capable de prendre en compte des récompenses à long terme, comme la sécurité du plongeur sur plusieurs plongées. 	<ol style="list-style-type: none"> 1. Difficile à mettre en œuvre et peut nécessiter beaucoup d'interactions pour obtenir de bons résultats. 2. Moins adapté à une mise en œuvre rapide et simple

Quant au tableau II.2, il résume la comparaison entre les algorithmes d'apprentissage supervisé pour la détection des dangers en apnée sous marine. De cette étude, nous avons opté pour la méthode Random Forest car elle nous semble la plus adaptée du point de vue gestion des données complexes. Bien qu'un plus grand nombre d'arbres peut rendre l'algorithme trop lent pour la prediction en temps réel, il fournit des informations plus précises qui justifient notre choix.

Tableau II.2: Comparaison entre les algorithmes d'apprentissage supervisé pour la détection des dangers en apnée

Algorithme	Avantages	Inconvénients
Random Forest	<ol style="list-style-type: none"> 1. Gère bien les complexités des données de plongée. 2. Robuste et évite le surapprentissage. 	<ol style="list-style-type: none"> 1. Nécessite plus de temps et de puissance de calcul.
Arbre de décision	<ol style="list-style-type: none"> 1. Fournit une explication visuelle claire pour la détection des dangers. 2. Facile à comprendre. 	<ol style="list-style-type: none"> 1. Peut surapprendre et mal généraliser à de nouveaux scénarios.
SVM (Support Vector Machine)	<ol style="list-style-type: none"> 1. Efficace avec des données moins nombreuses mais pertinentes. 2. Capable de modéliser des frontières de décision complexes. 	<ol style="list-style-type: none"> 1. Difficile à régler et à interpréter. 2. Peut être lent avec de grandes quantités de données
Régression Logistique	<ol style="list-style-type: none"> 1. Simple à comprendre et à interpréter. 2. Fonctionne bien avec des variables binaires, comme la détection des dangers 	<ol style="list-style-type: none"> 1. Peut être sensible au surapprentissage.

Conclusion:

Après avoir passé en revue les différentes techniques de Machine Learning, il est évident que l'algorithme Random Forest offre les meilleures performances pour notre système de détection de danger. Ces connaissances seront utilisées dans ce qui suit, pour expliquer la conception et la mise en œuvre de notre solution, en se concentrant sur les algorithmes spécifiques utilisés pour la détection de danger.

Chapitre III :

SOLUTION PROPOSEE

Introduction :

Nous proposerons et discuterons dans le volet suivant, la conception de notre système, y compris les algorithmes spécifiques utilisés pour la détection de danger. Cette section couvrira les algorithmes de détection de danger en profondeur et en ascension, l'algorithme de régulation de la durée de plongée, l'algorithme de calibration, ainsi que l'algorithme général avec leurs organigrammes.

III.1. Description générale :

Le projet est centré sur la sécurité des plongeurs et vise à surveiller en temps réel les paramètres importants de la plongée, tels que la profondeur et les valeurs d'accéléromètre, afin de prévenir les situations de danger. L'objectif principal est d'améliorer la sécurité des plongeurs en détectant les anomalies et en fournissant des alertes en temps réel.

Le système surveille en permanence le plongeur et utilise des algorithmes spécifiques pour calibrer les valeurs d'accéléromètre, évaluer le taux d'ascension, la profondeur, et d'autres paramètres pour déterminer si le plongeur est en état de danger. Si un état de danger est détecté et que le plongeur appuie sur un bouton, l'envoi du signal est annulé et du temps supplémentaire est ajouté au seuil maximal de plongée.

La figure III.1 montre l'organigramme général proposé pour mieux clarifier notre démarche. Il sera suivi d'algorithmes séparés décrivant le danger en profondeur, le danger en ascension, la régulation de la durée de plongée et enfin la calibration.

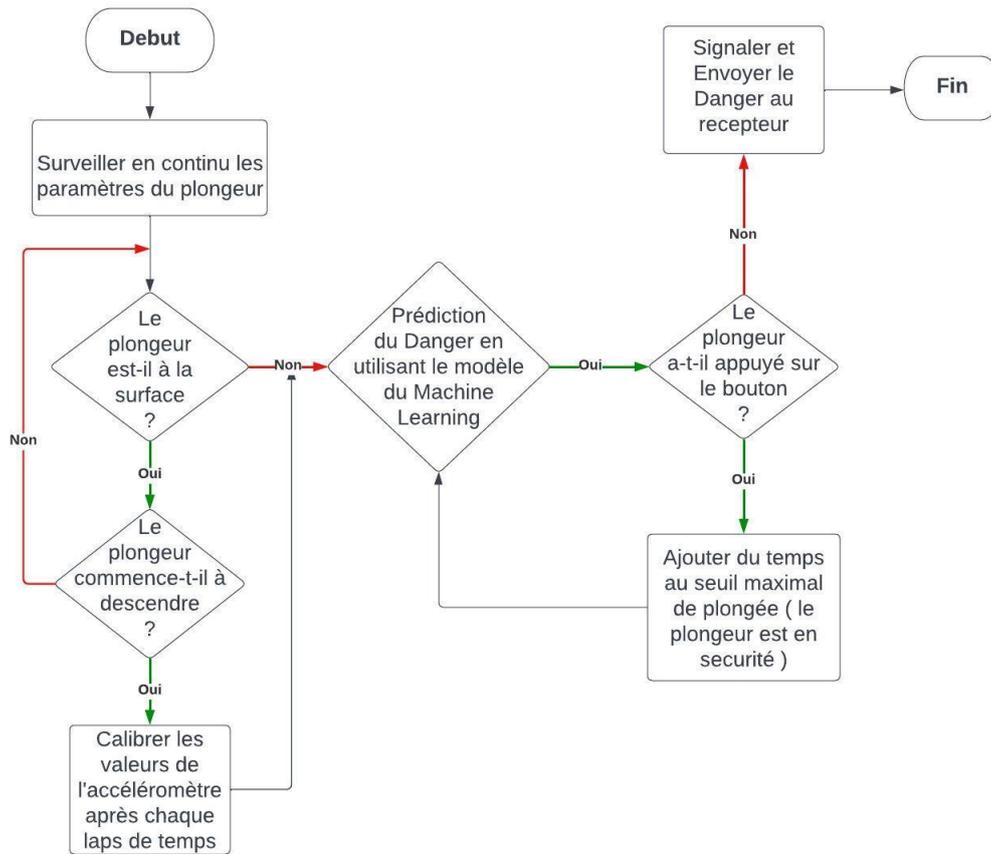


Figure III.1. Organigramme général

La solution proposée se résume en un algorithme général (A) détaillé présenté ci-dessous.

(A) :

Début

Variables : profondeur, acceleration, durée, tempsEtabli, taux_ascension, acceleration_précédente, durée_plongée, durée_totale, durée_moyenne, seuil, seuil_préétabli, nb_plongées : Réel ;

Valeurs_Acceleromètre, Valeurs_Calibrage, Calibrage, Temps, Temps_Dernier_Calibrage : Réel ;

Etat : Chaîne de Caractères

Initialiser toutes les variables // Initialisation des variables

Tant que vrai Faire // Boucle principale de surveillance

```
profondeur, acceleration = Mesurer_paramètres(); // Mesure des paramètres
```

```
Si ( profondeur >= 10 ) et ( accelerationFaible (acceleration) == Vrai ) Faire // Condition de  
profondeur et d'accélération faible
```

```
Compter(durée); // Incrémentation du compteur de durée
```

```
Si ( durée >= tempsEtabli ) Alors // Si la durée dépasse le temps établi
```

```
etat = "Danger"; // État de danger
```

```
Afficher(etat); // Affichage de l'état de danger
```

```
FinSi;
```

```
Si ( boutonActivé() = Vrai ) Alors // Si le bouton d'activation est activé
```

```
Réinitialiser(durée); // Réinitialisation de la durée
```

```
tempsEtabli = tempsEtabli + 15; // Augmentation du temps établi
```

```
FinSi;
```

```
FinSi;
```

```
taux_ascension, acceleration = Mesurer_paramètres(); // Mesure des paramètres pour  
l'ascension
```

```
Si ( taux_ascension > 0 ) et ( acceleration < acceleration_précédente ) Alors // Condition  
de danger lors de l'ascension
```

```
etat = "Danger"; // État de danger
```

```
Afficher (etat); // Affichage de l'état de danger
```

```
FinSi;
```

```
Pour chaque plongée Faire // Enregistrement de la durée de chaque plongée
```

```
Enregistrer(durée_plongée); // Enregistrement de la durée de la plongée
```

```
durée_totale := durée_totale + durée_plongée; // Mise à jour de la durée totale
```

```
nb_plongées := nb_plongées + 1; // Incrémentation du nombre de plongées
```

```
Si ( nb_plongées mod 10 = 0 ) Alors // Si le nombre de plongées est un multiple de 10
```

```
durée_moyenne := durée_totale / 10; // Calcul de la durée moyenne

Si ( durée_moyenne < seuil ) Alors // Si la durée moyenne est inférieure au seuil
    seuil = durée_moyenne + 5; // Mise à jour du seuil
FinSi;

Réinitialiser(durée_totale); // Réinitialisation de la durée totale
FinSi;
FinPour;

Temps = Obtenir_Temps_Courant(); // Obtention du temps courant
Surveiller(Profondeur, Valeurs_Acceleromètre); // Surveillance de la profondeur et de
l'accéléromètre

Si ( Profondeur <= Surface ) Alors // Si la profondeur est inférieure ou égale à la surface
    Tant que ( Profondeur <= Surface ) Faire // Tant que la profondeur est inférieure ou
égale à la surface
        Surveiller(Profondeur, Acceleromètre) // Surveillance de la profondeur et de
l'accéléromètre
    Fin Tantque;

Si ( Temps - Temps_Dernier_Calibrage >= 30_minutes ) Alors // Si le temps depuis le
dernier calibrage est supérieur ou égal à 30 minutes
    Valeurs_Calibrage = Obtenir_valeurs(Valeurs_Acceleromètre,
10_secondes_avant_descente); // Obtention des valeurs de calibrage
    Calibrage := Valeurs_Calibrage; // Mise à jour du calibrage
    Temps_Dernier_Calibrage = Temps; // Mise à jour du temps du dernier calibrage
FinSi;
FinSi;
Fin Tantque;
Fin;
```

III.2. Détection de danger en profondeur:

L'algorithme s'attache à identifier une situation de danger potentiel lorsque le plongeur reste immobile à une certaine profondeur pendant un laps de temps prolongé. Le mouvement est un signe essentiel de la conscience et de la capacité du plongeur à maintenir le contrôle de son activité de plongée. L'immobilité à une profondeur significative peut indiquer plusieurs problèmes graves.

