

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
UNIVERSITE MOULOUD MAMMERI DE TIZI-OUZOU



FACULTE DU GENIE ELECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'ELECTRONIQUE

**Mémoire de Fin d'Etudes
De MASTER ACADEMIQUE**

Domaine : Sciences et Technologies

Filière : Electronique

Spécialité : Electronique des Systèmes Embarqués

Présenté par

Houfel Lyna

Zennouche Merina

Thème

**Développement d'un prototype non invasif pour
la prédiction du risque de diabète de type 2 basé
sur l'Intelligence Artificielle.**

Mémoire soutenu publiquement le 03 /07/2025 devant le jury composé de :

Mr Oualouche Fethi
President, MCA, UMMTO

Mr Lahdir Mourad
Encadreur, Professeur, UMMTO

Mr Attaf Youcef
Examineur, MCA, UMMTO

Année universitaire 2024-2025

Résumé :

Le diabète est une maladie chronique qui touche des millions de personnes à travers le monde, il nécessite une surveillance continue et précise pour prévenir les complications graves.

La détection précoce des risques liés au diabète est essentielle pour améliorer la qualité de vie des patients et faciliter une prise en charge rapide et efficace. Cependant, les méthodes traditionnelles de suivi sont souvent insuffisantes pour générer des données et anticiper les changements dans l'état de santé des patients.

Dans ce contexte, l'intégration de l'intelligence artificielle (IA) et des technologies de surveillance des paramètres vitaux constitue une avancée majeure. Ce travail propose le développement d'un prototype non invasif pour la prédiction du risque du diabète de type 2, basé sur l'IA. Le système est conçu pour utiliser un microcontrôleur ESP32 couplé à des capteurs biomédicaux afin de mesurer la fréquence cardiaque (HR), la variabilité de la fréquence cardiaque (HRV) et la pression artérielle (PA).

Toutefois, en raison de certaines limitations matérielles, seul le paramètre HR a été effectivement implémenté dans le prototype réalisé. Les données collectées sont transmises à une plateforme cloud et analysées par l'algorithme d'apprentissage Random Forest. Une application mobile permet aux utilisateurs de suivre leur état de santé et de recevoir des alertes personnalisées. Les tests effectués ont permis de valider le bon fonctionnement de la chaîne de mesure, de transmission et de prédiction à partir de la HR.

Ce dispositif constitue une première étape prometteuse vers un système complet et intelligent d'aide à la prévention du diabète.

Mots-clés : diabète, surveillance des paramètres vitaux, fréquence cardiaque (HR), variabilité de la fréquence cardiaque (HRV), pression artérielle, intelligence artificielle, ESP32, capteurs biomédicaux, cloud, application mobile, détection précoce.

Abstract:

Diabetes is a chronic disease that affects millions of people around the world. It requires continuous and precise monitoring to prevent serious complications.

The early detection of diabetes-related risks is essential to improve patients' quality of life and facilitate rapid and effective care. However, traditional monitoring methods are often insufficient to generate real-time data and anticipate changes in patients' health status.

In this context, the integration of artificial intelligence (AI) and vital signs monitoring technologies represents a major advancement. This work proposes the development of a non-invasive prototype for predicting the risk of type 2 diabetes, based on AI. The system is designed to use an ESP32 microcontroller coupled with biomedical sensors to measure heart rate (HR), heart rate variability (HRV), and blood pressure (BP).

However, due to certain hardware limitations, only the HR parameter was effectively implemented in the prototype developed. The collected data are transmitted to a cloud platform and analyzed by an AI algorithm (Random Forest). A mobile application allows users to monitor their health status and receive personalized alerts. Tests carried out validated the proper functioning of the measurement, transmission, and prediction chain based on HR.

This device represents a promising first step toward a complete and intelligent system for diabetes prevention support.

Keywords: diabetes, vital signs monitoring, heart rate (HR), heart rate variability (HRV), blood pressure (PA), artificial intelligence, ESP32, biomedical sensors, cloud, mobile application, early detection.

Liste des Acronymes

HR	Heart Rate – Fréquence cardiaque
HRV	Heart Rate Variability – Variabilité de la Fréquence cardiaque
PA	Pression Artérielle
RHR	Resting Heart Rate – Fréquence cardiaque au repos
RR	Risque Relatif
PAS	Pression Artérielle Systolique
PAD	Pression Artérielle Diastolique
PAM	Pression Artérielle Moyenne
BPM	Beats Per Minute.
BMI	Body Mass Index.
IC	Intervalle de Confiance
ECG	Electrocardiogramme
PPG	photo-pléthysmographie
SPO2	Saturation pulsée en oxygène
CA	Composante AC- Alternative Current
CC	Composante DC- Direct Current
IP	Indice de perfusion
IR	InfraRouge
Mmgh	Millimètres de mercure
Nm	Nanomètre
IA	Intelligence Artificielle.
API	Application Programming Interface.
SOC	System On chip – Système sur puce
I2C	Inter-Integrated Circuit
SPI	Serial Peripheral Interface
UART	Universal Asynchronous Receiver Transmitter

SDA Serial Data Line
SCL Serial Clock Line
RAM Random Access Memory
GND Ground
VIN Voltage Input
GPOI General Purpose Input/output
WI-FI Wireless Fidelity.
USB Universal Serial Bus
ADC Analog-to-Digital Converter
NOSQL Not Only SQL.
IDE Integrated Development Environment
IOS iPhone Operating System
GUI Graphical User Interface
CSV Comma-Separated Values
JSON JavaScript Object Notation

Table des matières

Introduction.....	8
Contexte de travail :.....	1
Problématique :.....	1
Objectifs du Travail.....	1
Organisation du mémoire :.....	2
Chapitre 1 :.....	0
I.1 Préambule	3
I.2 La maladie du diabète :.....	3
I.3 Les types de diabète :.....	4
I.3.1 Le diabète de type 1.....	4
I.3.2 Le diabète de type 2 :.....	5
I.4 Les symptômes du diabète :.....	6
I.5 Les paramètres vitaux et le diabète de type 2 :.....	7
I.5.1 La fréquence cardiaque (HR) :.....	7
I.5.1.1 La fréquence cardiaque dans le contexte du diabète :.....	7
I.5.2 La variabilité de la fréquence cardiaque (HRV) :.....	8
I.5.2.1 La variabilité de la fréquence cardiaque dans le contexte du diabète :.....	8
I.5.3 La pression artérielle (PA) :.....	9
I.5.3.1 La pression artérielle dans le contexte du diabète :.....	10
I.5.4 Indice de Masse Corporelle (BMI) :.....	11
I.5.5 Niveau de stress :.....	11
I.6 Technique de mesure des paramètres vitaux :.....	11
I.6.1 Mesure de fréquence cardiaque et variabilité de la fréquence cardiaque :.....	11
I.6.2 Mesure de la Pression Artérielle (PA) :.....	14
I.7 Apport de l'intelligence artificielle dans la prédiction du risque de diabète.....	15
I.7.1 Présentation de l'intelligence artificielle :.....	15
I.7.2 Rôle de l'intelligence artificielle dans le domaine médical :.....	15
I.7.3 Intérêt de l'intelligence artificielle pour la prédiction du diabète :.....	16
I.7.4 Avantages et limites de l'intelligence artificielle en santé :.....	16
I.8 Discussion :.....	17
CHAPITRE 2 :.....	3
II.1 Préambule :.....	18
II.2 Architecture du Système :.....	18
II.2.1 Fonctionnement du système :.....	18
II.2.2 Description des interactions :.....	19

II.3 Présentation et choix technologiques :	19
II.3.1 Composants matériels.....	19
II.3.1.1 Microcontrôleur ESP32 :	19
II.3.1.2 MAX30102 :	21
II.3.1.3 Protocole I2C :	22
II.3.2 Partie logicielle :	23
II.3.2.1 Cloud Firestor :	23
II.3.2.2 Flutter :	24
II.3.2.3 Arduino IDE :	24
II.4 Algorithme d'intelligence artificielle Random Forest :	26
II.4.1 Fonctionnement : Construction des arbres :	27
II.4.2 Avantages de l'algorithme Random Forest.....	28
II.4.3 Application dans notre système :	28
II.5 Environnement d'entraînement Python :	29
II.6 Discussion :	30
Chapitre 3 :	17
III.1 Préambule	31
III.2 Bloc de ESP32 :	31
III.3 L'organigramme du code arduino :	32
III.4 Mise en œuvre du Firebase :	33
III.4.1 Création du projet via la console Firebase :	33
III.5 Développement du modèle d'intelligence artificielle.....	37
III.5.1 Etape de développement :	37
III.5.2 Analyse des Résultats :	41
III.6 Application Flutter.....	42
III.6.1 Interface Utilisateur	43
III.6.2 Back-end de l'application	45
III.7 Intégration du modèle IA	46
III.8 Discussion :	48
Conclusion	33
Bibliographie	49

Introduction

Contexte de travail :

Le diabète représente un énorme problème de santé publique, touchant plus de 422 million de personnes à travers le monde, selon les données de l'Organisation Mondiale de la Santé. Généralement cette maladie est sans asymptotique au début mais cause de sévère complications si elle n'est pas diagnostiquée dès les débuts. Grâce aux récentes avancées technologiques en matériel embarqué, capteurs biomédicaux et intelligence artificielle (IA), il est désormais possible de concevoir des systèmes de surveillance non invasifs, offrant un suivi continu et précis des indicateurs physiologiques. Ces systèmes ouvrent la voie à une détection précoce des anomalies métaboliques liées au diabète.

Problématique :

Face à l'augmentation constante des cas de diabète dans le monde et aux limites des méthodes actuelles de dépistage précoces et fiable, une question cruciale s'impose :

- Comment concevoir un système embarqué capable d'exploiter les paramètres vitaux d'un individu via l'intelligence artificielle afin de prédire le risque de diabète de type 2 ?

Deux axes de recherche guideront notre travail :

- Quels sont les paramètres physiologiques les plus pertinents pour la détection précoce du diabète de type 2 ?
- En quoi l'intelligence artificielle peut-elle améliorer l'interprétation des données et la prédiction du risque diabétique ?

Objectifs du Travail

Notre projet vise à développer un prototype embarqué non invasif capable d'exploiter certains paramètres vitaux pour prédire le risque du diabète de type 2, pour cela nous avons identifié les principaux axes de travail suivants :

- ✓ Mesurer et transmettre les données à l'aide d'un capteur MAX30102 vers une base Firebase.

- ✓ Prédire le risque diabétique à l'aide d'un modèle d'apprentissage automatique (Random Forest).
- ✓ Afficher les résultats sur une application mobile.

Organisation du mémoire :

Notre mémoire est organisé en trois chapitres qui couvrent systématiquement les différentes étapes du développement et de l'évaluation de notre système :

Le premier chapitre présente les paramètres vitaux pertinents, des technologies de mesure, et des liens entre les paramètres physiologiques et le diabète de type 2, ainsi qu'une introduction à l'apport de l'intelligence artificielle.

Le deuxième chapitre détaille la conception et la méthodologie utilisées pour développer notre système en justifiant les choix technologiques et présente l'architecture proposée.

Le troisième chapitre se concentre sur le développement technique de notre système, tant au niveau matériel que logiciel et présente les résultats obtenus.

Nous terminons notre mémoire par une conclusion tout en donnant les perspectives nécessaires pour améliorer notre système.

Chapitre 1 :
Les paramètres vitaux pour
la prédiction du risque
diabétique

I.1 Préambule

Dans le contexte de la prévention du diabète de type 2, l'étude et la surveillance intelligente de certains paramètres vitaux constituent une voie innovante et prometteuse. L'intégration de l'intelligence artificielle dans l'analyse de données physiologiques comme la fréquence cardiaque (HR), la variabilité de la fréquence cardiaque (HRV) et la pression artérielle (PA), ouvre des perspectives vers des systèmes de détection précoce plus précis et personnalisés. Ce chapitre dresse un aperçu des connaissances actuelles sur le diabète aussi bien que sur les paramètres physiologiques pertinents et les technologies de mesure disponibles et l'approche de l'IA destinée à prédire le risque.