Un plongeur peut être immobile en raison de divers problèmes comme une perte de conscience, un épuisement, une panne d'équipement ou des problèmes de santé soudains. La perte de conscience peut résulter de l'hypoxie, une condition qui survient lorsque le corps ou une région du corps est privé d'oxygène suffisant. Ceci est particulièrement pertinent pour la plongée en apnée, où le plongeur retient son souffle.

Dans le contexte de la plongée, l'immobilité pendant une période prolongée peut également indiquer que le plongeur est pris au piège ou enchevêtré dans des obstacles sous-marins, ou qu'il fait face à d'autres dangers environnementaux. C'est pourquoi il est essentiel de prédire rapidement un état de "Danger" dans ces situations.

L'algorithme fonctionne en surveillant en continu la profondeur à laquelle le plongeur se trouve et l'amplitude de son accélération dynamique.

Si l'amplitude de l'accélération reste faible pendant une période prolongée à une certaine profondeur, cela indique que le plongeur est immobile. Dans ce cas, l'algorithme prédit un état de "Danger", ce qui permet d'initier des mesures d'urgence pour aider le plongeur si nécessaire. La figure III.2 montre l'organigramme de la détection du danger en profondeur.

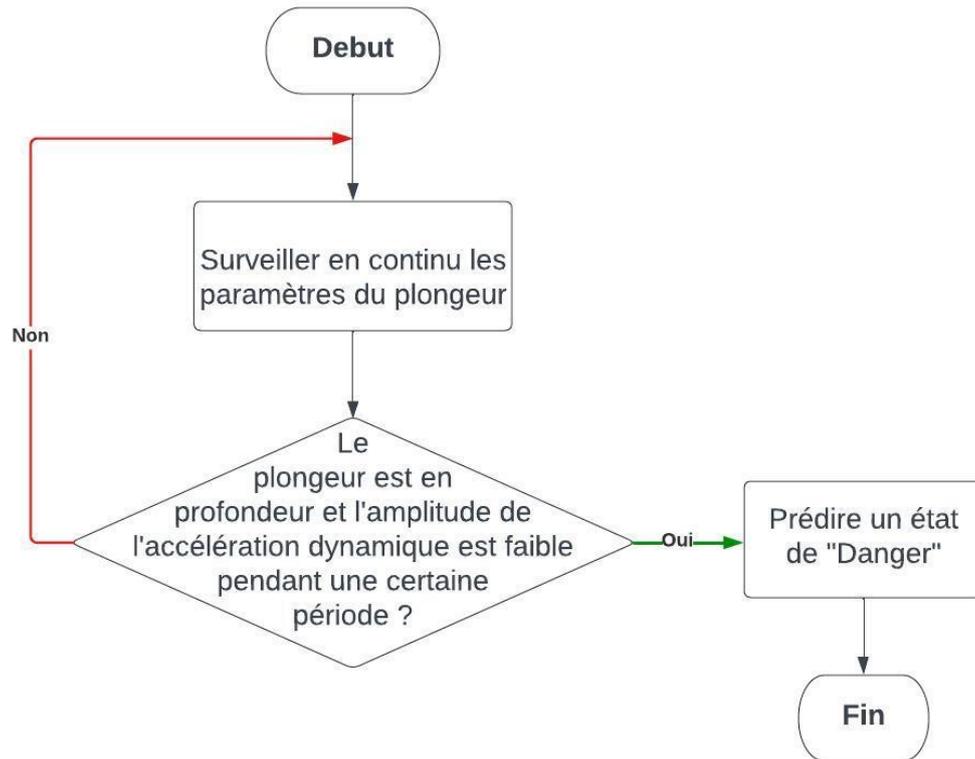


Figure III.2. Organigramme de la detection du danger en profondeur.

L'Algorithme (B) proposé s'écrit alors comme suit:

(B):

Début

Variables : profondeur, acceleration, durée, temps Etabli : Réel ;

Etat : Chaîne de caractères ;

Tantque Vrai Faire // Boucle infinie pour continuer à surveiller le plongeur

profondeur, acceleration = Mesurer_paramètres(); // Prendre des mesures

Si (profondeur ≥ 10) et (accelerationFaible (acceleration) == Vrai) Faire // Vérifier si le plongeur est profondément sous l'eau et s'il est en mouvement lent

compter(durée); // Commencer à compter le temps

Si (durée \geq tempsEtabli) Alors // Vérifier si le plongeur est en état d'immobilité pour un temps plus long que le seuil préétabli

```
etat = "Danger"; // Déclencher un état de danger
afficher(etat); // Afficher le message de danger
FinSi;

Si ( boutonActivé() = Vrai ) Alors // Vérifier si le bouton est pressé
réinitialiser(durée); // Réinitialiser la durée
tempsEtabli = tempsEtabli + 15; // Augmenter le temps établi
FinSi;
FinSi;
FinTantque;
Fin;
```

III.3. Détection de danger en ascension:

Cet algorithme joue un rôle crucial dans la détection d'une situation potentiellement dangereuse pendant l'ascension du plongeur. Dans le cadre de la plongée en apnée, la phase d'ascension est critique car elle comporte plusieurs risques inhérents. L'un d'eux est l'arrêt soudain du mouvement du plongeur.

Un arrêt soudain pendant l'ascension peut être dû à divers problèmes, comme une panne d'équipement ou des problèmes de santé tels que la syncope de l'apnée, qui est une perte de conscience due à une diminution de l'oxygène dans le cerveau. En outre, le plongeur peut également être entravé par des éléments naturels tels que des courants sous-marins puissants.

L' algorithme (C) proposé repose sur l'observation du taux d'ascension du plongeur et de l'amplitude de l'accélération dynamique. Si le taux d'ascension est positif, ce qui indique que le plongeur est en train de remonter, et que l'amplitude de l'accélération dynamique chute soudainement, cela peut signaler un arrêt brutal du mouvement du plongeur.

La chute soudaine de l'amplitude de l'accélération peut être due à un arrêt du mouvement du plongeur, ce qui peut signifier qu'il est en état de détresse. C'est pourquoi, dans un tel cas, l'algorithme prédit un état de "Danger", permettant ainsi d'alerter rapidement l'équipe de

surveillance ou les secouristes potentiels.

L'efficacité de cet algorithme dans la détection de ces situations potentiellement dangereuses contribue grandement à la sécurité du plongeur, en offrant une intervention rapide en cas de besoin.

La figure III.3. présente l'organigramme montrant la détection du danger en ascension.

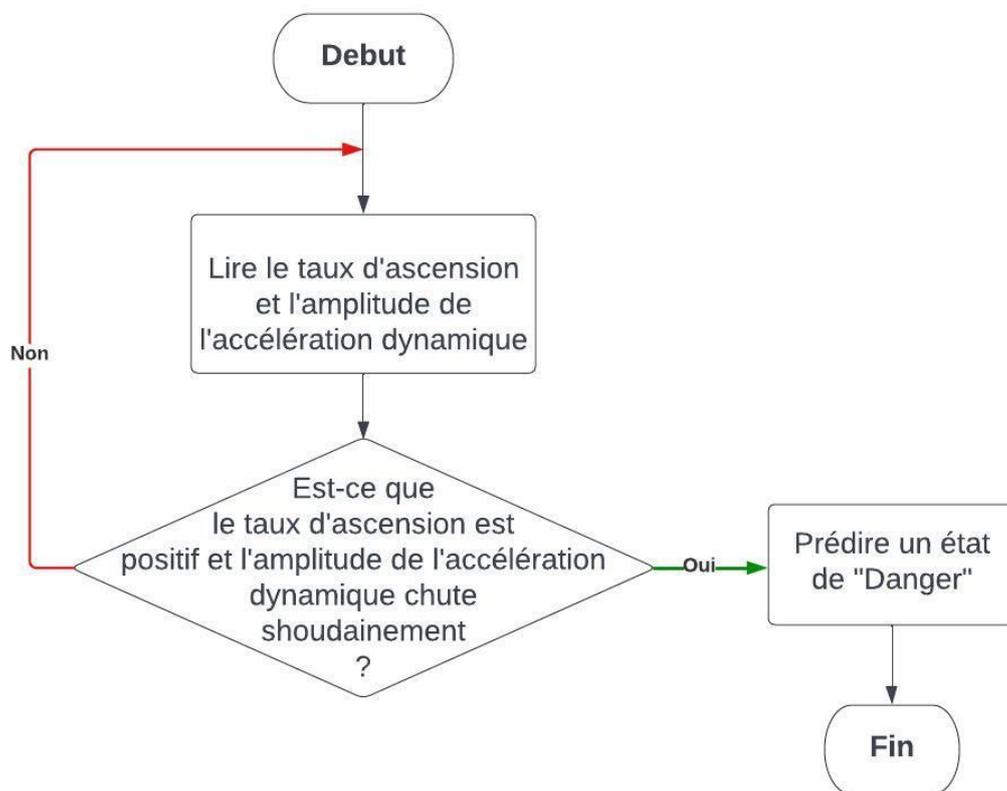


Figure III.3 : Organigramme de la détection du danger en ascension

(C)

Début

Variables : taux_ascension, acceleration, acceleration_précédente : Réel;

etat: Chaîne de caractères;

Tantque Vrai Faire // Boucle infinie pour continuer à surveiller le plongeur

taux_ascension, acceleration = Mesurer_paramètres(); // Prendre des mesures

```
Si ( taux_ascension > 0 ) et ( acceleration < acceleration_précédente ) Alors // Vérifier si
le plongeur monte et si son accélération diminue
    etat = "Danger"; // Déclencher un état de danger
    Afficher (etat); // Afficher le message de danger
FinSi;

    acceleration_précédente = acceleration; // Garder la trace de l'accélération précédente
pour la prochaine itération
FinTantque;
Fin;
```

III.4. Algorithme de régulation de la durée de plongée :

Il s'inscrit dans une approche adaptative de la sécurité du plongeur. Dans le contexte de la plongée, le plongeur peut déterminer un seuil maximal de temps de plongée, par exemple une minute. Cependant, au fil de la session de plongée, et avec la fatigue accumulée, le plongeur peut ne pas être capable de maintenir ce temps maximal à chaque plongée.