I.2 La maladie du diabète :

Le diabète est une maladie chronique qui touche des millions de personnes, indépendamment de leur âge ou leur mode de vie. Cette maladie se caractérise par un taux élevé de glucose (sucre) dans le sang, appelé hyperglycémie (voir figure 1). Cela se produit lorsque le pancréas ne produit pas suffisamment d'insuline, ou lorsque le corps devient résistant à cette hormone. (1)

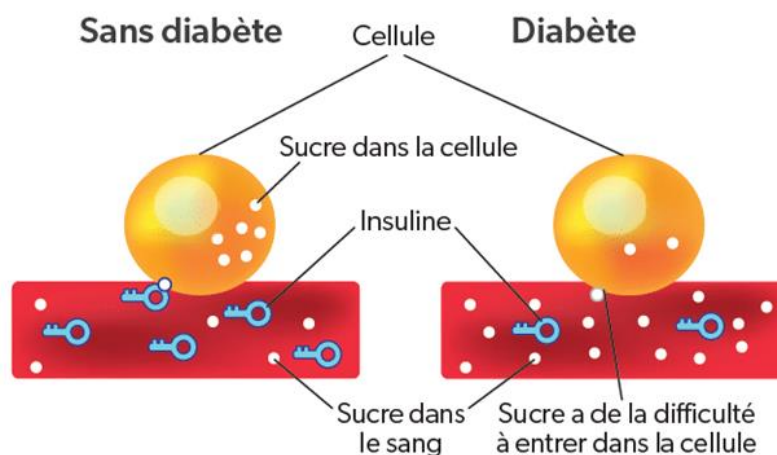


Figure 1. Absorption du glucose en présence ou en absence de diabète

L'insuline :

L'insuline est une hormone essentielle sécrétée par le pancréas, dont le rôle principal est de réguler le taux de glucose dans le sang, en permettant au sucre d'entrer dans les cellules pour être utilisé comme source d'énergie. En cas de déséquilibre, une quantité excessive de glucose s'accumule dans le sang, ce qui peut entraîner à long terme des complications graves pour la santé (voir figure 2). (2) (3)

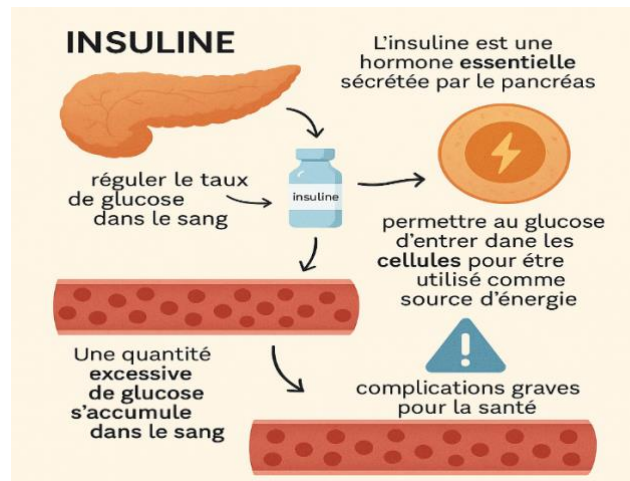


Figure 2. Le rôle de l'insuline dans le corps

I.3 Les types de diabète :

Le diabète ne se résume pas à une seule maladie clairement définie, mais il s'agit d'un regroupement de plusieurs états pathologiques partageant un point commun : un taux de glucose trop élevé dans le sang, aussi appelé hyperglycémie. Cependant, les causes ainsi que les traitements diffèrent d'un type à l'autre. Connaître ces différentes formes, permet de le détecter plus tôt et de mieux le gérer. (2)

I.3.1 Le diabète de type 1

Le diabète de type 1 est une maladie auto-immune, ça veut dire que le système immunitaire attaque sans raison les cellules bêta du pancréas, responsables de la production d'insuline (voir figure 3). Il touche souvent les enfants et les adultes de moins de 35ans.

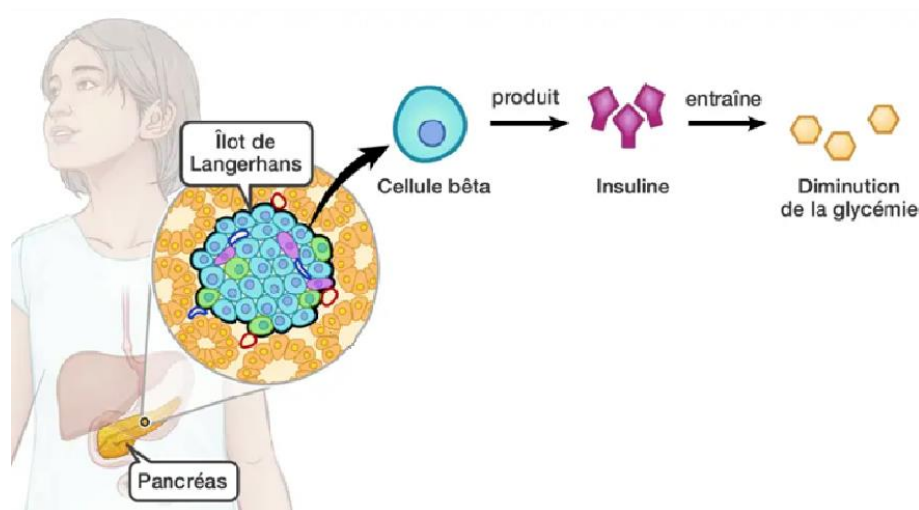


Figure 3. Fonction des cellules pancréatiques

Cette maladie peut rapidement entraîner des lésions de différents organes (voir figure 4), principalement les reins, les yeux les nerfs, ou les vaisseaux sanguins. (4)

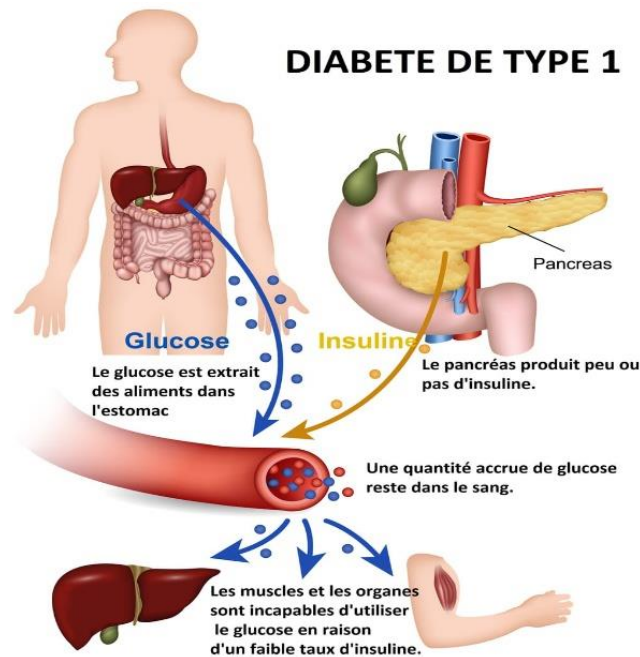


Figure 4. Schéma du diabète de type 1

I.3.2 Le diabète de type 2 :

Le diabète de type 2 (voir figure 5), représente la forme la plus courante du diabète, avec environ 90 % des cas dans le monde. Il touche essentiellement des adultes de plus de 40 ans.

Cette forme de diabète se manifeste lorsque l'organisme devient résistant à l'insuline, c'est à dire qu'il n'utilise plus correctement cette hormone ou que le pancréas ne fabrique plus suffisamment d'insuline.

Le diabète de type 2 est souvent associé à plusieurs facteurs de risque : surpoids ou obésité, mauvaise alimentation, manque d'activité physique, stress chronique ou antécédents familiaux. (5)

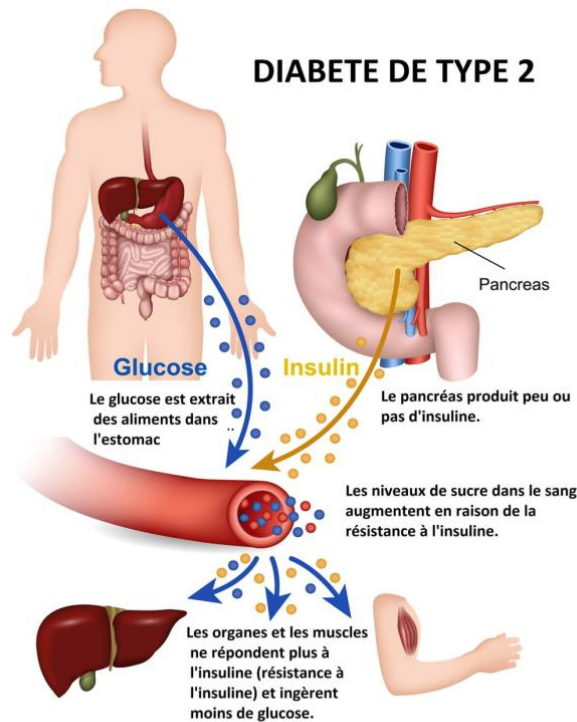


Figure 5. Schéma du diabète de type 2

I.4 Les symptômes du diabète :

Le diabète est caractérisé par divers symptômes dont l'intensité et la rapidité d'apparition varient selon le type. Parfois évidents, parfois silencieux, tous traduisent un dérèglement métabolique. Les reconnaître tôt permet un diagnostic rapide et une prise en charge adaptée.

Dans le tableau 1, nous avons cité les symptômes de diabète de type 1 et de type 2 : (4)

Diabète de type 1	Diabète de type 2
Soif intense et fréquente	Fatigue persistante
Urines abondantes et fréquentes	Infections fréquentes (urinaires, cutanées...)
Perte de poids rapide et inexplicée	Cicatrisation lente des plaies
Faim excessive (même après avoir mangé)	Engourdissement ou picotements dans les extrémités
Fatigue importante malgré le repos	Vision floue

Tableau 1. Symptômes du diabète de type 1 et de type 2

I.5 Les paramètres vitaux et le diabète de type 2 :

Le diabète affecte non seulement le métabolisme du glucose, mais aussi le système cardiovasculaire. (6)

Les paramètres les plus étudiés sont :

- La fréquence cardiaque (HR).
- La variabilité de la fréquence cardiaque (HRV)
- La pression artérielle (PA).

Ces paramètres sont parmi les critères de diagnostic les plus fiables pour juger l'état du système nerveux autonome et les risques associés au diabète de type 2. Leur surveillance permet d'évaluer précocement les déséquilibres et d'adapter la prise en charge en conséquence.

I.5.1 La fréquence cardiaque (HR) :

La fréquence cardiaque (HR) représente le nombre de battements du cœur par minute. Chez une personne adulte en bonne santé, elle varie généralement entre 60 et 100 battements par minute (BPM) au repos. Cette fréquence est contrôlée principalement par le système nerveux autonome, qui ajuste le rythme cardiaque en fonction des besoins du corps (repos, activité physique, stress...). (7)



Figure 6. Signal de fréquence cardiaque

I.5.1.1 La fréquence cardiaque dans le contexte du diabète :

Plusieurs études ont montré que des anomalies de la fréquence cardiaque peuvent constituer des indicateurs précieux du risque de développer cette maladie. En particulier, la fréquence

cardiaque au repos (RHR) est souvent utilisée comme paramètre pour détecter les signes précoces de dysfonctionnement métabolique, y compris le diabète. (8)

Une étude récente menée par Kim (9) a fourni des preuves solides de cette relation. Cette recherche a suivi 8 392 participants non diabétiques pendant une période médiane de 12,3 ans. Les résultats ont révélé que :

- Une RHR ≥ 80 battements par minute (bpm) est fortement associée à un risque accru de diabète, tant chez les hommes que chez les femmes.
- Une diminution de la RHR d'au moins 5 bpm par rapport à une RHR stable est associée à une réduction du risque de diabète chez les deux sexes.

De manière notable, une augmentation de la RHR d'au moins 5 bpm est associée à un risque accru de diabète, mais uniquement chez les femmes, suggérant l'existence de différences physiologiques liées au sexe dans cette relation.

Ces résultats sont corroborés par la méta-analyse (8) , qui a démontré qu'une augmentation de 10 bpm de la HR est associée à une hausse de 25 % du risque de diabète de type 2 (RR = 1,25 ; IC 95 % : 1,14–1,36).

I.5.2 La variabilité de la fréquence cardiaque (HRV) :

La variabilité de la fréquence cardiaque est un indicateur qui mesure les variations d'intervalle entre deux battements cardiaques successifs. Elle reflète la capacité du cœur à s'adapter aux stimuli internes et externes grâce à l'action combinée du système nerveux autonome. (10)

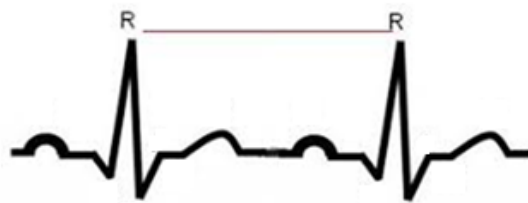


Figure 7. La variabilité de la fréquence cardiaque

I.5.2.1 La variabilité de la fréquence cardiaque dans le contexte du diabète :

La variabilité de la fréquence cardiaque (HRV) est un indicateur clé de la régulation du système nerveux autonome et présente une forte corrélation avec le diabète. (11)

Une HRV faible est fréquemment observée chez les patients diabétiques, ce qui suggère une dysfonction autonome.

Les personnes atteintes de diabète de type 2, en particulier, présentent souvent une réduction significative de la HRV, traduisant un déséquilibre entre les branches sympathique et parasympathique du système nerveux autonome.

L'analyse des signaux ECG, notamment via des techniques d'apprentissage automatique, permet de détecter précocement ces anomalies de la HRV. Cette approche facilite l'identification des individus à risque avant même l'apparition de symptômes cliniques. Ainsi, la HRV s'impose comme un outil de dépistage précoce et de gestion du diabète. (12)

I.5.3 La pression artérielle (PA) :

La pression artérielle est la force que le sang exerce contre les parois des artères, généralement mesurée en millimètres de mercure (mm-Hg). On distingue trois pressions : la pression systolique (lors de la contraction du cœur), la pression diastolique (lorsque le cœur est au repos) et la pression artérielle moyenne (voir figure 6). (13) (14)

- **La pression artérielle systolique (PAS) :** Elle Correspond à la contraction de ventricule gauche et l'éjection du sang dans l'aorte et puis dans le système vasculaire.
- **La pression artérielle diastolique (PAD) :** Elle Correspond à la phase de relâchement du ventricule gauche et le sang a déjà pris son chemin dans le système vasculaire.
- **La pression artérielle moyenne (PAM) :** Elle représente la moyenne temporelle des pressions artérielles enregistrées pendant un cycle cardiaque quand seules PAS et PAD sont disponibles ; elle est donnée par l'équation (1) :

$$PAM = \frac{1}{T} \int Pa(t) dt \quad (1)$$

Les trois pressions sont liées par l'équation (2):

$$PAM = PAD + k. [PAS - PAD] \quad (2)$$

Avec **K** est une constante.

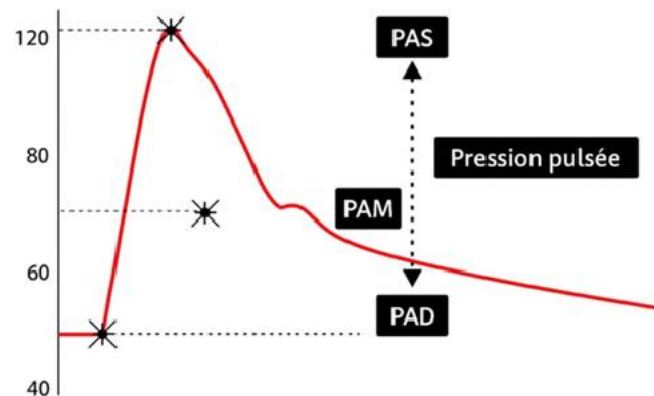


Figure 8. Courbe de la pression

I.5.3.1 La pression artérielle dans le contexte du diabète :

La pression artérielle (PA) est identifiée comme un facteur de risque majeur du diabète de type 2. Une étude prospective, menée dans le cadre du Women's Health Study (15) , a mis en évidence qu'un niveau élevé de PA, même chez des individus initialement en bonne santé, est fortement associé à une augmentation du risque de développer un diabète de type 2.

L'étude a suivi 38 172 femmes sans antécédents de diabète ni de maladies cardiovasculaires pendant une période médiane de 10,2 ans. Les participantes ont été réparties en quatre groupes selon leur PA initiale. Les résultats ont montré que :

- Les femmes avec une pression artérielle systolique (PAS) de 130–139 mmHg et une pression diastolique (PAD) de 85–89 mmHg présentaient un risque accru de diabète (multiplié par 1,45).
- Celles ayant une hypertension ($PAS \geq 140$ mmHg et $PAD \geq 90$ mmHg) voyaient leur risque multiplié par 2,03 par rapport aux femmes ayant une PA normale.

Par ailleurs, l'élévation progressive de la PA durant la période de suivi était également associée à un risque accru de diabète, soulignant l'importance de surveiller régulièrement la PA pour prévenir le diabète de type 2.

Ces résultats confirment que la pression artérielle, qu'elle soit déjà présente ou en développement, constitue un facteur de risque dans la prévention du diabète.

I.5.4 Indice de Masse Corporelle (BMI) :

L'indice de masse corporelle est un indicateur utilisé pour évaluer la corpulence d'un individu. Il se calcule à partir du poids et de la taille selon l'équation (3) :

$$BMI = \frac{\text{poids (kg)}}{[\text{taille (m)}]^2} \quad (3)$$

Le BMI permet de classer les individus en différentes catégories : insuffisance pondérale, poids normal, surpoids ou obésité. Un BMI élevé est reconnu comme un facteur de risque majeur du diabète de type 2, notamment en lien avec l'insulino-résistance souvent associée à l'obésité. (16)

I.5.5 Niveau de stress :

Le niveau de stress est également un indice pertinent dans l'évaluation du risque de diabète.

Une étude menée par des chercheurs de l'Université de Göteborg en Suède a montré que les personnes ayant un stress permanent ont 45% de risques en plus de développer un diabète de type 2, par rapport à ceux qui ne déclaraient aucun stress. Ce lien reste significatif, même après ajustement par l'âge, le statut ou l'inactivité physique. (17)

I.6 Technique de mesure des paramètres vitaux :

Avec les avancées techniques, il est aujourd'hui possible de mesurer en continu des indicateurs comme la fréquence cardiaque (HR), la variabilité de la fréquence cardiaque (HRV) ou encore la pression artérielle (PA), en utilisant des capteurs plus miniaturisés, précis et accessibles. Cette partie présente les principaux types de capteurs et dispositifs utilisés pour la collecte de ces données physiologiques.

I.6.1 Mesure de fréquence cardiaque et variabilité de la fréquence cardiaque :

La mesure du (HR) repose principalement sur deux catégories de capteurs appliquant deux principes physiques :

- Le mode électrique basé sur l'électrocardiographie (ECG),

- Le mode optique basé sur la photo-pléthysmographie (PPG).

Dans notre cas, nous intéressons au mode optique :

Le Photo-pléthysmogramme (PPG) est un signal mesurable à travers des circuits optoélectroniques à des longueurs d'ondes spécifiques de façon non-invasive (voir figure 8).

La photo-pléthysmographie (PPG) (voir figure 7), est une méthode optique non invasive permettant de détecter les variations du volume sanguin dans les tissus. Elle repose sur le principe d'absorption de la lumière.

Cette technique consiste à placer le doigt (généralement l'index) dans un capteur équipé de deux sources lumineuses, une LED rouge et une LED infrarouge permettant l'émission de la lumière à travers le doigt sur lequel une vasodilatation est appliquée et un phototransistor permettant la détection de la lumière transmise. (18)

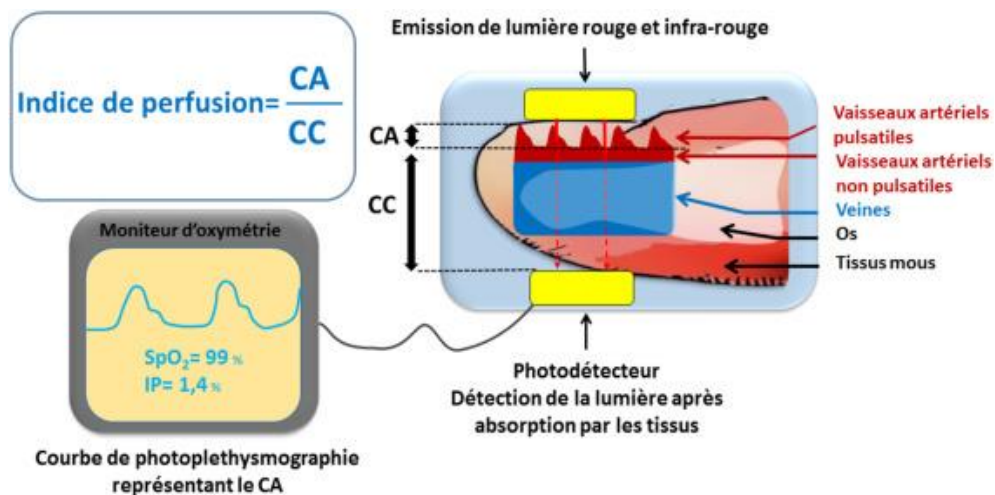


Figure 9. Principe de la photo-pléthysmographie

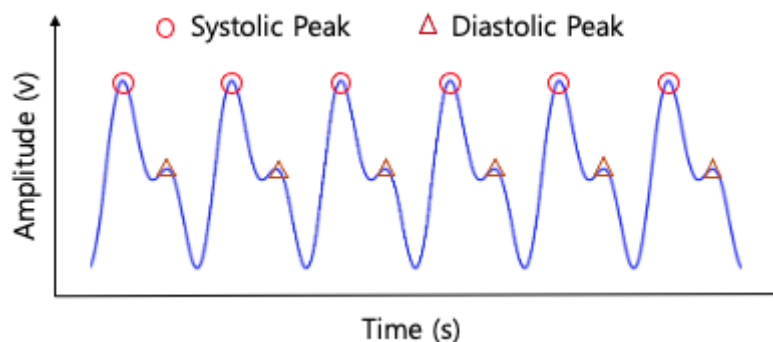


Figure 10. Signal PPG

La fréquence cardiaque peut être mesurée en utilisant l'équation (4) :

$$HR = \frac{60}{T} \quad (4)$$

Avec **T** est le temps mesuré entre deux pics successifs du signal PPG (voir figure 9).

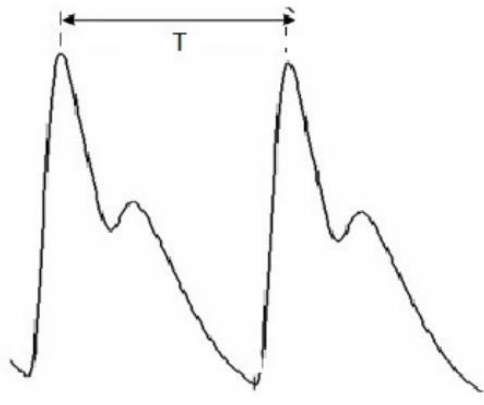


Figure 11. Le temps entre deux pics du signal PPG

- **Fonctionnement :**

Lorsqu'un battement cardiaque se produit, la teneur en sang dans les capillaires sous-cutanés augmente, modifiant la manière dont la lumière est absorbée ou réfléchi.

Ces variations lumineuses sont détectées par le photo-détecteur et transformées en un signal PPG, qui peut être utilisé pour estimer la fréquence cardiaque (HR), ainsi que pour dériver des mesures telles que la variabilité de la fréquence cardiaque (HRV).

- Exemples de capteurs PPG utilisés dans les systèmes embarqués :

- ✓ MAX30100 : combine un émetteur LED (rouge et infrarouge) et un photo-détecteur. Il permet de déterminer la fréquence cardiaque ainsi que la saturation en oxygène du sang (SpO2). Couramment utilisé avec des microcontrôleurs comme l'ESP32 dans des applications de santé connectée.

- ✓ MAX30102 : version améliorée du MAX30100, offrant une meilleure sensibilité et de meilleures performances anti-bruit. Il est privilégié dans les dispositifs de suivi de santé portables grâce à sa fiabilité et sa faible consommation d'énergie.

I.6.2 Mesure de la Pression Artérielle (PA) :

La pression artérielle est traditionnellement mesurée de plusieurs manières : (19)

- ✓ Invasive → cathéter artériel (en centre hospitalier)
 - ✓ Non invasive → brassard gonflable (méthode oscillométrique)
 - ✓ Non invasive sans brassard → par estimation sur signaux PPG et électriques et utilisant des algorithmes sophistiqués.
- **Fonctionnement :**
 - **Méthode oscillométrique :** Un brassard se gonfle et se dégonfle autour du bras pour enregistrer les variations de pression et calculer la pression systolique, diastolique, et moyenne (voir figure 10).

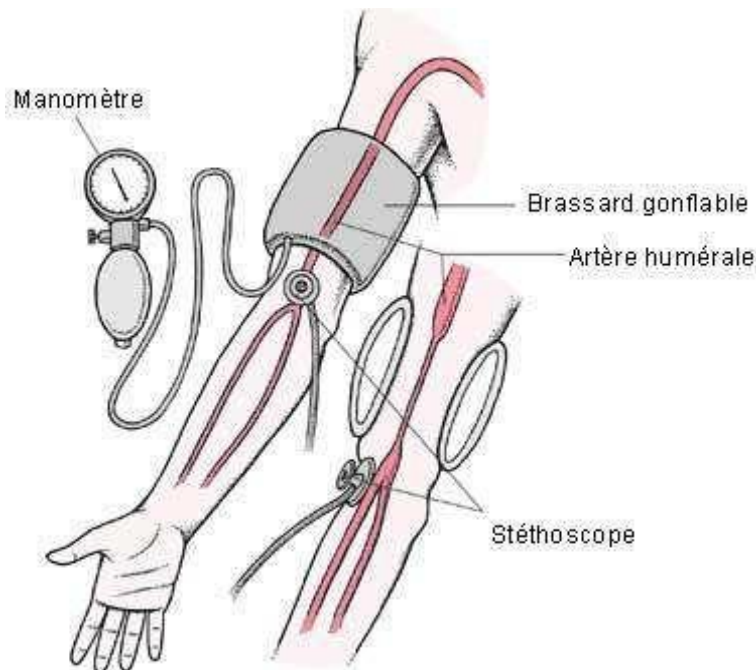


Figure 12. Méthode de mesure de la PA

- **Méthode optique :** Utilisation de capteurs PPG pour estimer les variations de volume sanguin et, grâce au traitement du signal, estimer la pression artérielle sans brassard.

Les techniques de mesures donnent des données brutes sur l'état psychologique du patient. Pour approfondir l'analyse et permettre une détection prématurée des risques, il est obligatoire de les traiter intelligemment grâce à l'aide de méthodes avancées telles que l'intelligence artificielle.

I.7 Apport de l'intelligence artificielle dans la prédiction du risque de diabète

I.7.1 Présentation de l'intelligence artificielle :

L'intelligence artificielle (IA) présente un ensemble de techniques qui permettent à une machine d'apprendre à partir de données, reconnaître des modèles et aussi de prendre des décisions. Dans le domaine médical, elle est notamment utilisée pour analyser de grandes quantités de données physiologiques, détecter des anomalies et prédire le risque de maladies chroniques. Elle permet ainsi de renforcer la précision des diagnostics et d'anticiper les complications avant l'apparition des symptômes.