Afin de tenir compte de ces variations et d'assurer la sécurité du plongeur, cet algorithme ajuste dynamiquement le seuil maximal de plongée en se basant sur les performances récentes du plongeur. Après chaque dixième plongée, il calcule la durée moyenne des dix dernières plongées (figure III.4). Si cette durée moyenne est significativement inférieure au seuil maximal, il réduit ce seuil pour correspondre plus étroitement à la capacité actuelle du plongeur.

En adaptant le seuil de manière dynamique, cet algorithme assure que le système est toujours aligné avec l'état actuel du plongeur, favorisant ainsi la sécurité (Algorithme D). Par exemple, en réduisant le seuil lorsque la durée moyenne des plongées est nettement inférieure à celui-ci, il peut aider à éviter les situations où le plongeur pourrait être tenté de dépasser ses limites pour atteindre le seuil original. Ainsi, cet algorithme contribue à une plongée plus sûre en adaptant les attentes aux performances actuelles du plongeur.

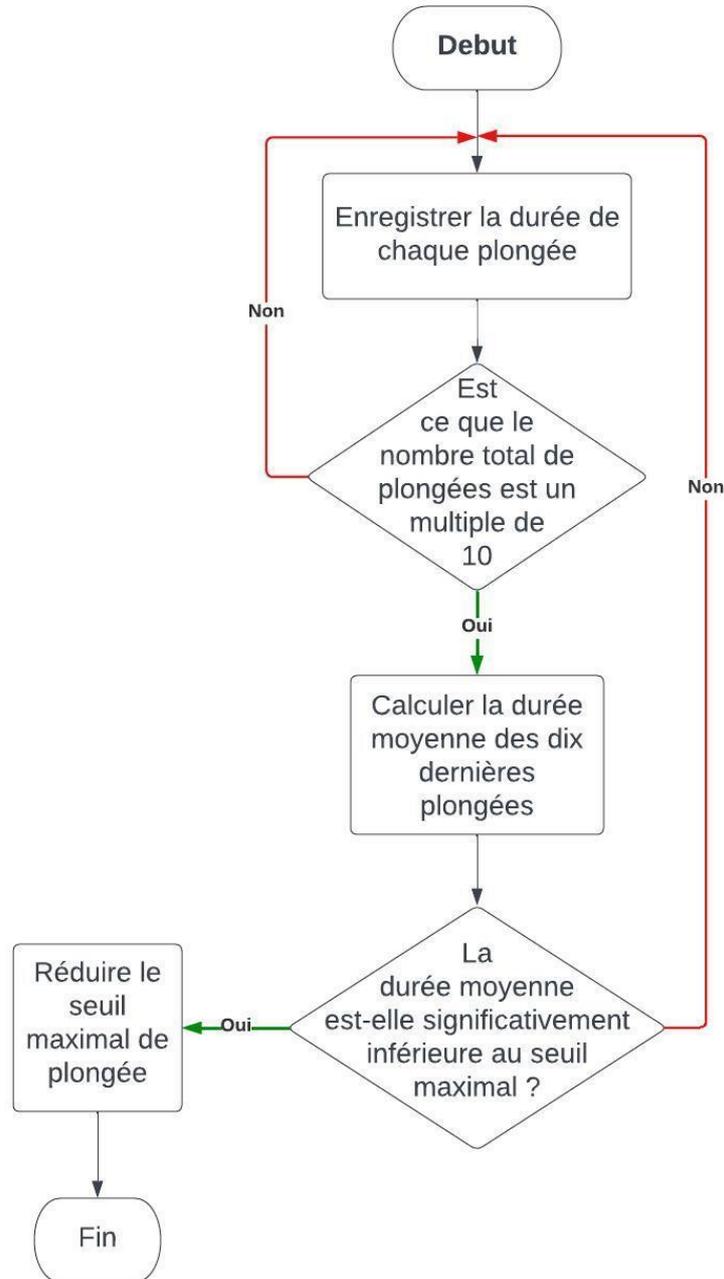


Figure III.4. Organigramme de regulation de la durée de plongée

(D):

Début

Variables : durée_plongée, durée_totale, durée_moyenne, seuil, seuil_préétabli: Réel;
 nb_plongées: Entier;

nb_plongées = 0; // Initialiser le compteur de plongées

```
durée_totale = 0; // Initialiser la durée totale de plongée
seuil = seuil_préétabli; // Définir le seuil initial

Pour chaque plongée Faire
    enregistrer(durée_plongée); // Enregistrer la durée de la plongée
    durée_totale := durée_totale + durée_plongée; // Ajouter la durée de la plongée à la durée
totale
    nb_plongées := nb_plongées + 1; // Incrémenter le compteur de plongées

Si ( nb_plongées mod 10 = 0 ) Alors // Vérifier si 10 plongées ont été effectuées
    durée_moyenne := durée_totale / 10; // Calculer la durée moyenne de plongée

    Si ( durée_moyenne < seuil ) Alors // Si la durée moyenne est inférieure au seuil
        seuil = durée_moyenne + 5; // Augmenter le seuil
    FinSi;

    réinitialiser(durée_totale); // Réinitialiser la durée totale pour les prochaines 10 plongées
FinSi;
FinPour;
Fin;
```

III.5. Algorithme de calibration :

Cet algorithme est dédié au calibrage des valeurs de l'accéléromètre, ce qui est essentiel pour une détection précise des mouvements du plongeur. Les mouvements naturels des vagues à la surface peuvent affecter les valeurs d'accéléromètre et fausser la perception du mouvement réel du plongeur. Pour éviter cela, le calibrage est nécessaire (algorithme E).

La surveillance continue des paramètres du plongeur est le premier pas dans cet algorithme. Cette surveillance constante comprend la profondeur et les valeurs d'accéléromètre du plongeur. La profondeur donne une idée de l'endroit où le plongeur se trouve, tandis que les valeurs de l'accéléromètre reflètent l'intensité du mouvement du plongeur.

Une fois qu'il est confirmé que le plongeur est à la surface, l'algorithme attend que le plongeur commence à descendre. Cette attente assure que le plongeur est stable et que les valeurs d'accéléromètre sont dans leur état le plus naturel.

Au moment de la redescente, l'algorithme récupère les valeurs de l'accéléromètre des 10 dernières secondes précédant la descente. Ces 10 secondes assurent que les valeurs d'accéléromètre sont prises alors que le plongeur est encore relativement immobile, ce qui assure des valeurs de calibrage précises.

Ces valeurs récupérées sont ensuite utilisées comme nouvelles valeurs de calibrage. Ce recalibrage aide à maintenir la précision de la détection du mouvement, même face aux variations naturelles comme les vagues à la surface. Cette réévaluation régulière garantit que le système reste précis et fiable tout au long de la session de plongée comme le montre la figure III.5 suivante.

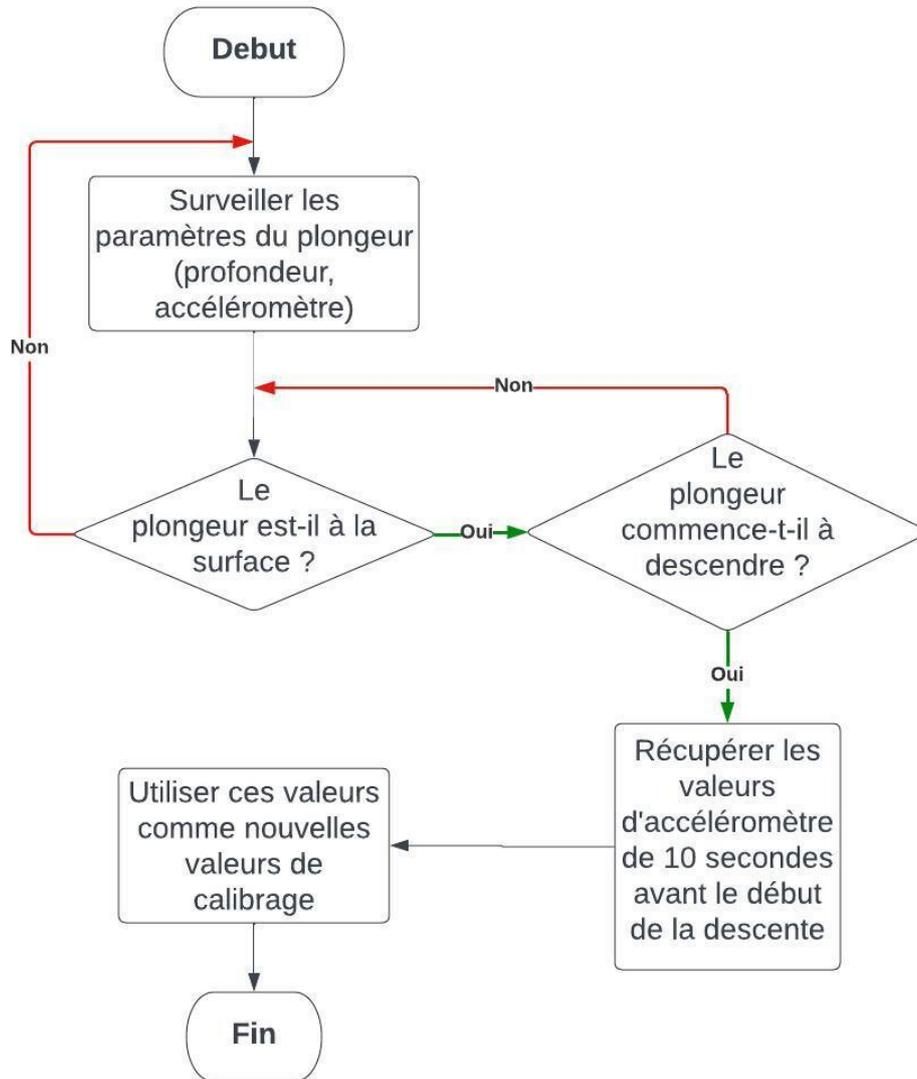


Figure III.5 : Organigramme de calibration.