Dans notre projet, l'intelligence artificielle est exploitée à travers le Machine Learning, une branche de l'IA qui permet aux algorithmes d'apprendre à partir de données étiquetées afin de prédire un résultat. Plus précisément, nous utilisons un algorithme d'apprentissage supervisé appelé Random Forest qui est bien connu pour sa robustesse, sa capacité à gérer diverses données et ses fortes performances dans les tâches de classification médicale. (20)

I.7.2 Rôle de l'intelligence artificielle dans le domaine médical :

L'intelligence artificielle transforme progressivement le domaine médical en offrant de nouvelles approches pour analyser, interpréter et exploiter les données de santé. Grâce à ses capacités d'apprentissage à partir de grandes quantités de données, elle permet de détecter automatiquement des anomalies, d'identifier des schémas complexes et d'anticiper certains risques de santé. (21)

Son utilisation s'étend aujourd'hui à plusieurs domaines, notamment :

- Aide au diagnostic à partir de données cliniques ou d'images médicales.
- Amélioration de la prise de décision clinique
- La personnalisation des traitements en fonction du profil du patient.

I.7.3 Intérêt de l'intelligence artificielle pour la prédiction du diabète :

L'intelligence artificielle permet de surmonter les limites des approches traditionnelles en identifiant automatiquement des modèles cachés au sein des données. Elle peut analyser de nombreuses variables simultanément, repérer des schémas complexes, et ainsi prédire avec plus de précision le risque de développer un diabète. Cette capacité d'analyse est particulièrement précieuse dans un contexte de médecine préventive, où l'objectif est d'agir avant l'apparition des symptômes. (22)

En résumé, l'IA améliore l'efficacité du dépistage, renforce la personnalisation des recommandations médicales, et facilite l'intégration de la prévention dans des dispositifs connectés accessibles au grand public.

I.7.4 Avantages et limites de l'intelligence artificielle en santé :

L'intelligence artificielle (IA) apporte de nombreux bénéfices dans le domaine de la santé, en particulier dans la détection précoce des maladies chroniques comme le diabète. Son principal avantage réside dans sa capacité à analyser rapidement de grandes quantités de données, à en extraire des schémas complexes et à fournir des prédictions personnalisées. (23)

Parmi les principaux atouts de l'IA en santé, on peut citer :

- La rapidité et l'automatisation de l'analyse, réduisant le temps nécessaire à la prise de décision.
- La personnalisation du suivi, en adaptant les recommandations à chaque patient.
- La capacité à améliorer le dépistage précoce, en identifiant des signaux faibles souvent négligés.
- L'intégration dans des dispositifs embarqués, permettant un suivi continu et à distance.

Cependant, l'utilisation de l'IA en santé présente également certaines limites :

- La dépendance à la qualité des données : des données incomplètes ou bruitées peuvent fausser les résultats.
- Le manque de transparence de certains algorithmes, souvent perçus comme des « boîtes noires ».
- Les questions éthiques liées à la confidentialité, à la sécurité des données de santé et à la responsabilité en cas d'erreur.

Malgré ces défis, l'intelligence artificielle représente une avancée majeure dans le domaine médical et continue d'évoluer vers des solutions plus fiables, transparentes et accessibles.

I.8 Discussion :

Ce premier chapitre a permis de dresser un état des lieux intégrant les connaissances sur le diabète de type 2, les paramètres physiologiques (HR, HRV, PA), les technologies de mesure correspondantes, ainsi que l'intégration de l'intelligence artificielle dans la prédiction du risque diabétique. Dans le chapitre suivant, nous allons présenter la conception de notre système de prédiction du diabète de type 2

CHAPITRE 2 :

Conception et Méthodologie

II.1 Préambule :

Ce chapitre décrit le processus adopté pour la collecte de données, leur traitement à l'aide d'un modèle IA, le stockage cloud et la visualisation sur une application mobile. L'objectif est d'expliquer l'architecture technique du prototype développé, en mettant en évidence les différents composants matériels et logiciels.

II.2 Architecture du Système :

Le système que nous avons développé pour ce projet est basé sur une architecture modulaire et connectée, qui combine la collecte de données biométriques, le traitement embarqué, le stockage à distance et l'exploitation de l'intelligence artificielle.

II.2.1 Fonctionnement du système :

Le système démarre lorsque l'utilisateur lance l'application et clique sur le bouton « + » pour obtenir les mesures (ce qui envoie un signal via Firebase pour que le capteur commence à mesurer). Une fois les mesures effectuées, elles sont envoyées du capteur vers Firebase, puis récupérées par l'application et affichées à l'utilisateur.

L'utilisateur continue ensuite en saisissant ses informations personnelles, et toutes les données, y compris celles du capteur, sont enregistrées.

Enfin, en cliquant sur l'option « Check », les données sont analysées par un d'apprentissage automatique (Random Forest) intégré à l'application, afin de déterminer le niveau de risque de diabète : faible ou élevé (voir la figure 11).



Figure 13. Diagramme de l'architecture du système

Plus précisément, notre système utilise des composants matériels et logiciels qui fonctionnent ensemble de manière cohérente :

- MAX30102 : capteur optique permettant de mesurer la fréquence cardiaque et l'intensité infrarouge (IR), à partir desquelles d'autres paramètres sont estimés.
- ESP32 : microcontrôleur connecté en Wi-Fi, chargé de lire les données du capteur, de les traiter (HR, HRV estimée, PA simulée), puis de les envoyer vers le cloud.
- Firebase Realtime Database : cloud utilisée pour stocker les données en temps réel, permettant leur consultation à distance.
- Modèle IA (Random Forest) : algorithme d'apprentissage automatique utilisé pour prédire le risque de diabète de type 2 à partir des données collectées.
- Application mobile Flutter : interface utilisateur permettant d'afficher les données en temps réel via une connexion à Firebase.

II.2.2 Description des interactions :

Les différentes unités du système interagissent comme suit :

- ✓ Le capteur MAX30102 transmet les données de fréquence cardiaque (HR) au microcontrôleur ESP32 via le protocole I2C.
- ✓ L'ESP32 transmet les résultats à Firebase via Wifi.

Les données stockées dans Firebase sont accessibles :

- ✓ Par un modèle IA Random Forest, pour prédire les risques du diabète,
- ✓ Par une application mobile Flutter, qui permet à l'utilisateur de consulter ses mesures à distance.

II.3 Présentation et choix technologiques :

II.3.1 Composants matériels

II.3.1.1 Microcontrôleur ESP32 :

L'ESP32 (voir la figure 12), est un système sur puce (SoC) développé par la société Espressif Systems, qui intègre à la fois un microcontrôleur 32 bits, une connectivité Wi-Fi et Bluetooth, ainsi que de nombreuses interfaces de communication (I2C, SPI, UART), ce qui le rend compatible avec de nombreux capteurs. Il repose sur un processeur dual-core Tensilica LX6

cadencé jusqu'à 240 MHz, et dispose d'une mémoire RAM suffisante pour des applications embarquées complexes. (24)

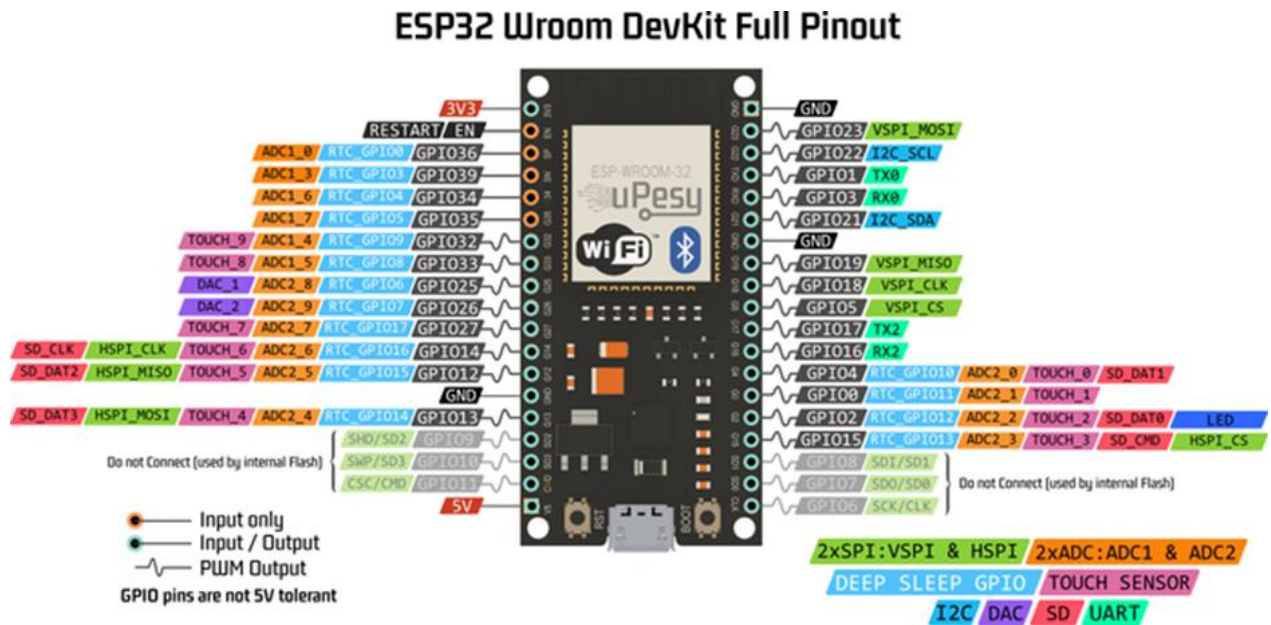


Figure 14. Schéma des broches de la carte ESP32

Le choix de l'ESP32 dans notre système s'explique par plusieurs facteurs :

- ✓ Sa connectivité Wi-Fi intégrée, essentielle pour la transmission des données vers Firebase.
- ✓ Sa capacité de traitement suffisante, permettant d'effectuer localement les calculs simples (moyennes, estimations) et de gérer le capteur sans composant externe.
- ✓ Large compatibilité logicielle, facilitant l'intégration avec des applications mobiles et des services cloud.
- ✓ Une faible consommation énergétique, avantage majeur pour les dispositifs portables.
- ✓ Son faible coût et sa disponibilité sur le marché en font un choix idéal pour des prototypes IoT dans le domaine biomédical.

II.3.1.2 MAX30102 :

Le MAX30102 (voir la figure 13) est un capteur optique intégré, conçu pour mesurer la fréquence cardiaque (HR) et la saturation en oxygène du sang (SpO₂). Il combine deux LED (rouge et infrarouge), un photo-détecteur, des optiques, ainsi qu'un circuit de traitement analogique/numérique, le tout dans un boîtier compact à faible consommation énergétique. Il communique avec les microcontrôleurs via le protocole I2C, ce qui le rend facile à intégrer dans des systèmes embarqués (25)

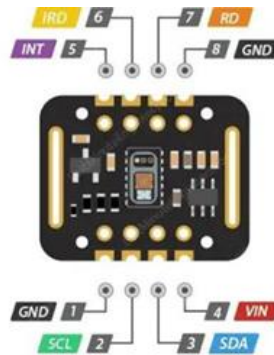


Figure 15. Le capteur MAX30102

- **Fonctionnement du capteur MAX30102 :**

La figure (14) présente le fonctionnement de capteur MAX30102, en mesurant la quantité de lumière réfléchiée par le sang. Pour mesurer la fréquence cardiaque, il utilise l'absorption de la lumière infrarouge par l'hémoglobine oxygénée dans le sang artériel. (25)

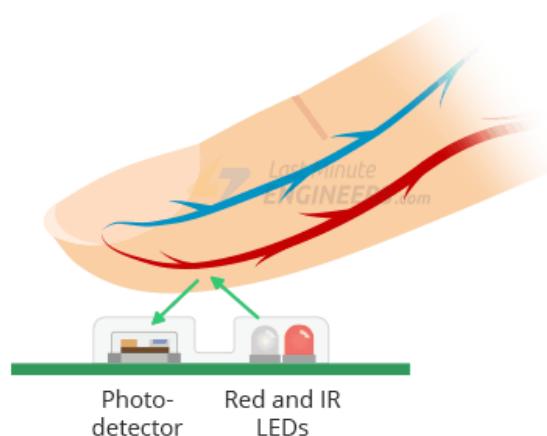


Figure 16. Fonctionnement de capteur MAX30102

Nous avons choisi le capteur MAX30102 pour les raisons suivantes :

- ✓ Mesure non invasive de la fréquence cardiaque et du signal IR, utile pour estimer d'autres paramètres comme le HRV ou la pression artérielle simulée.
- ✓ Communication I2C, compatible avec l'ESP32 sans ajout de convertisseur externe.
- ✓ Taille compacte et faible consommation, idéal pour les dispositifs embarqués ou portables.
- ✓ Documentation et bibliothèques disponibles, facilitant l'intégration et l'utilisation sous Arduino.
- ✓ Bon rapport qualité/prix, très accessible pour les projets académiques.