(E):

Début

Variables : profondeur, valeurs_Accelerometre, valeurs_Calibrage, calibrage, temps, temps_Dernier_Calibrage : Réel;

temps_Dernier_Calibrage = obtenir_Temps_Courant(); // Enregistrer le temps actuel

Tant que Vrai Faire // Boucle infinie pour continuer à surveiller le plongeur

```
temps = obtenir_Temps_Courant(); // Obtenir le temps actuel
surveiller(profondeur, valeurs_Accelerometre); // Surveiller la profondeur et
l'accéléromètre

Si ( profondeur <= surface ) Alors // Si le plongeur est à la surface
    Tant que ( profondeur <= surface ) Faire // Continuer à surveiller tant que le plongeur
est à la surface
        surveiller(profondeur, accelerometre);
    Fin Tantque;

Si ( temps - temps_Dernier_Calibrage >= 30_minutes ) Alors // Si plus de 30 minutes se
sont écoulées depuis le dernier calibrage
    valeurs_Calibrage = obtenir_valeurs(valeurs_Accelerometre,
10_secondes_avant_descente); // Obtenir des valeurs de calibrage
    temps_Dernier_Calibrage = temps; // Enregistrer le temps du dernier calibrage
    FinSi;
    FinSi;
    Fin Tantque;
Fin;
```

Conclusion:

Ce chapitre a exploré en détail la conception et les différents algorithmes que notre système utilise pour détecter les dangers. Il a également exposé comment ces algorithmes sont intégrés dans une solution cohérente.

Chapitre IV :

EXPERIMENTATION ET RESULTATS

Introduction:

Dans ce quatrième chapitre, nous nous concentrerons sur les outils de développement utilisés pour mettre en œuvre notre solution. Ce chapitre abordera également les résultats de l'émulation de quatre scénarios de plongée en apnée, illustrant comment le système se comporte en situation réelle.

IV.1 Matériel utilisé:

Dans cette partie de notre mémoire, nous présenterons les constituants du système de surveillance de la plongée sous marine en apnée.

Pour rappel et d'une manière générale, cet objet intelligent sera équipé de capteur, de processeur et de technologie de communication. Pour ce faire, nous avons utilisé une association de matériel électronique spécialisé que nous décrirons en détail dans ce qui suit.

IV.1.1. Accéléromètre MPU 6050:

Un capteur doit être capable de détecter les changements dans l'environnement du plongeur. De ce fait, notre capteur "Accéléromètre MPU6050" est au cœur de notre système de détection de mouvement. Il intègre à la fois un gyroscope et un accéléromètre sur trois axes (x, y, z). Ceci nous permet de surveiller les mouvements du plongeur dans toutes les directions avec une grande précision.

Cette capacité à suivre le mouvement est très importante pour détecter tout comportement suspect qui pourrait indiquer un danger, comme une immobilité prolongée dans le temps. De plus, le MPU6050 a une faible consommation d'énergie, ce qui est essentiel pour notre application où l'autonomie de la batterie est vivement recommandée [64]. La figure IV.1, ci dessous montre l'image d'un Accéléromètre MPU6050

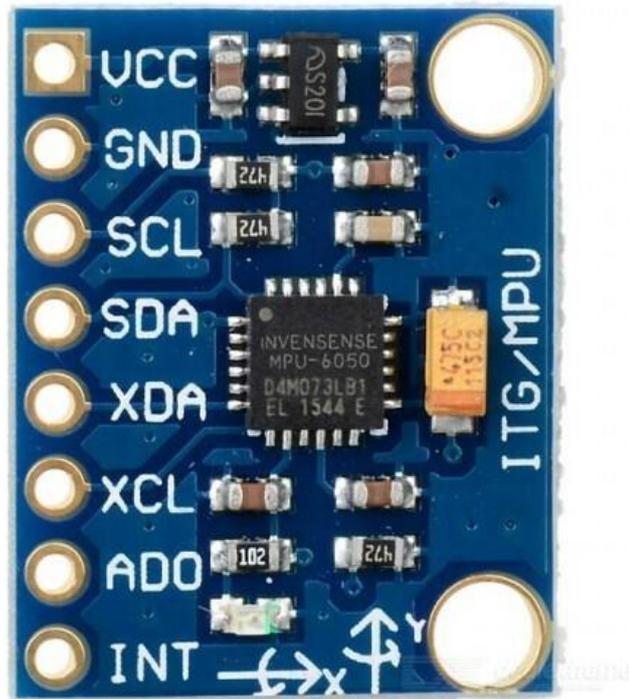


Figure IV.1. Image d'un accéléromètre MPU6050

IV.1.2. Arduino UNO et Arduino NANO:

C'est une plateforme open-source d'électronique programmée qui est basée sur une simple carte à microcontrôleur, et un logiciel, incontestable environnement de développement intégré, pour écrire, compiler et transférer le programme vers la carte à microcontrôleur [65].

L'objet intelligent doit être capable de traiter les données qu'il reçoit. Grace à des microcontrôleurs. Pour cette raison, les microcontrôleurs Arduino sont la colonne vertébrale de notre système.

L'Arduino UNO, avec ses 14 broches d'E/S numériques (dont 6 peuvent être utilisées comme sorties PWM), 6 entrées analogiques, un oscillateur à quartz 16 MHz, une connexion USB, une prise d'alimentation et un bouton de réinitialisation, sert de plateforme principale pour le traitement des données et l'exécution de nos algorithmes de détection. D'autre part, l'Arduino NANO, est une carte microcontrôleur prête à l'emploi et entièrement programmable avec nous pouvons piloter et créer tout ce que nous souhaitons.

L'arduino nano offre des fonctionnalités similaires à l'arduino uno mais dans un format beaucoup plus compact. Il est ainsi idéal pour les parties du système qui doivent être portables et économiques en énergie [66]. La figure IV.2 montre l'image des deux types d'arduino.

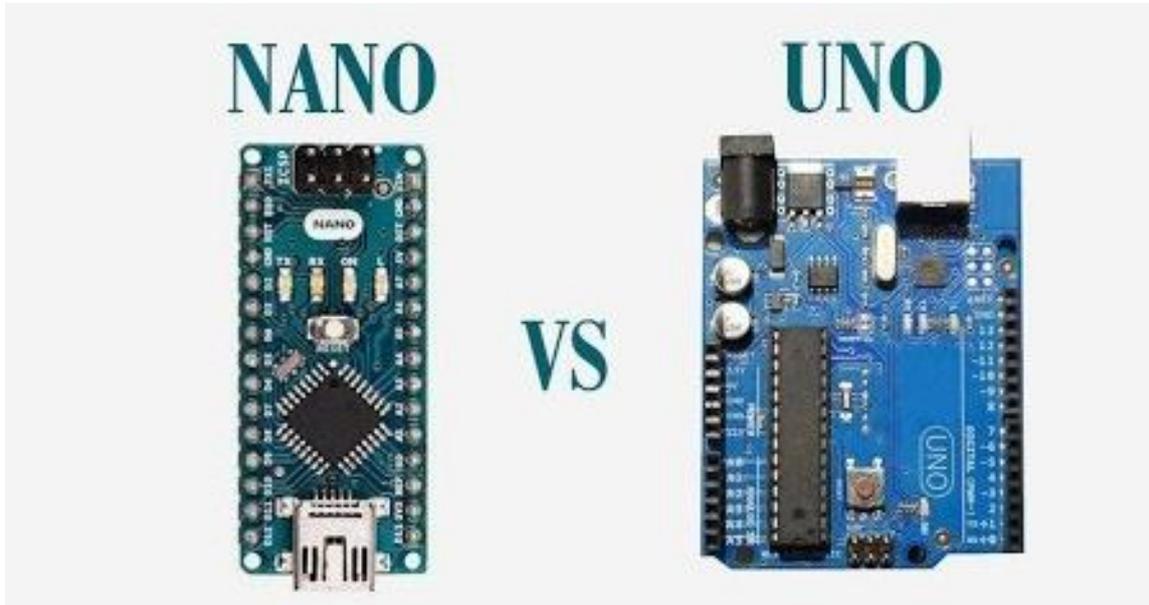


Figure IV.2. Photographie des Arduino nano et uno

IV.1.3. NRF24L01 Wireless Transmitter:

Dans ce travail, nous avons opté pour le NRF24L01 Wireless Transmitter comme outil de communication comme le montre la figure IV.3 ci après.

Ce module d'émission-réception sans fil permet une communication bidirectionnelle entre nos dispositifs. Il opère à une fréquence de 2,4 GHz. Comme ce système a une faible consommation d'énergie, il va permettre de prolonger la durée de vie de la batterie de notre système, et il est capable de transmettre des données à un débit allant jusqu'à 2 Mbps, ce qui est suffisant pour transmettre nos données de capteur [67].

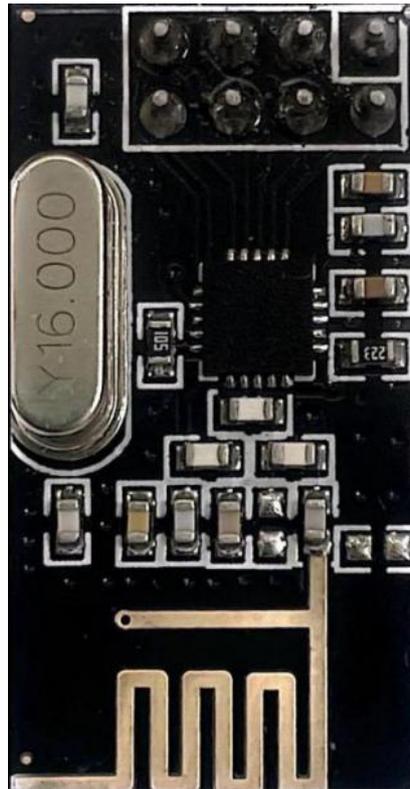


Figure IV.3. Image du NRF24L01 Wireless Transmitter

IV.1.4. Capteur de pression d'eau :

Dans une application réelle de notre algorithme, un capteur de pression d'eau serait un élément essentiel du système. Bien que nous n'ayons pas utilisé ce capteur dans notre simulation, nous reconnaissons son importance et l'incluons donc dans la description du matériel nécessaire.

Un capteur de pression d'eau (figure IV.4) est un appareil qui peut mesurer avec précision la pression d'un fluide, en l'occurrence l'eau. Dans le cadre de notre projet, il serait utilisé pour surveiller en continu la profondeur du plongeur.

Le capteur de pression d'eau fonctionne sur une idée simple : plus la profondeur est grande, plus la pression de l'eau est forte. En mesurant cette pression, le capteur peut donc nous dire à quelle profondeur se trouve le plongeur à tout moment.

Ces informations de profondeur sont essentielles pour notre algorithme. Elles permettent de déterminer si le plongeur est à la surface ou sous l'eau, ce qui est une composante clé de

notre système de détection des conditions de plongée et de prédiction des situations potentiellement dangereuses [68].



Figure IV.4. Image d'un capteur de pression SEN0257

IV.2. Présentation des résultats et émulation:

Le langage **PYTHON** a été utilisé pour la programmation du machine learning. Il offre de nombreuses bibliothèques comme **scikit-learn** pour l'entraînement et **MicroMLGen** pour la traduction du code Python en C++ optimisé (ANNEXE A).

De plus, L'IDE Arduino (environnement de développement intégré) qui est une plateforme logicielle a été utilisée pour la programmation et le développement des microcontrôleurs Arduino (ANNEXE B)

IV.2.1. Mise en oeuvre de l'objet connecté:

Les systèmes Emetteurs et Receveurs mis en oeuvre par nos soins pour la concrétisation de notre objet intelligent, sont présentés sur les figures IV.5 et figure IV.6 respectivement.

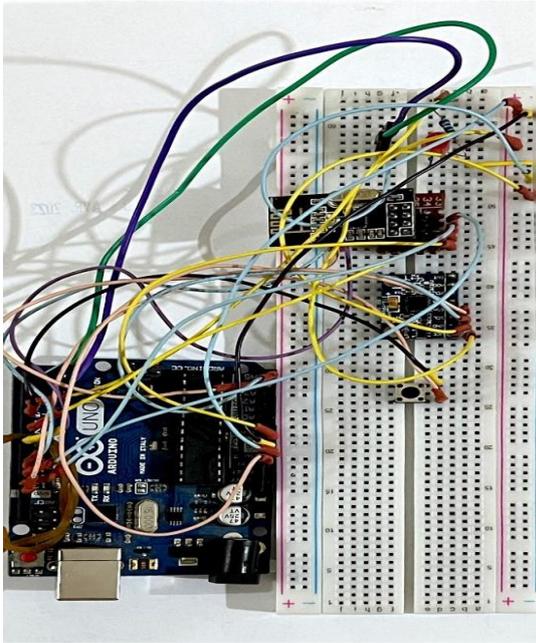


Figure IV.5: Système Emetteur de l'objet connecté

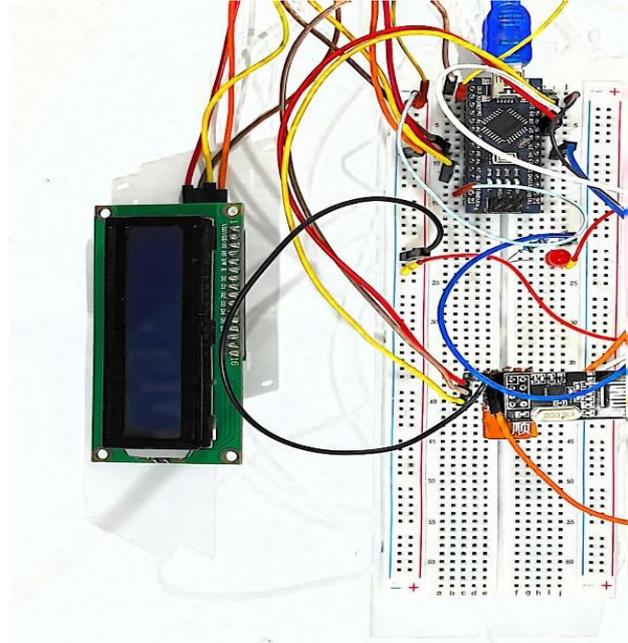


Figure IV.6: Système Receveur de l'objet connecté

A ce sujet, l'émetteur comprend une carte Arduino Uno, un accéléromètre de type MPU6050, une lampe LED, un bouton, un émetteur sans fil de type nRF24L01 et un ensemble de fils pour relier ces composants. L'accéléromètre lit les données qui sont ensuite analysées à l'aide d'un modèle d'apprentissage automatique sur Arduino. S'il détecte un danger, la LED s'allume et l'information est envoyée au récepteur pour indiquer que la situation est dangereuse. En cas de fausse détection de danger, nous appuyons sur le bouton pour empêcher de signaler le danger au récepteur. Nous ajoutons ensuite un laps de temps et effectuons une nouvelle prédiction.

Quant à la seconde partie de l'objet intelligent élaboré, elle est composée d'un Arduino Nano et d'un autre module sans fil qui joue le rôle de récepteur pour recevoir les données de l'autre carte. Si le récepteur détecte un danger, il allume la LED et affiche des messages sur l'écran LCD.

IV.2.2. Résultats et intégration du modèle Machine Learning sur Arduino :

Dans cette section, nous présenterons le modèle Machine Learning que nous avons développé ainsi que son intégration sur un microcontrôleur Arduino pour une inférence en temps réel dans les systèmes embarqués. Notre objectif principal est d'explorer les performances du modèle et de démontrer sa faisabilité dans un environnement embarqué.

IV.2.2.1. Résultats et évaluation du modèle Machine Learning

Le modèle Machine Learning a obtenu une précision de 0,983, ce qui indique sa capacité à classifier correctement les instances. La matrice de confusion fournit des informations sur les performances du modèle. Elle montre que sur les 270 instances étiquetées comme "safe" (sûr), 260 ont été correctement classées, tandis que 10 ont été classées à tort comme "danger" (dangereux). De plus, les 333 instances étiquetées comme "danger" ont toutes été correctement classées.

Grâce au processus d'optimisation, les meilleurs hyperparamètres du modèle ont été déterminés : max depth=5, max features='sqrt' et number of estimators=100. Ces paramètres ont été choisis pour optimiser les performances du modèle en fonction de l'ensemble de données utilisé.

IV.2.2.2. Intégration avec Arduino en utilisant micromlgen

Le modèle Machine Learning développé avec scikit-learn a été converti en code C++ optimisé à l'aide de la bibliothèque micromlgen. Le code du modèle converti a été placé dans un fichier de bibliothèque appelé "model.h". Cette bibliothèque encapsule les fonctions nécessaires pour effectuer des prédictions basées sur le modèle Machine Learning.

La bibliothèque "model.h" a été incluse dans le programme principal d'Arduino. La fonction de prédiction du modèle a ensuite été appelée dans le programme pour exploiter les capacités du modèle Machine Learning. La bibliothèque eloquent, basée sur TensorFlow Lite, a été utilisée pour gérer les appels de fonction et les interactions avec le modèle Machine Learning dans l'environnement Arduino.

IV.2.2.3. Implémentation et utilisation

Dans le code Arduino, la fonction de prédiction du modèle Machine Learning a été invoquée avec les paramètres pertinents (dynamicAccelMagnitude, depth, ascentRate) pour obtenir une prédiction. Le résultat de la prédiction, stocké dans la variable "predictedLabel", représente la classification du modèle pour l'entrée donnée en tant que "safe" (sûr) ou "danger" (dangereux).

```
// Effectuer une prédiction en utilisant le modèle Machine Learning  
float features[] = {dynamicAccelMagnitude, depth, ascentRate};
```

```
int prediction = rf.predict(features);  
String predictedLabel = rf.idxToLabel(prediction);
```

IV.2.3. Emulation:

Avant d'aborder les résultats de l'émulation des scénarios, il convient de présenter le profil du plongeur sur lequel les simulations ont été basées. Il est présenté ci dessous:

Sexe:	Masculin
Âge:	34 Ans
Poids:	83 kg
Durée max de plongée:	1 min 20 sec
Profondeur max:	18 mètres
Durée max d'arrêt en profondeur:	30 secondes

Nous présenterons dans ce qui suit, 4 scénarios que nous avons jugé probables lors d'une plongée sous marine en apnée. Néanmoins, il est impératif de définir au préalable, les facteurs utiles en plongée sous-marine. A cet effet, nous avons suivi deux paramètres essentiels en fonction du temps de plongée à savoir, la profondeur et l'accélération du mouvement du plongeur.

Sachant que la vitesse moyenne de nage d'un plongeur expérimenté varie entre 2 et 4 km/h lorsqu'il n'est pas en face de courants forts [69], nous avons opté pour une accélération qui oscille entre -1.8 à $+1.8$ m/s². Quant à la profondeur, elle sera limitée à 18 mètres. Les résultats des 4 scénarios sont résumés sur les tableaux I, II, III et IV présentés en Annexe C.

IV.2.3.1 Premier Scénario: Plongée régulière sans danger

Après un repos de 5 minutes, le plongeur effectue une plongée d'une durée totale de 1 minute. Il s'arrête à une profondeur de 15 mètres pendant 20 secondes.

Dans ce scénario présenté sur la figure IV.7, aucun danger n'est détecté par notre système car il n'a pas dépassé les 30 secondes programmées en profondeur et par conséquent, aucun signal n'est émis.