II.3.1.3 Protocole I2C :

Le protocole I2C (voir la figure 15) est un protocole de communication série synchrone permettant à un microcontrôleur de dialoguer avec plusieurs périphériques via seulement deux lignes pour transmettre et recevoir des données :

- **SDA (Serial Data)** : ligne de données. Elle sert à transmettre les bits entre le maître et l'esclave, un bit à la fois, synchronisé par les impulsions de l'horloge sur SCL.
- **SCL (Serial Clock)** : ligne d'horloge. Elle est générée par le maître pour synchroniser les échanges de données sur le bus.

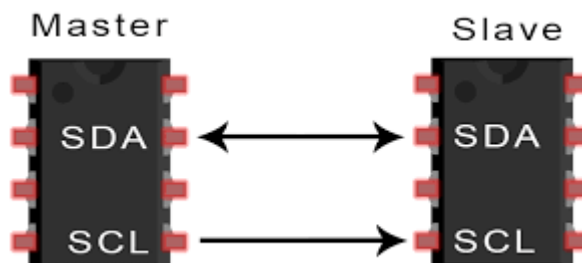


Figure 17. Schéma de communication I2C maître/esclave

Ce protocole a été développé par Philips pour réduire le nombre de connexions nécessaires à la communication entre les composants électroniques. Il est largement utilisé dans les systèmes embarqués, notamment pour interfacer des capteurs, mémoires, afficheurs ou horloges. (26)

Dans notre système :

- L'ESP32 agit comme maître I2C,
- Le capteur MAX30102 est un esclave,

Les connexions sont réalisées via les broches :

- GPIO 21 (SDA)
- GPIO 22 (SCL)

II.3.2 Partie logicielle :

II.3.2.1 Cloud Firestore :

Cloud Firestore est un service de base de données NoSQL proposé par Google via la plateforme Firebase. Il permet de stocker, synchroniser et interroger des données structurées à l'aide de documents et de collections. Firestore est conçu pour les applications mobiles, web et serveur, offrant une forte évolutivité, une structure de données flexible et une synchronisation des données entre les clients et le cloud. (27)

Dans notre projet, Firebase est utilisé pour stocker les données physiologiques (HR) mesurées par l'ESP32 ainsi que les données saisies par l'utilisateur, afin qu'elles puissent être consultées depuis une application mobile ou utilisées pour des traitements par intelligence artificielle (voir la figure 16).

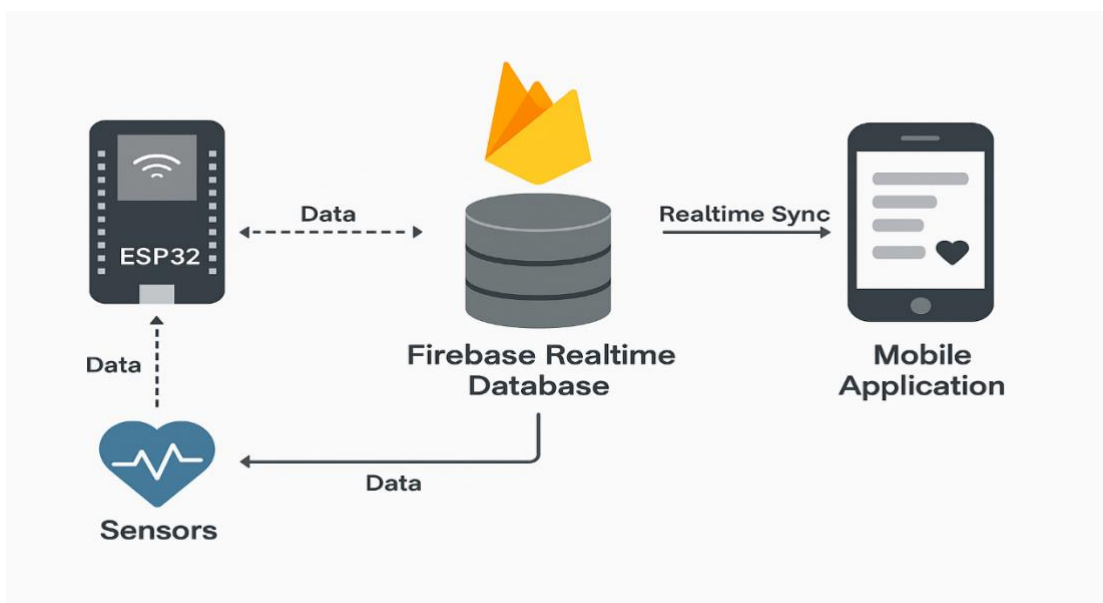


Figure 18. Architecture de communication entre esp32, capteur, Firebase et application

II.3.2.2 Flutter :

Flutter est un Framework open-source de Google, permet de développer des applications mobiles multiplateformes (Android, iOS), avec une seule base de code en Dart. Il compile en natif, offre d'excellentes performances grâce à son moteur graphique Skia, et permet une interface fluide et réactive. Sa compatibilité avec Firebase et son large écosystème en font un outil puissant pour créer des applications modernes et connectées. (28)

Dans notre projet, Flutter est utilisé pour développer une application mobile, permettant à l'utilisateur final de consulter les données physiologiques stockées dans Firebase.

Nous avons choisi flutter pour les raisons suivantes :

- ✓ Développement multiplateforme (Android/iOS), avec un seul code source.
- ✓ Excellente intégration avec Firebase, facilitant l'affichage en temps réel des mesures.
- ✓ Interfaces modernes et fluides, améliorant l'expérience utilisateur.
- ✓ Large communauté et documentation, accélérant le développement.
- ✓ Compatible avec des API IA ou de visualisation, pour les versions futures.

II.3.2.3 Arduino IDE :

Arduino IDE est un environnement de développement open-source conçu pour programmer des microcontrôleurs.

L'IDE Arduino permet :

- D'éditer un programme : les programmes sont écrits en langage C.
- De compiler ce programme : la compilation est une traduction du langage C vers le langage du microcontrôleur.
- De téléverser le programme : le téléversement se passe via le port USB de l'ordinateur
- De communiquer avec la carte Arduino grâce au terminal (ou moniteur série).

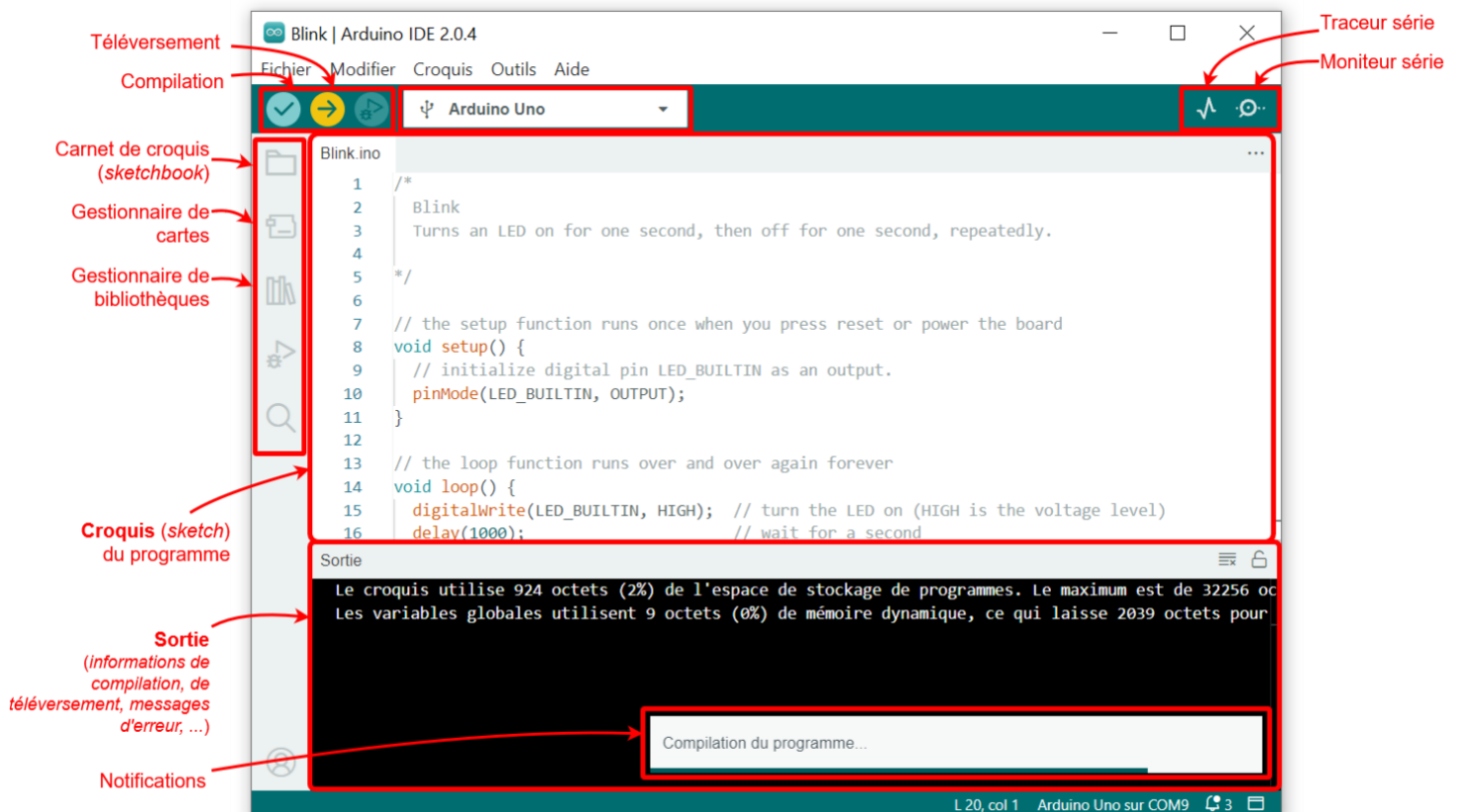


Figure 19. Interface de l'environnement Arduino IDE

L'interface graphique (voir la figure 17) du logiciel comprend plusieurs composants essentiels à la programmation :

- **Une barre de menu** : Comme dans la majorité des logiciels dotés d'une interface graphique (GUI), elle regroupe les différentes options et fonctions accessibles à partir de menus déroulants.
- **Une barre de boutons** : cette barre nous offre un accès rapide et direct aux fonctions essentielles du logiciel.
- **Un Editeur** : c'est l'espace dans lequel on peut écrire notre programme.
- **Une zone de message** : elle permet d'afficher et indiquer l'état des actions en cours comme la vérification du code ou son téléversement sur la carte.
- **Une console texte** : elle permet d'afficher les messages concernant le résultat de la compilation du programme (29)

- **Un terminal série** : ce moniteur permet de visualiser les données envoyées depuis la carte vers l'ordinateur telles que des messages ou des valeurs numériques (figure 18).

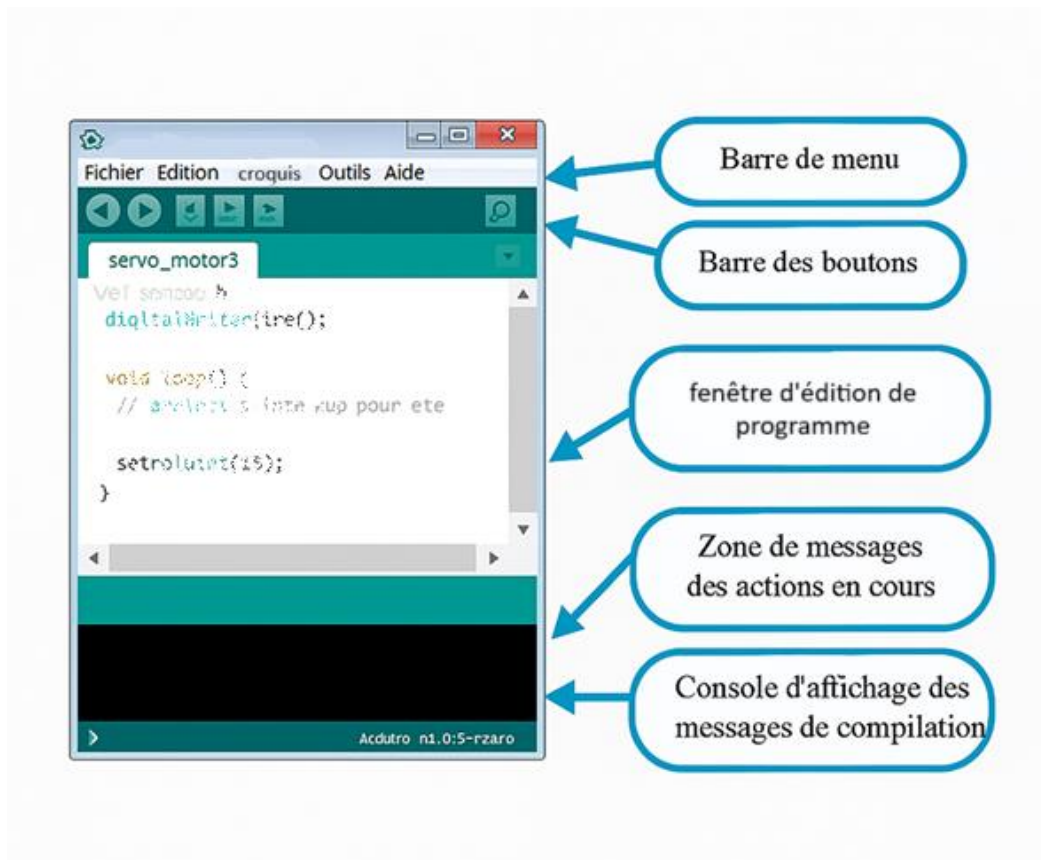


Figure 20. Les différentes parties de la fenêtre principale du logiciel Arduino

II.4 Algorithme Random Forest :

Random Forest (forêt aléatoire) est un algorithme d'apprentissage supervisé basé sur l'apprentissage ensembliste, qui combine les prédictions de plusieurs arbres de décision pour améliorer la précision et la robustesse. (30)

Contrairement à un arbre de décision unique, qui peut souffrir de surajustement (overfitting), le Random Forest introduit de l'aléatoire à deux niveaux :

Échantillonnage des données : Chaque arbre est entraîné sur un sous-ensemble aléatoire des données (méthode dite bootstrap).