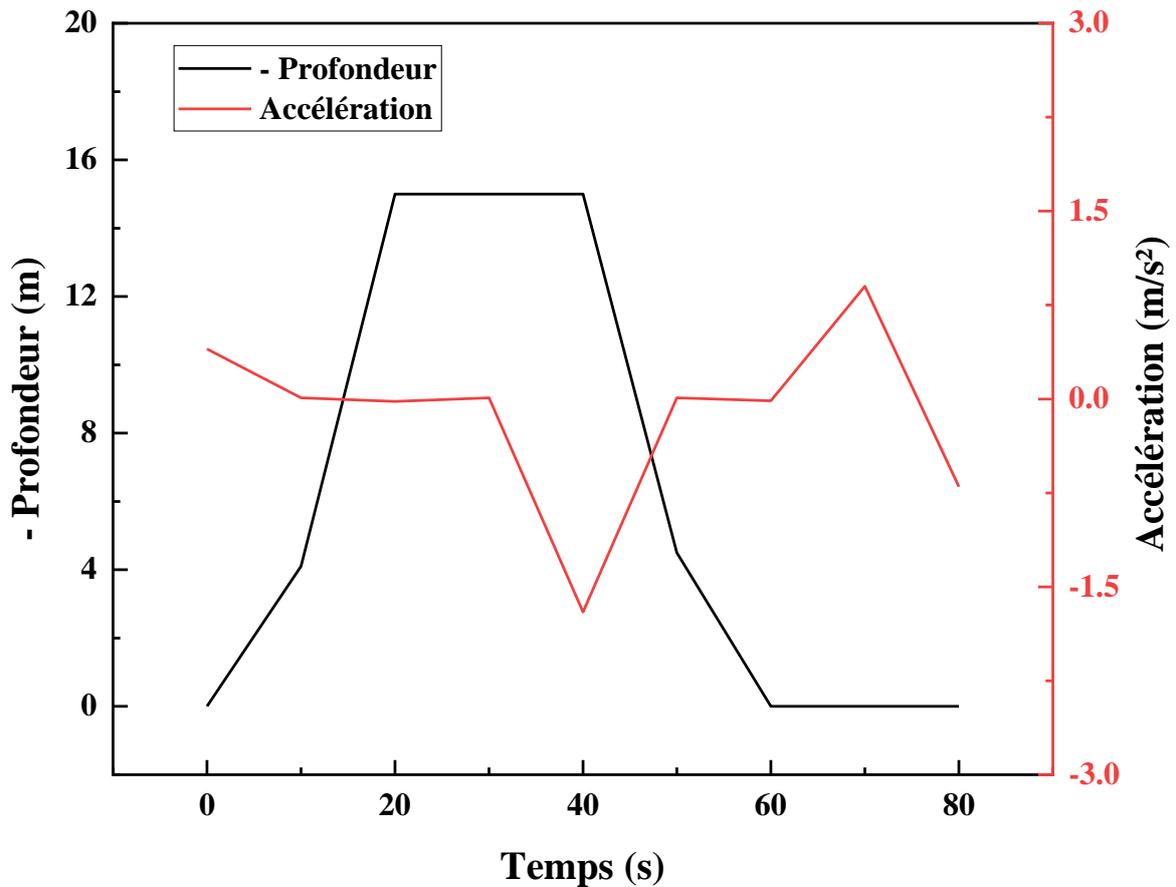


Figure IV.7. Scénario 1 de plongée en apnée

IV.2.3.2 Deuxième Scénario: Danger dû à un arrêt prolongé en profondeur

Suite à un repos de 6 minutes, le plongeur commence une plongée et s'arrête à une profondeur de 16 mètres pendant 40 secondes, dépassant ainsi le seuil de sécurité.

La figure IV.8 montre bien la quasi absence de mouvement du nageur en profondeur et qui persiste au delà de la 60^{ème} seconde. Le système détecte ce comportement comme un danger potentiel après dépassement des 30 seconde (A) et emet un signal (B) dans les 5 secondes qui suivent. L'intervention est alors imminente et le nageur peut être remonté en surface pour bénéficier des premiers secours.

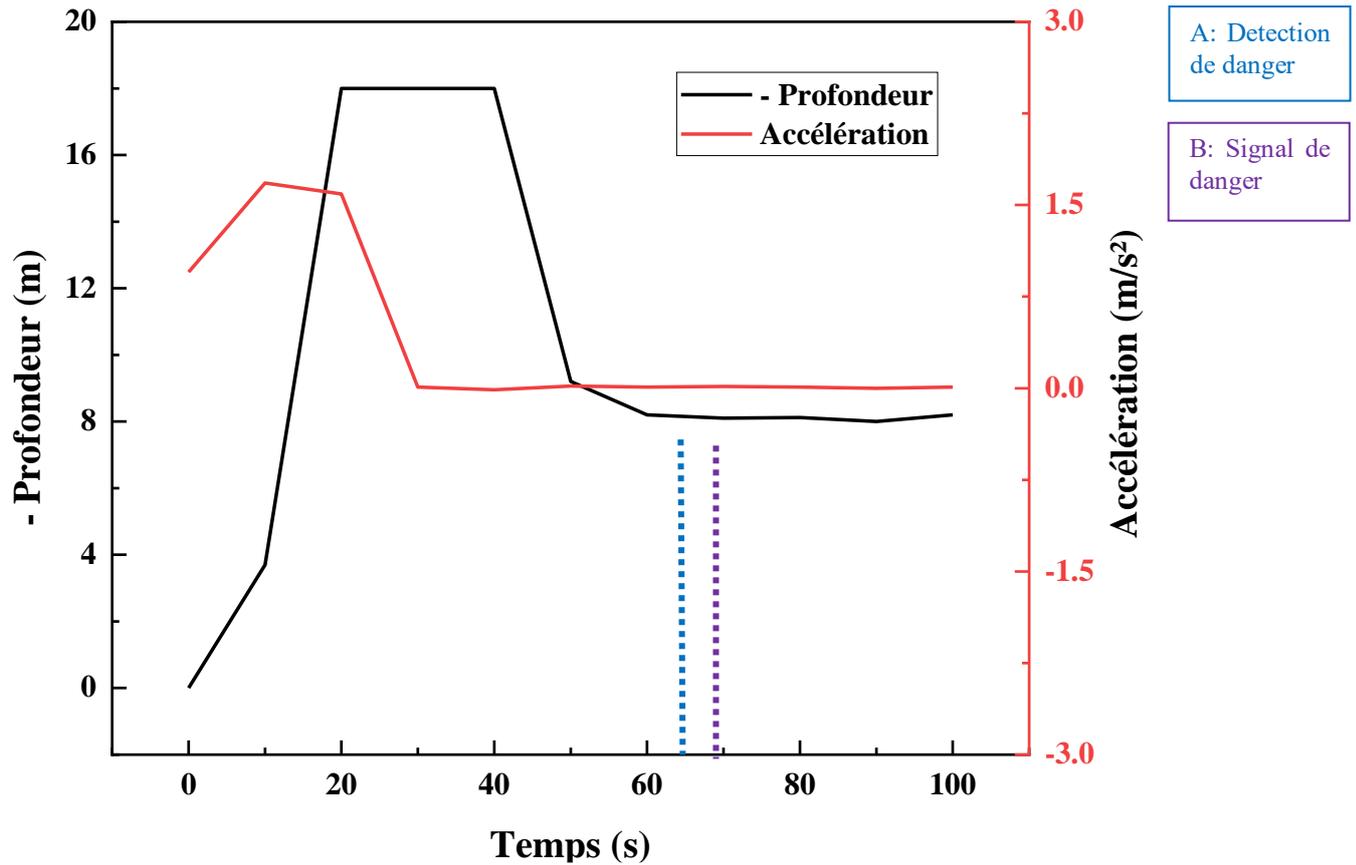


Figure IV.8. Scénario 2 de plongée en apnée

IV.2.3.3 Troisième Scénario: Danger potentiel lors de la remontée

Après un repos de 7 minutes, le plongeur commence sa plongée. Il atteint sa profondeur maximale de 18 mètres, où il reste immobile pendant 25 secondes. Lors de la remontée, l'accélération devient soudainement stable (figure IV.9).

L'immobilité du plongeur, liée soit à une hypoxie soit à une décompression soudaine est un signe de danger détecté par notre système qui émet un signal dans les 5 secondes suivantes. La réception du signal sera alors salutaire au plongeur en difficulté.

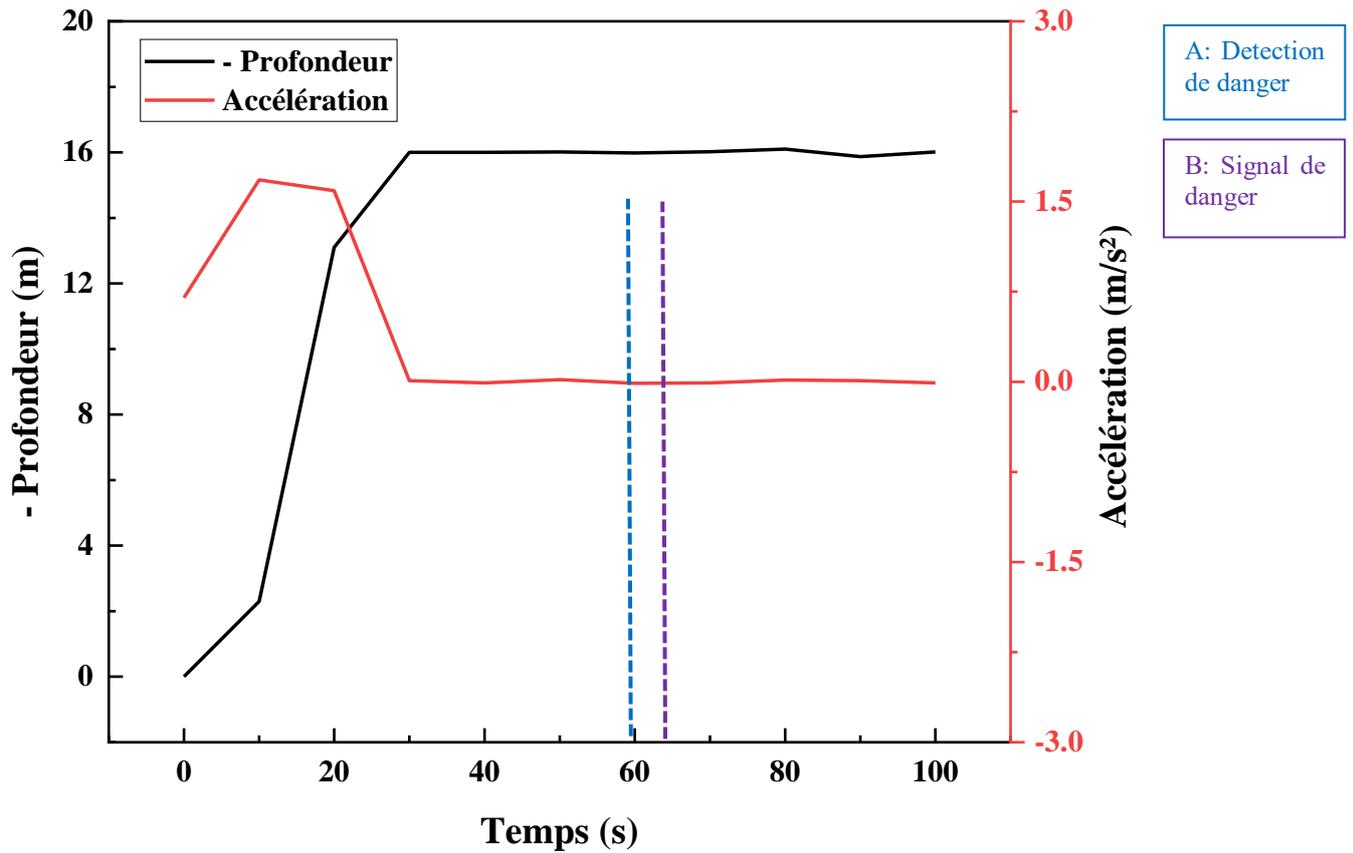


Figure IV.9. Scénario 3 de plongée en apnée

IV.2.3.4 Quatrième Scénario: Danger à la surface

Suite à un repos de 8 minutes, le plongeur commence une plongée, atteignant une profondeur de 18 mètres et restant immobile pendant 25 secondes comme le montre la figure IV.10. Cependant, une fois à la surface, le plongeur cesse de bouger, ce qui est également détecté comme un danger par notre système sachant que la syncope peut survenir même en surface lorsque les réserves en oxygène sont complètement épuisées.

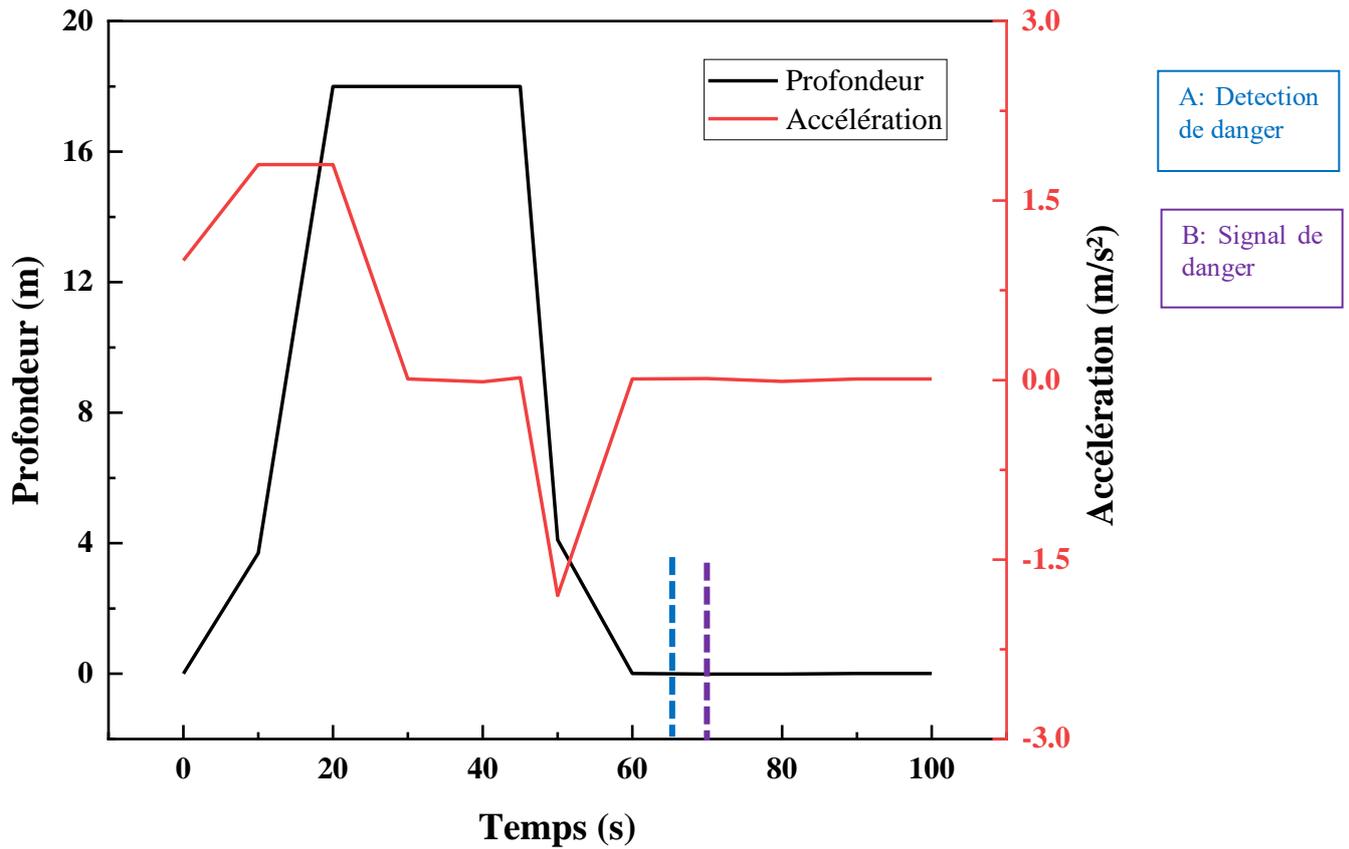


Figure IV.10. Scénario 4 de plongée en apnée.

Conclusion:

Ce dernier chapitre a permis de mettre en lumière les résultats de l'expérimentation de notre système de détection de danger en apnée.

Les différents scénarios ont démontré l'efficacité de notre système dans la détection de divers types de dangers. Les outils et techniques présentés tout au long de ces chapitres serviront de base pour les futures améliorations et adaptations du système.

CONCLUSION GENERALE

Conclusion Générale:

Ce travail représente une contribution dans la prévention des accidents en plongée sous-marine en apnée.

L'objectif majeur de notre thème est la conception et le développement d'un objet connecté novateur permettant l'intervention de sauvetage en cas d'accident, sachant qu'à ce jour, aucun travail de recherche n'a été réalisé sur les outils capables de détecter les accidents de plongée sous-marine en apnée.

Des concepts fondamentaux, tels que l'IoT et l'e-Santé ont été présentés montrant ainsi que les objets connectés dotés de machine Learning sont des outils puissants qui ouvrent de nouvelles voies dans les soins de santé et de prévention des risques d'accidents.

La comparaison des différentes techniques de Machine Learning, a mis en évidence que l'algorithme Random Forest offre les meilleures performances pour notre système de détection de danger en plongée sous-marine en apnée.

Pour assurer la sécurité des plongeurs et prévenir les situations de danger, une surveillance en temps réel des paramètres de la plongée, a nécessité la construction d'algorithmes spécifiques. A cet effet, différents algorithmes ont été proposés pour la détection de danger en profondeur et en ascension, la régulation de la durée de plongée, ainsi que la calibration qui garantit la précision et la fiabilité du système tout au long de la session de plongée.

Des outils de développement ont été mis en œuvre. Par ailleurs, quatre scénarios probables en plongée sous-marine en apnée ont été proposés et ont démontré l'efficacité de notre système dans la détection de divers types de dangers. De ce fait, le système proposé servira de base pour les futures améliorations et adaptations.

Bien que des paramètres comme le mouvement, la profondeur, le temps et la pression ont été identifiés, nous avons été confrontés à des difficultés pour mesurer la pression en raison de la complexité de sa mise en œuvre en milieu sous-marin.

En perspectives d'amélioration et de précision de notre objet connecté, il est utile d'intégrer de nouveaux paramètres essentiels pour une évaluation plus complète de la plongée sous-marine en apnée.

Deux mesures clés seraient la fréquence cardiaque du plongeur et le taux d'oxygène dans son sang. La fréquence cardiaque fournirait des informations sur l'effort physique et le niveau de stress du plongeur permettant ainsi de détecter des situations potentiellement dangereuses. Quant à la mesure du taux d'oxygène dans le sang, elle est cruciale pour surveiller la saturation en oxygène du plongeur et identifier tout risque d'hypoxie.

Références bibliographiques :

- [1]: Corriol J.H. La plongée en apnée. Physiologie et médecine. Paris: Masson, 2002.
- [2]: Sébastien Celle, Vincent Pichot, (2009) Undiagnosed sleep-related breathing disorders are associated with focal brainstem atrophy in the elderly Human Brain Mapping 30: 2090–2097
- [3]: Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of Things: A survey on enabling technologies, protocols, and applications. IEEE Communications Surveys & Tutorials, 17(4), 2347-2376.
- [4]: Ashton, K. (2009). That 'Internet of Things' Thing. RFID Journal, 22(7), 97-114.
- [5]: Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. IEEE Communications Surveys & Tutorials, 17(4), 2347-2376.
- [6]: Miorandi, D., Sicari, S., De Pellegrini, F., & Chlamtac, I. (2012). Internet of Things: Vision, Applications and Research Challenges. Ad Hoc Networks, 10(7), 1497-1516.
- [7]: Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. Future Generation Computer Systems, 29(7), 1645-1660.
- [8]: Islam, S. M. R., Kwak, D., Kabir, M. H., Hossain, M., & Kwak, K. S. (2015). The Internet of Things for health care: a comprehensive survey. IEEE Access, 3, 678-708.
- [9]: Ray, P. P. (2017). Internet of Things for smart agriculture: Technologies, practices and future direction. Journal of Ambient Intelligence and Smart Environments, 9(4), 395-420.
- [10]: Atzori, L., Iera, A., & Morabito, G. (2010). The Internet of Things: A survey. Computer Networks, 54(15), 2787-2805.
- [11]: https://wikimemoires.net/2019/09/domaines-d-applications-de-l'iot/#4_Domaines-d%E2%80%99applications_de_1%E2%80%99IoT
- [12]: Mekki, K., Bajic, E., Chaxel, F., & Meyer, F. (2018). A comparative study of LPWAN technologies for large-scale IoT deployment. ICT Express, 5(1), 1-7.
- [13]: Victor A.M. Jorge, Alaxandre M. Amori, (2019) A Survey on Unmanned Surface Vehicles for Disaster Robotics: Main Challenges and Directions MPDI, 19 (3), 1-44

- [14]: Varghese, A., & Tamma, B. R. (2017). Wireless sensor networks: a survey on recent developments and potential synergies. *The Journal of Supercomputing*, 73(4), 1597-1628.
- [15]: Roman, R., Zhou, J., & Lopez, J. (2013). On the features and challenges of security and privacy in distributed internet of things. *Computer Networks*, 57(10), 2266-2279.
- [16]: Granjal, J., Monteiro, E., & Silva, J. S. (2015). Security for the Internet of Things: A Survey of Existing Protocols and Open Research Issues. *IEEE Communications Surveys & Tutorials*, 17(3), 1294-1312.
- [17]: Weber, R. H. (2018). Internet of Things – Governance quo vadis? *Computer Law & Security Review*, 34(4), 669-678.
- [18]: Bandyopadhyay, D., & Sen, J. (2011). Internet of Things: Applications and Challenges in Technology and Standardization. *Wireless Personal Communications*, 58(1), 49-69.
- [19]: Ray, P. P. (2016). A survey on Internet of Things architectures. *Journal of King Saud University - Computer and Information Sciences*, 30(3), 291-319.
- [20]: Alrawais, A., Alhothaily, A., Hu, C., & Cheng, X. (2017). Fog Computing for the Internet of Things: Security and Privacy Issues. *IEEE Internet Computing*, 21(2), 34-42.
- [21]: Vincenzo Della Mea. (2001) What is e-health (2): The death of telemedicine, *Journal of Medical Internet Research*, 3(2), 1-2
- [22]: Rizoulières, R. (2016) Confiance dans le numérique en santé. Quelles priorité pour l'action, *I2D-Information, Données et Documents vol 53 (3)*, 59-60.
- [23]: Eysenbach, G. (2001). What is e-health? *Journal of Medical Internet Research*, 3(2), e20.
- [24]: Oh, H., Rizo, C., Enkin, M., & Jadad, A. (2005). What is eHealth (3): a systematic review of published definitions. *Journal of Medical Internet Research*, 7(1), e1.
- [25] <https://www.journalducim.com/sante-connectee-e-sante/>
- [26]: Bashshur, R., & Shannon, G. W. (2009). History of Telemedicine: Evolution, Context, and Transformation. *Telemedicine and e-Health*, 15(1), 91-93.
- [27]: Wootton, R. (2012). Twenty years of telemedicine in chronic disease management – an evidence synthesis. *Journal of Telemedicine and Telecare*, 18(4), 211-220.
- [28]: Piwek, L., Ellis, D. A., Andrews, S., & Joinson, A. (2016). The rise of consumer health wearables: promises and barriers. *PLoS Medicine*, 13(2), e1001953.
- [29]: Terrance J. Dishongh and Michael McGrath. Wireless sensor networks for health-care

application. Artech House, 1 edition, 2009.