Sélection des caractéristiques : À chaque split d'un nœud, seule une fraction aléatoire des caractéristiques est considérée (voir la figure 19).

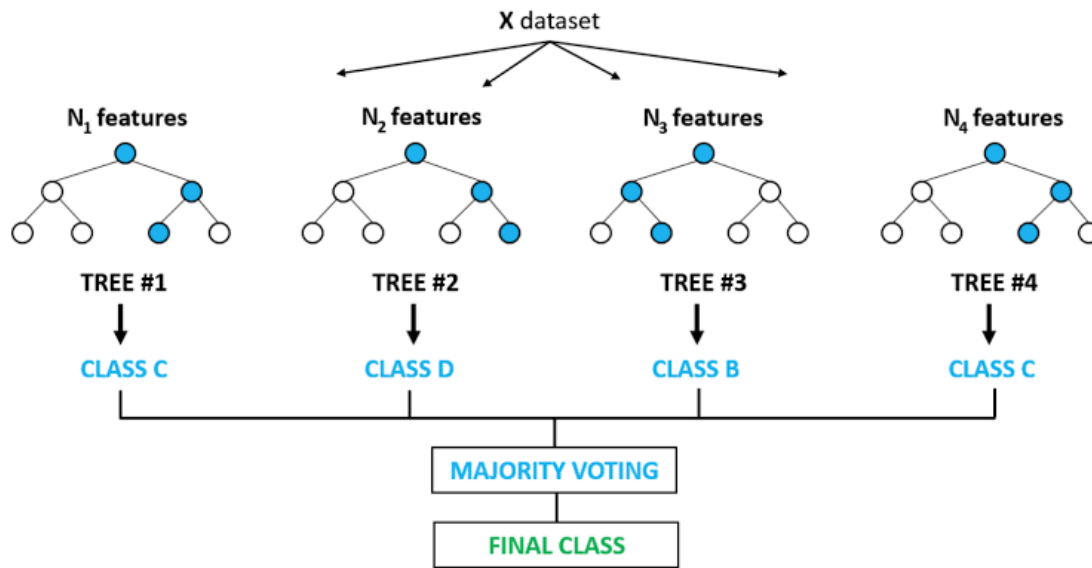


Figure 21. Algorithme de Random Forest

II.4.1 Fonctionnement : Construction des arbres :

L'algorithme Random Forest repose sur la création de plusieurs arbres de décision indépendants, construits à partir de sous-échantillons aléatoires du jeu de données initial. Cette technique s'appelle le bootstrap aggregating ou bagging.

Pour chaque arbre :

- Un échantillon aléatoire avec remise est tiré du jeu de données d'apprentissage. Chaque arbre est donc entraîné sur un jeu de données légèrement différent.
- À chaque nœud de décision, seul un sous-ensemble aléatoire de variables est considéré pour effectuer la séparation optimale. Cela limite la corrélation entre les arbres.

Une fois tous les arbres construits, la prédiction finale se fait (voir la figure 20) :

- Soit par vote majoritaire (classification),
- Soit par moyenne des résultats (régression).

Cette méthode d'ensemble permet d'obtenir un modèle plus stable, moins sensible au bruit, et plus performant que les arbres de décision pris individuellement. (31)

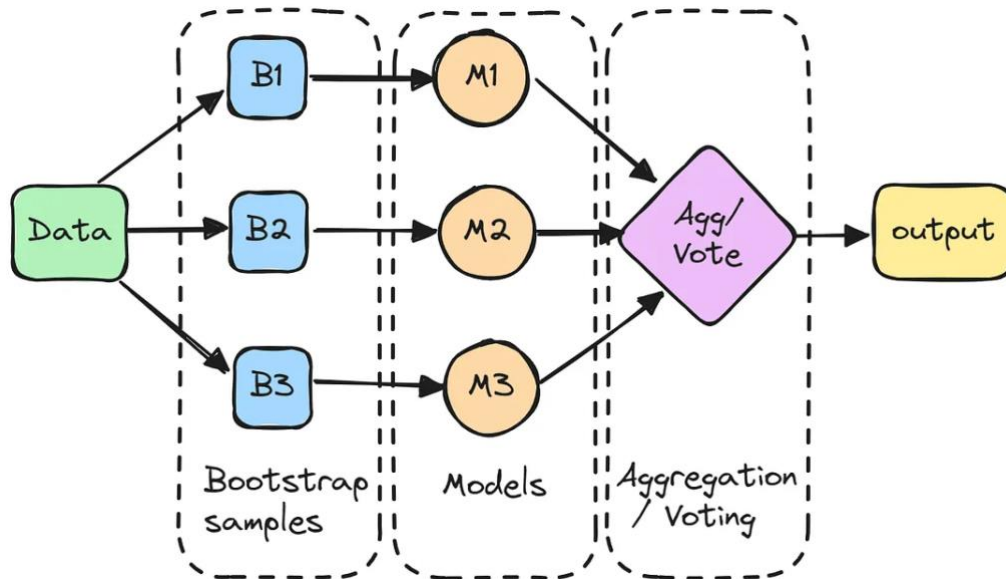


Figure 22. Schéma de fonctionnement de l'algorithme Random Forest

II.4.2 Avantages de l'algorithme Random Forest

- ✓ Robustesse au surajustement : La diversification des arbres réduit le risque de mémorisation du bruit.
- ✓ Gestion des données complexes : Efficace avec des données non linéaires, des caractéristiques corrélées, et des valeurs manquantes.
- ✓ Interprétabilité partielle : Mesure de l'importance des caractéristiques (gain moyen de pureté des nœuds) pour identifier les variables clés.
- ✓ Polyvalence : Adapté à la classification (ex. détection d'anomalies cardiaques) et à la régression (ex. prédiction de la SpO2).

II.4.3 Application dans notre système :

Dans notre projet, l'algorithme Random Forest est utilisé pour prédire le risque de diabète de type 2 à partir de :

- ✓ Fréquence cardiaque (HR).
- ✓ Indice de masse corporelle (BMI).
- ✓ Gender.
- ✓ Family history.
- ✓ Âge.
- ✓ Niveau de stress.

En s'appuyant sur un jeu de données préalablement généré et annoté, le modèle est entraîné hors ligne, puis intégré au système pour fournir une évaluation du risque, basée sur les mesures transmises par le microcontrôleur ESP32 vers Firebase.

Cette approche permet de passer d'une simple surveillance de signaux à une interprétation clinique intelligente, rendant le système plus utile dans un contexte de santé préventive.

II.5 Environnement d'entraînement Python :

Python est un langage de programmation interprété, simple et puissant, particulièrement utilisé dans les domaines de l'intelligence artificielle, du Machine Learning, et de la science des données. Sa syntaxe claire et sa courbe d'apprentissage rapide permettent de se concentrer sur la logique du projet sans se perdre dans des détails techniques complexes. (32)

Dans le cadre de ce projet, Python a été utilisé pour entraîner, évaluer et valider le modèle Random Forest, à partir d'un jeu de données contenant des paramètres physiologiques tels que la fréquence cardiaque, la variabilité de fréquence cardiaque, la pression artérielle, l'âge et le BMI.

➤ Le choix de python

- ✓ Python a été choisi pour les raisons suivantes :
- ✓ Riche écosystème de bibliothèques IA, notamment pandas, scikit-learn, matplotlib.
- ✓ Facilité de traitement de jeux de données tabulaires (CSV).
- ✓ Outils puissants de visualisation pour l'analyse des résultats.
- ✓ Rapidité de développement, adaptée au cadre d'un projet académique.
- ✓ Interopérabilité, permettant d'exporter les résultats vers Firebase ou d'autres services.

II.6 Discussion :

Dans ce chapitre, nous avons présenté l'ensemble des choix techniques et méthodologiques qui ont permis de concevoir l'architecture du système de surveillance des paramètres vitaux. On a justifié le choix des composants matériels (ESP32, MAX30102) et logiciels (Firebase, Flutter, Python) intégrant un modèle d'IA de type Random Forest pour la prédiction du risque diabète de type 2. On a mené une attention particulière sur l'estimation théorique des paramètres non mesurés directement, dans le but de permettre la continuité fonctionnelle du système malgré certaines limitations matérielles.

Le chapitre suivant sera dédié à la mise en œuvre pratique de cette architecture, au travers du développement du firmware, de l'application mobile, et les tests de validation du système.

Chapitre 3 :

Développement et Tests

III.1 Préambule

Ce chapitre présente la réalisation du système. Il détaille le développement du firmware embarqué sur l'ESP32, la création de l'application mobile, la configuration de la base de données Firebase ainsi que les tests réalisés pour valider le bon fonctionnement de l'ensemble. Enfin, l'intégration du modèle d'intelligence artificielle et les limites rencontrées seront abordées.

III.2 Bloc matériel du système :

Comme nous l'avons vu précédemment, le capteur MAX30102 dispose de quatre broches principales : VIN, GND, SCL et SDA.

VIN, GND : Le capteur peut être alimenté avec une tension de 3.3 V. Dans notre projet, l'alimentation a été fournie directement depuis la broche 3.3 V de l'ESP32, tandis que la masse (GND) est connectée à la masse du microcontrôleur pour assurer une référence commune.

SCL, SDA : Ces deux broches sont utilisées pour la communication I2C.

- SCL (Serial Clock) est reliée à la broche GPIO 22 de l'ESP32.
- SDA (Serial Data) est connectée à la broche GPIO 21 du microcontrôleur.

(Voir la figure (21) et tableau (2)).

Grâce à cette configuration, le capteur peut transmettre en continu les données physiologiques (signaux PPG) à l'ESP32 via le protocole I2C pour traitement.

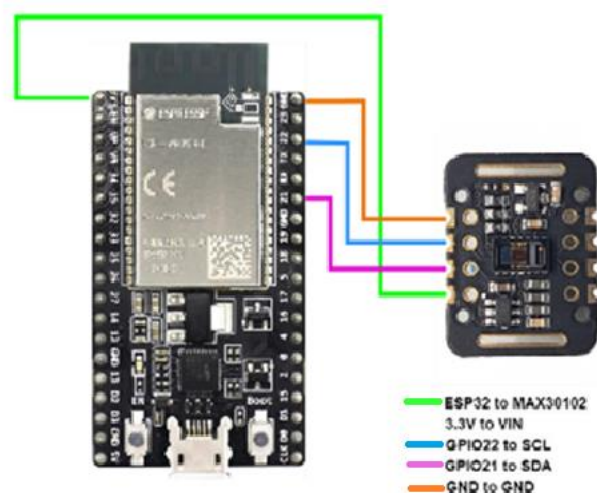


Figure 23. Schéma de branchement de MAX30102 à l'ESP32

Broche MAX30102	ESP32	Description
VIN	3.3V	Alimentation
GND	GND	Masse
SCL	GPIO 22	Horloge(I2C)
SDA	GPIO 21	Données(I2C)

Tableau 2. Tableau du branchement

III.3 Organigramme du code arduino :

- Connexion Wifi : Le module ESP32 se connecte au réseau Wifi défini dans le code.

- Initialisation du capteur : Le capteur MAX30105 est configuré pour mesurer le signal infrarouge.

- Lecture du signal IR : Le capteur lit en continu la valeur de lumière infrarouge réfléchiée par le doigt.

- Détection de battement : Si un battement est détecté, le BPM est calculé à partir du temps écoulé depuis le précédent.

- Calcul de la moyenne : Une moyenne mobile des 4 dernières valeurs de BPM est effectuée (beatAvg).

- Condition d'envoi : Si la moyenne est supérieure à 30 BPM et que 15 secondes sont passées depuis le dernier envoi, la valeur est prête à être transmise.

- Envoi vers Firestore : Une requête PATCH contenant la donnée hr est envoyée à Firestore via HTTP.