[30]: Swan, M. (2012). Sensor Mania! The Internet of Things, Wearable Computing, Objective Metrics, and the Quantified Self 2.0. *Journal of Sensor and Actuator Networks*, 1(3), 217-253.

[31]: Jiang, F., Jiang, Y., Zhi, H., Dong, Y., Li, H., Ma, S., ... & Wang, Y. (2017). Artificial intelligence in healthcare: past, present and future. *Stroke and Vascular Neurology*, 2(4), 230-243.

[32]: Pendergast, D. R., et al. (2015). The dangers of breath-hold diving. In: Lundgren CEG, Miller JN, editors. *The Lung at Depth*. Springer New York, 57-72.

[33]: Brubakk, A. O., & Neuman, T. S. (2003). Decompression sickness. In: Brubakk AO, Neuman TS, editors. *Bennett and Elliott's Physiology and Medicine of Diving*. Saunders, 467-517.

[34]: Edmonds, C., et al. (2012). Barotrauma. In: Edmonds C, et al., editors. *Diving and Subaquatic Medicine*. CRC Press, 109-146.

[35]: E. Alpaydin, "Introduction to machine learning," MIT press, 2020.

[36]: S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd edition. Prentice Hall, 2009.

[37]: Y. LeCun, Y. Bengio and G. Hinton, (2015) "Deep learning," *Nature*, vol. 521, 436-444.

[38]: R. Baeza-Yates, B. Ribeiro-Neto, et al. *Modern information retrieval*. Vol. 463. ACM press New York, 1999.

[39]: D. Hand, H. Mannila, and P. Smyth, *Principles of data mining*. MIT press, 2001.

[40]: T. Mitchell, "Machine Learning," McGraw Hill, 1997.

[41]: <https://towardsdatascience.com/what-are-the-types-of-machine-learning-e2b9e5d1756f>

[42]: Murphy, K. P. *Machine learning: a probabilistic perspective*. MIT press. 2012

[43]: James, G., Witten, D., Hastie, T., & Tibshirani, R. *An introduction to statistical learning* (Vol. 112, p. 18). New York: springer 2013.

[44] <https://slideplayer.fr/slide/491441/>

[45]: P. Domingos, (2012) "A Few Useful Things to Know about Machine Learning," in *Communications of the ACM*, vol. 55, no. 10, pp. 78-87.

[46]: Vapnik, V. N., & Vapnik, V. (1998). *Statistical learning theory* (Vol. 1). Wiley New York.

[47]: L. Breiman, J. Friedman, R. Olshen, and C. Stone, "Classification and Regression Trees".

Wadsworth, Belmont, CA, 1984.

[48] <https://blent.ai/blog/a/arbres-de-decision-en-machine-learning>.

[49]: L. Breiman, (2001) "Random Forests", *Machine Learning*, 45(1), 5-32.

[50] https://www.researchgate.net/figure/Construction-dune-foret-aleatoire_fig7_281184702

[51]: MacQueen, J. (1967) Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability* Vol. 1(14) 281-297.

[52]: Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996, August). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd* Vol. 96,(34) 226-231.

[53]: Johnson, S. C. (1967). Hierarchical clustering schemes. *Psychometrika*, 32(3), 241-254.

[54]: P. Domingos, (2012) "A Few Useful Things to Know About Machine Learning," *Communications of the ACM*, vol. 55, (10),78-87.

[55]: T. G. Dietterich, (2000) "Ensemble Methods in Machine Learning," in *Multiple Classifier Systems*, Springer,1-15.

[56]: J. H. Friedman, (2001)"Greedy Function Approximation: A Gradient Boosting Machine," *Annals of Statistics*, vol. 29, no. 5, pp. 1189-1232.

[57]: Richard S. Sutton and Andrew G. Barto, *Reinforcement Learning: An Introduction*, 2nd edition, MIT Press, 2018.

[58]: Michael L. Littman, (1994)"Markov games as a framework for multi-agent reinforcement learning," in *Proceedings of the Eleventh International Conference on Machine Learning*, 157-163.

[59]: Richard, S. Sutton et Andrew G. Barto, *Reinforcement Learning: An Introduction*, Ed. The MIT Press 2018.

[60]: Leo Breiman, (2001)"Random Forests," *Machine Learning*, vol. 45, (1), 5-32.

[61]: Ho, Tin Kam. Random decision forests. *IEEE Proceedings of 3rd International Conference on Document Analysis and Recognition*, 1995.

[62]: Robin Genuer, Jean-Michel Poggi. (2017) Arbres CART et Forêts aléatoires, Importance et sélection de variables.. fhal-01387654v2

[63]: Simon Bernard. Forêts Aléatoires: De l'Analyse des Mécanismes de Fonctionnement à la Construction Dynamique. Apprentissage Thèse de Doctorat, Université de Rouen, 2009.

- [64]: <https://invensense.tdk.com/products/motion-tracking/6-axis/mpu-6050/>, "MPU-6050 Six-Axis (Gyro + Accelerometer) MEMS MotionTracking™ Devices," InvenSense Inc.
- [65]: Margolis, M. (2011) Arduino Cookbook, O'Reilly Media, Inc.
- [66]: <https://www.arduino.cc>, "Arduino UNO & Genuino UNO, Arduino NANO," Arduino.
- [67]: <https://www.nordicsemi.com>, "nRF24L01+ Single Chip 2.4GHz Transceiver Product Specification," Nordic Semiconductor.
- [68]: <https://www.gotronic.fr/art-capteur-de-pression-sen0257-27841.htm>, "Capteur de pression SEN0257," GoTronic.
- [69]: Umberto Pelizzari, Stefano Tovagliari, "Manual of Freediving: Underwater on a Single Breath, Ed. Idelson Gnocci. 2004

ANNEXE A

1- Python :

Python est un langage de programmation interprété de haut niveau reconnu pour sa simplicité et sa lisibilité. Il offre une grande variété de fonctionnalités et dispose d'un vaste écosystème de bibliothèques et de Framework, ce qui en fait un choix populaire pour diverses applications, notamment l'analyse de données, le développement web, l'apprentissage automatique et le calcul scientifique. La syntaxe de Python est concise et expressive, ce qui facilite l'écriture et la compréhension du code. Sa polyvalence et le soutien de la communauté ont favorisé son adoption généralisée tant dans le milieu académique qu'industriel.



REF: Van Rossum, G., & Drake, F. L. (2009). Python 3 Reference Manual. CreateSpace.

2- scikit-learn:

scikit-learn est une bibliothèque Machine Learning open-source pour Python qui propose une large gamme d'algorithmes et d'outils pour des tâches telles que la classification, la régression, le regroupement et la réduction de dimension. Elle offre une interface simple et efficace, ce qui la rend populaire pour la mise en œuvre de modèles Machine Learning.



REF: Pedregosa, F., et al. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825-2830.

3- MicroMLGen:

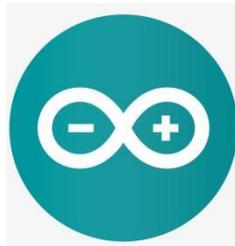
micromlgen est une bibliothèque Python qui convertit des modèles Machine Learning entraînés en code C++ optimisé pour des microcontrôleurs tels que l'Arduino Uno. Elle se spécialise dans la génération de code compact et efficace pouvant s'exécuter sur des appareils avec des ressources informatiques limitées. Elle permet le déploiement de modèles Machine Learning sur des microcontrôleurs pour l'inférence en temps réel dans les systèmes embarqués.

REF: Varat, S., et al. (2021). micromlgen: A Python library for converting machine learning models to C code for microcontrollers. *Journal of Open Source Software*, 6(59), 3074.

ANNEXE B

Arduino IDE:

L'IDE Arduino (environnement de développement intégré) est une plateforme logicielle utilisée pour la programmation et le développement des microcontrôleurs Arduino. Elle offre une interface conviviale, un langage de programmation simplifié et un ensemble de bibliothèques qui permettent la création de projets électroniques interactifs et le prototypage.



REF: Arduino. (n.d.). Arduino IDE. Récupéré sur <https://www.arduino.cc/en/software>

ANNEXE C

Tableau I : Evolution de la profondeur et l'accélération du mouvement en fonction du temps pour le scénario 1

t (min)	0	10	20	30	40	50	60	70	80
-P (m)	0	4	15	15	15	3.9	0	0	0
a (m/s ²)	0.75	0.09	0.01	-0.01	-0.8	0.01	0.01	0.8	-0.8

Tableau II : Evolution de la profondeur et l'accélération du mouvement en fonction du temps pour le scénario 2

t (min)	0	10	20	40	45	50	60	70	100
-P (m)	0	3.7	18	18	18	9.21	8.3	8.2	8.1
a (m/s ²)	0.8	1.8	0.03	0.01	0.015	0.00	0.01	0.01	0.00

Tableau III : Evolution de la profondeur et l'accélération du mouvement en fonction du temps pour le scénario 3

t (min)	0	10	20	30	40	50	60	70	80	100
-P (m)	0	2.3	13.1	16	16	16	16	16	16	16
a (m/s ²)	1.6	1.68	1.59	0.01	-0.01	0.015	-0.01	-0.01	0.015	0.00

Tableau IV : Evolution de la profondeur et l'accélération du mouvement en fonction du temps pour le scénario 4

t (min)	0	10	20	30	40	50	60	80	100
-P (m)	0	3.8	18	18	18	4.1	0	0	0
a (m/s ²)	1.5	1.75	1.75	0.01	0	-1.5	0.01	0	0