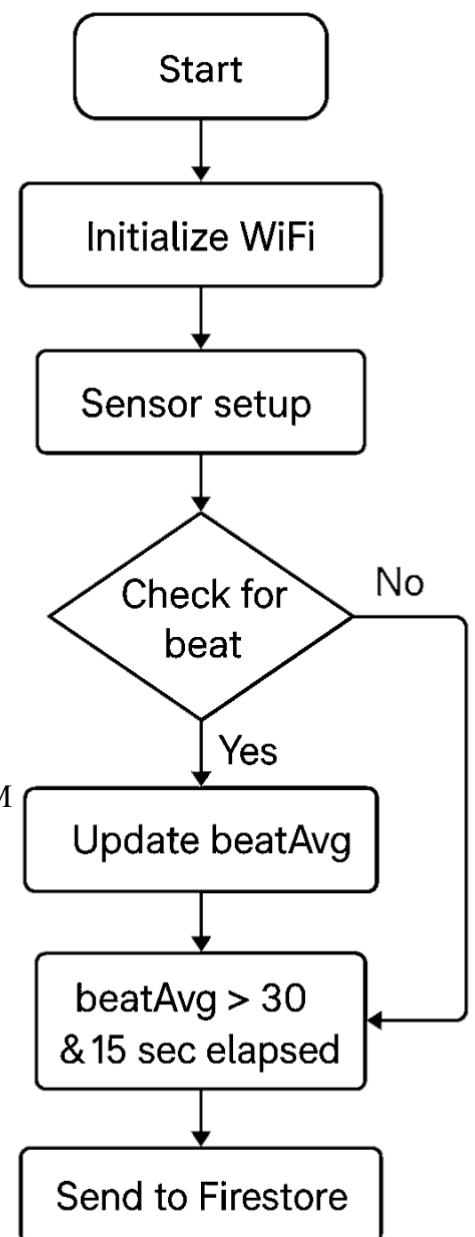


Figure 24. Organigramme du code arduino

III.4 Mise en œuvre du Firebase :

Pour assurer la communication entre le capteur (via l'ESP32) et l'application Flutter, Firebase est utilisé comme solution de base de données en temps réel. Sa compatibilité avec les plateformes mobiles et embarquées, sa simplicité d'intégration et ses fonctionnalités cloud en font un choix idéal pour une architecture légère et interactive.

III.4.1 Création du projet via la console Firebase :

Dans Firebase console :

Étape 1 : Création d'un nouveau projet

Pour créer notre projet sur Firebase

- ✓ On clique sur "**créer un projet Firebase**"
- ✓ On a ensuite donné un nom à notre projet (my project).
- ✓ On a désactivé Google Analytics (facultatif), puis nous avons validé la création.

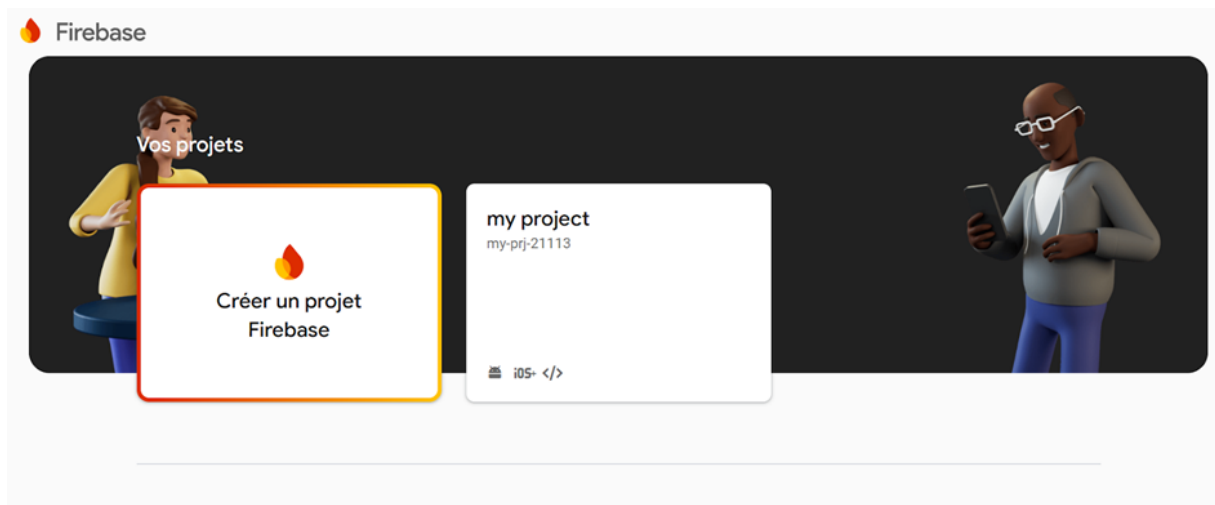
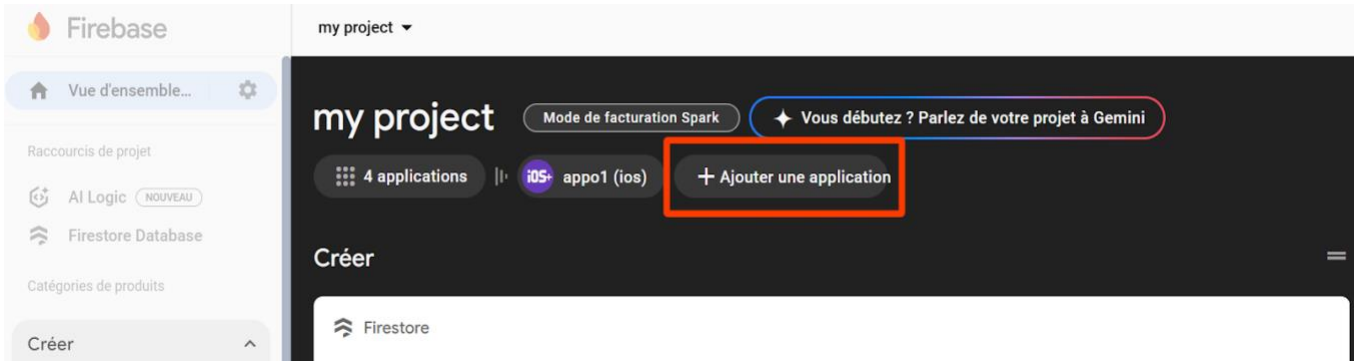


Figure 25 .Création d'un projet Firebase

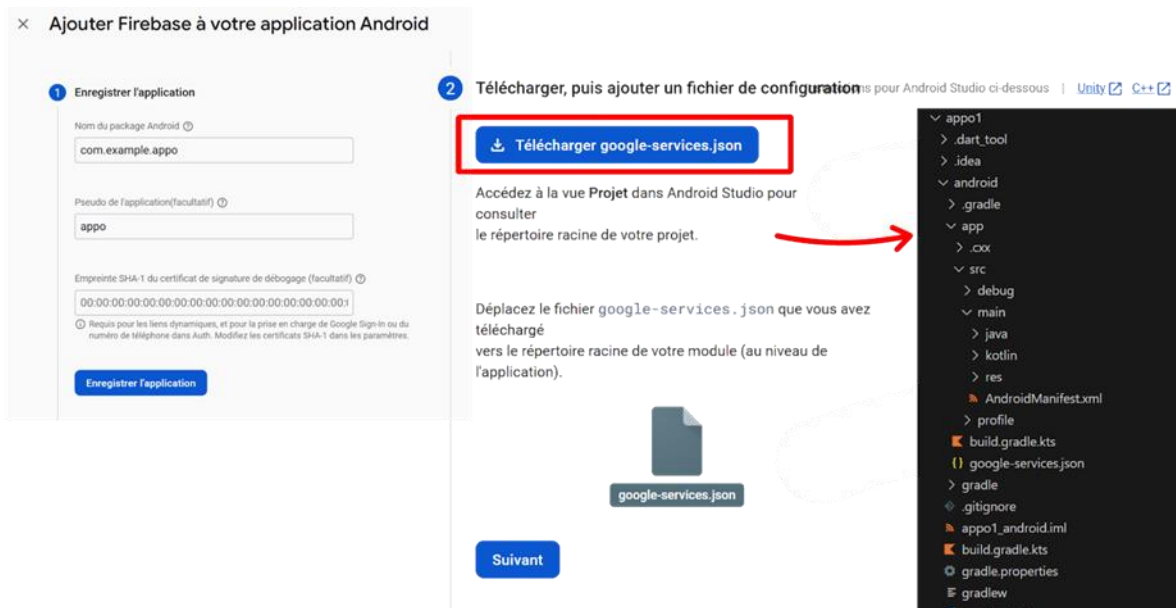
Étape 2 : Ajout d'une application Android

Une fois que notre projet a été créé :

- ✓ On clique sur "**Ajouter une application**" (icône Android).



- ✓ On a renseigné le nom du package Flutter
- ✓ On a laissé les champs facultatifs vides, puis cliqué sur "**Enregistrer l'application**".



- ✓ Le fichier **google-services.json** a été téléchargé, puis placé dans le projet flutter dans `android/app/google-services.json`

Étape 3 : Configuration dans Flutter

- ✓ On a ajouté les dépendances dans **pubspec.yaml** :

! pubspec.yaml

```

30 dependencies:
31   http: ^0.13.6
32   firebase_core: ^2.17.0
33   firebase_database: ^10.2.0
34   cloud_firestore: ^4.0.0

```

- ✓ Dans **android/build.gradle**, on ajoute :

android >  build.gradle.kts

```

1  buildscript {
2    repositories {
3      google()
4      mavenCentral()
5    }
6    dependencies {
7      classpath("com.android.tools.build:gradle:7.4.1")
8      classpath("com.google.gms:google-services:4.4.1")
9    }

```

- ✓ Dans **android/app/build.gradle** :

android > app >  build.gradle.kts

```

1  plugins {
2    id("com.android.application")
3    // START: FlutterFire Configuration
4    id("com.google.gms.google-services")
5    // END: FlutterFire Configuration
6    id("kotlin-android")
7    // The Flutter Gradle Plugin must be applied after the Android and Kotlin Gradle plugins.
8    id("dev.flutter.flutter-gradle-plugin")
9  }

```

- ✓ Dans **main.dart**, on initialise Firebase :

lib >  main.dart >  MyApp >  MyApp

```

8  void main() async {
14   // Initialize Firebase
15   try { // Catch any errors during Firebase initialization
16     await Firebase.initializeApp(
17       options: DefaultFirebaseOptions.currentPlatform,
18     );
19   } catch (e) {
20     debugPrint('Firebase init error: $e');
21   }
22
23   runApp(const MyApp());
24 }

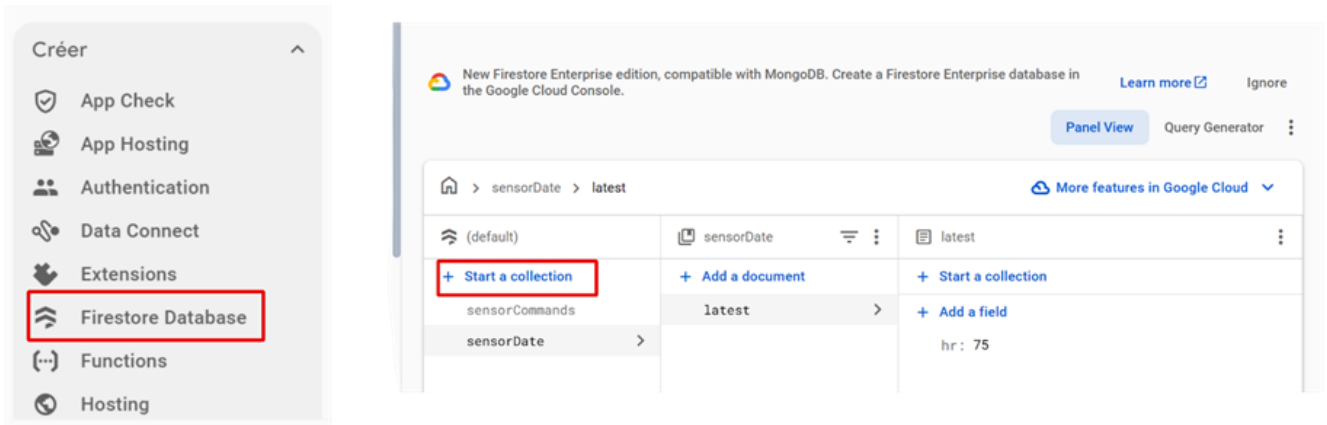
```

Étape 4: Activer Firestore (Cloud Firestore)

Dans la console Firebase, on va dans le menu de gauche :

Build → Firestore Database

- ✓ On clique sur **Créer une base de données.**



- ✓ Choisis le mode test pour démarrer rapidement et modifier les règles.

```
lib > pages > home_page.dart > _HomePageState
18 class _HomePageState extends State<HomePage> {
69   Future<int?> fetchHeartRate() async {
70     try {
71       await FirebaseFirestore.instance
72         .collection('sensorCommands')
73         .doc('startMeasurement')
74         .set({'start': true, 'timestamp': FieldValue.serverTimestamp()});
75     }
76     final snapshot = await FirebaseFirestore.instance
77       .collection('sensorDate')
78       .doc('latest')
79       .get();

```

- ✓ Interaction avec Firebase dans le code Flutter.

```
1 service cloud.firestore {
2   match /databases/{database}/documents {
3     match /{document=**} {
4       allow read, write: if true;
5     }
6   }
7 }
```

Ce code envoie une commande à Firebase pour démarrer la mesure, puis lit les données du capteur enregistrées par l'ESP32 et les affiche dans l'application.

Étape 5 : Traitement des commandes et envoi des données avec l'ESP32 :

Ce code sur l'ESP32 lit la commande startMeasurement depuis Firebase pour détecter une demande de mesure.

```

12 // Firestore
13 const char* PROJECT_ID = "my-prj-21113";
14 const char* API_KEY = "AIzaSyDBel8YHgk6y5iewbqfoeVcnrbJCFauAH4";
15 const char* BASE_PATH = "https://firestore.googleapis.com/v1/
16 projects/my-prj-21113/databases/(default)/documents";

```

Si la commande est active, le microcontrôleur lit les données du capteur puis les envoie vers Firebase, dans le document sensorDate/latest.

III.5 Développement du modèle d'intelligence artificielle

Dans cette partie, nous présentons les étapes de développement du modèle d'intelligence artificielle utilisé pour prédire le risque du diabète de type 2. Le modèle choisi est un Random Forest Classifier, entraîné à partir d'un jeu de données médicales.

III.5.1 Etape de développement :**Étape 1** : Chargement et préparation initiale des données

Ce bloc initialise l'environnement de travail avec les bibliothèques nécessaires (pandas, sklearn.) et lit le jeu de données à partir d'un fichier CSV.

Ensuite, les données sont nettoyées :

CODE

```

[72]: import pandas as pd
      from sklearn.metrics import classification_report, confusion_matrix
      from sklearn.ensemble import RandomForestClassifier
      from sklearn.metrics import classification_report
      from sklearn.utils import resample

[74]: df = pd.read_csv('/Users/Merina/Downloads/heart_attack_risk_dataset (1).csv')

[75]: df.dropna(how='all');

[76]: df.shape

[76]: (50000, 9)

[77]: df = df.drop(['Chest_Pain_Type', 'ECG_Results'], axis=1, errors='ignore')
      df['Gender'] = df['Gender'].map({'Male': 2, 'Female': 1})
      df['Stress_Level'] = df['Stress_Level'].map({'Low': 1, 'Moderate': 3, 'High': 3})

```

- Suppression des lignes totalement vides.
- Élimination de certaines colonnes non pertinentes.
- Encodage des variables catégorielles en valeurs numériques pour permettre leur traitement par les algorithmes de Machine Learning.

Étape 2 : Visualisation et compréhension du jeu de données

Cette partie permet une première inspection des données :

DATA

```
[78]: df.head()
```

```
[78]:
```

	Age	Gender	BMI	Heart_Rate	Family_History	Stress_Level	Diabetes
0	69	1	34.61	85	0	3	1
1	32	2	22.75	103	0	1	0
2	89	2	35.32	127	0	1	0
3	78	2	18.23	125	0	1	1
4	38	1	19.82	107	0	3	0

```
[79]: df['Diabetes'].value_counts()
```

```
[79]: Diabetes
0    40058
1     9942
Name: count, dtype: int64
```

- ✓ df.head () affiche les premières lignes pour avoir une vue rapide de la structure.
- ✓ df ['Diabetes'].value_counts() indique combien de cas positifs et négatifs sont présents, ce qui permet de détecter un éventuel déséquilibre entre les classes.

Étape 3 : Séparation des variables et division du Dataset :

On sépare les variables explicatives (features) de la variable cible (Diabetes), puis on divise le jeu de données en deux sous-ensembles :

- Un pour l'entraînement.
- Un pour le test.

L'utilisation de stratify=y garantit que les proportions de classes (diabétiques / non diabétiques) sont conservées dans les deux sous-ensembles.

Étape 4 : Entraînement du modèle Random Forest :

On instancie un classifieur de type Random Forest sans réglages particuliers, puis on l’entraîne sur les données d’apprentissage.

TRAINING

```
[81]: X = df.drop('Diabetes', axis=1)
      y = df['Diabetes']

[82]: from sklearn.model_selection import train_test_split

[83]: X_train ,X_test ,y_train ,y_test = train_test_split(X,y,random_state=42,test_size=0.12,stratify=y)

[84]: rf = RandomForestClassifier()

[85]: rf.fit(X_train, y_train)

[85]: RandomForestClassifier
      RandomForestClassifier()

[86]: y_pred = rf.predict(X_test)
```

Figure 26. Entraînement du modèle Random Forest

Ensuite, on utilise ce modèle pour prédire les classes sur le jeu de test.

Ce premier entraînement permet d’avoir une idée générale de la capacité du modèle à traiter les données.

Étape 5 : Évaluation à l’aide d’une matrice de confusion

On évalue les performances du modèle à l’aide d’une matrice de confusion, qui affiche le nombre de vraies prédictions et d’erreurs de classification.

confusion matrix

```
[87]: from sklearn.metrics import confusion_matrix
      import seaborn as sns
      import matplotlib.pyplot as plt

      cm = confusion_matrix(y_test, y_pred)
      sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')
      plt.xlabel('Predicted')
      plt.ylabel('Actual')
      plt.title('Confusion Matrix')
      plt.show()
```

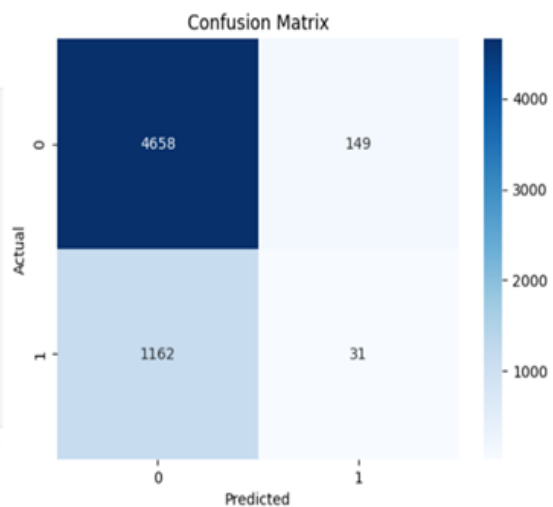


Figure 27. Matrice de confusion

La matrice est affichée sous forme graphique à l'aide de Seaborn, ce qui permet d'avoir une lecture rapide des résultats (vrais positifs, faux positifs).

Étape 6 : Calcul de la précision du modèle :

ACCURACY

```
[88]: rf.score(X_test,y_test)
```

```
[88]: 0.7815
```

Cette métrique donne un aperçu rapide de la performance générale du classifieur.

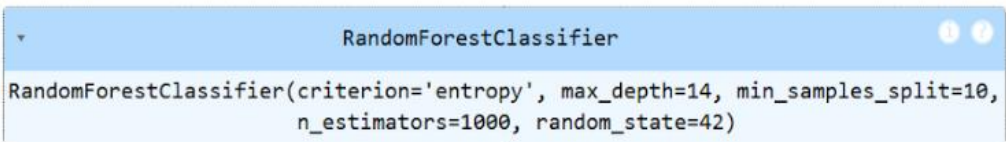
Étape 7 : Optimisation du modèle par réglage des hyper-paramètres :

Un deuxième modèle est construit, cette fois avec des hyper-paramètres ajustés pour améliorer la précision et la robustesse du modèle :

HYPERPARAMETERS

```
[89]: rf2 = RandomForestClassifier(n_estimators = 1000,
                                criterion = 'entropy',
                                min_samples_split = 10,
                                max_depth = 14,
                                random_state = 42
                                )
```

```
[90]: rf2.fit(X_train, y_train)
```

```
[90]: 
      RandomForestClassifier
      RandomForestClassifier(criterion='entropy', max_depth=14, min_samples_split=10,
                            n_estimators=1000, random_state=42)
```

ACCURACY

```
[91]: rf2.score(X_test, y_test)
```

```
[91]: 0.8011666666666667
```

Ces réglages permettent de mieux contrôler la complexité du modèle et de réduire le risque de sur-apprentissage.

III.5.2 Analyse des Résultats :

Nous avons évalué les performances de trois modèles différents, tous appliqués au même jeu de données.

L'objectif était de mesurer l'impact du déséquilibre des classes, du rééquilibrage, et du réglage des hyper-paramètres sur la détection des cas de diabète.

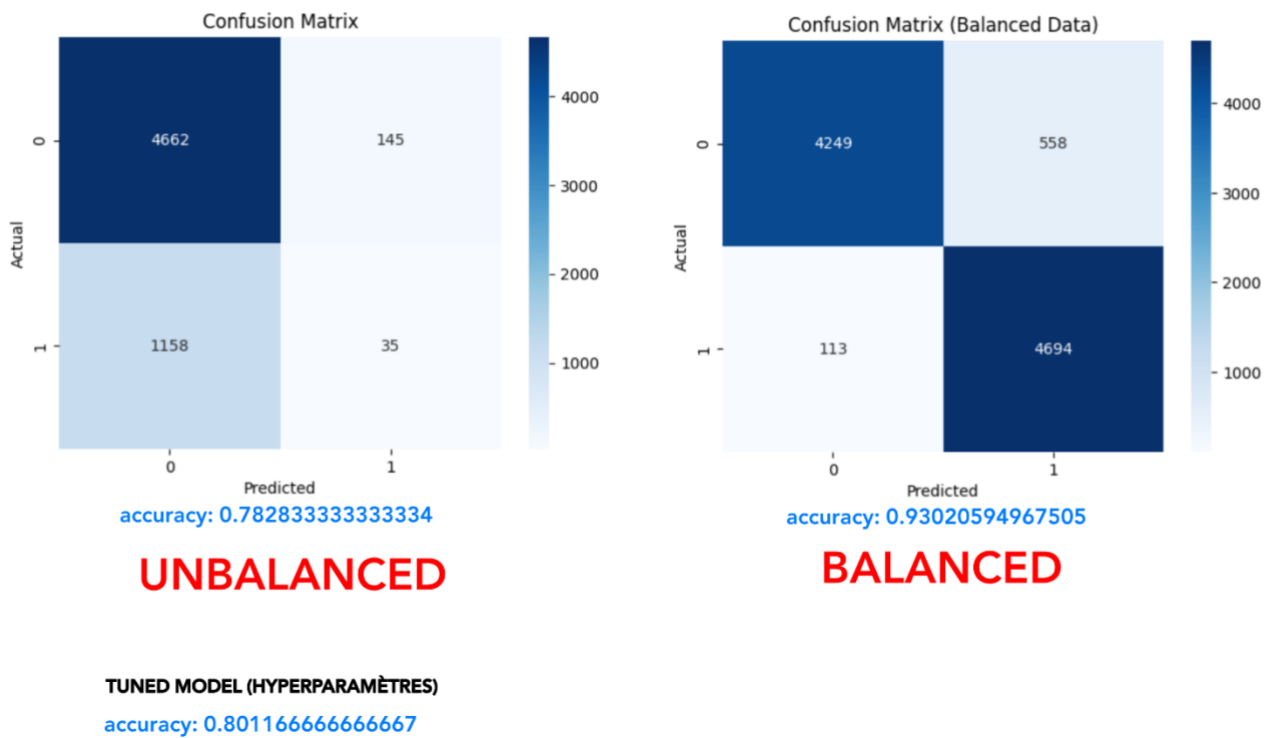


Figure 28. Comparaison des résultats

Chaque matrice de confusion montre :

- En haut à gauche : cas non diabétiques correctement identifiés (True Negatives)
- En bas à droite : cas diabétiques correctement détectés (True Positives)
- En haut à droite : faux positifs
- En bas à gauche : faux négatifs (les plus critiques médicalement)

Modèle avec données déséquilibrées (unbalanced)

Entraîné sur les données brutes (classe 0 largement majoritaire).

Résultat : très faible détection des cas positifs (classe 1).

La précision globale semble correcte, mais le modèle ignore presque totalement les cas de diabète.

Modèle avec données équilibrées (balanced)

Sur-échantillonnage de la classe 1 pour équilibrer les données.

Résultat : forte amélioration de la sensibilité (vrais cas positifs détectés).

Ce modèle offre le meilleur équilibre entre précision et utilité médicale.

Modèle équilibré avec hyper-paramètres

Random Forest ajusté (n_estimators, max_depth, etc.).

Résultat : performance inférieure au modèle de base.

Montre qu'un bon équilibrage est souvent plus efficace qu'un réglage complexe.

❖ Le choix final :

Nous avons choisi le modèle Random Forest, entraîné sur les données équilibrées du dataset, car il offre la meilleure précision.

De plus, selon la matrice de confusion, il parvient à prédire correctement la majorité des vrais positifs et vrais négatifs, ce qui en fait le modèle le plus adapté à notre système.

III.6 Application Flutter

L'application mobile a été développée avec Flutter (Dart) et constitue l'interface principale entre l'utilisateur et les composants du système.

Elle permet d'enregistrer des utilisateurs avec leurs données biométriques, de récupérer en temps réel la fréquence cardiaque depuis le capteur, de stocker localement les données via Hive, et d'évaluer le risque de diabète grâce à un modèle d'intelligence artificielle intégré.

III.6.1 Interface Utilisateur

L'interface a été pensée pour être directe, fluide et facile à prendre en main. Voici les principales étapes de navigation dans l'application :

Page 1 : page d'accueil :

Dès l'ouverture de l'application, l'utilisateur arrive sur une page d'accueil affichant la liste des personnes enregistrées, sous forme de tuiles (tiles). Chaque tuile affiche le nom de l'utilisateur, avec deux actions possibles :



Figure 29. Page d'accueil de l'application

- Supprimer une fiche utilisateur.
- Lancer l'évaluation du risque via un bouton "Check".

Page 2 : Ajout d'un utilisateur + Récupération de la fréquence cardiaque :

En cliquant sur le bouton '+', une boîte de dialogue propose de lancer une mesure du capteur (fréquence cardiaque via Firebase/ESP32). Ensuite, une deuxième boîte permet à l'utilisateur de renseigner :

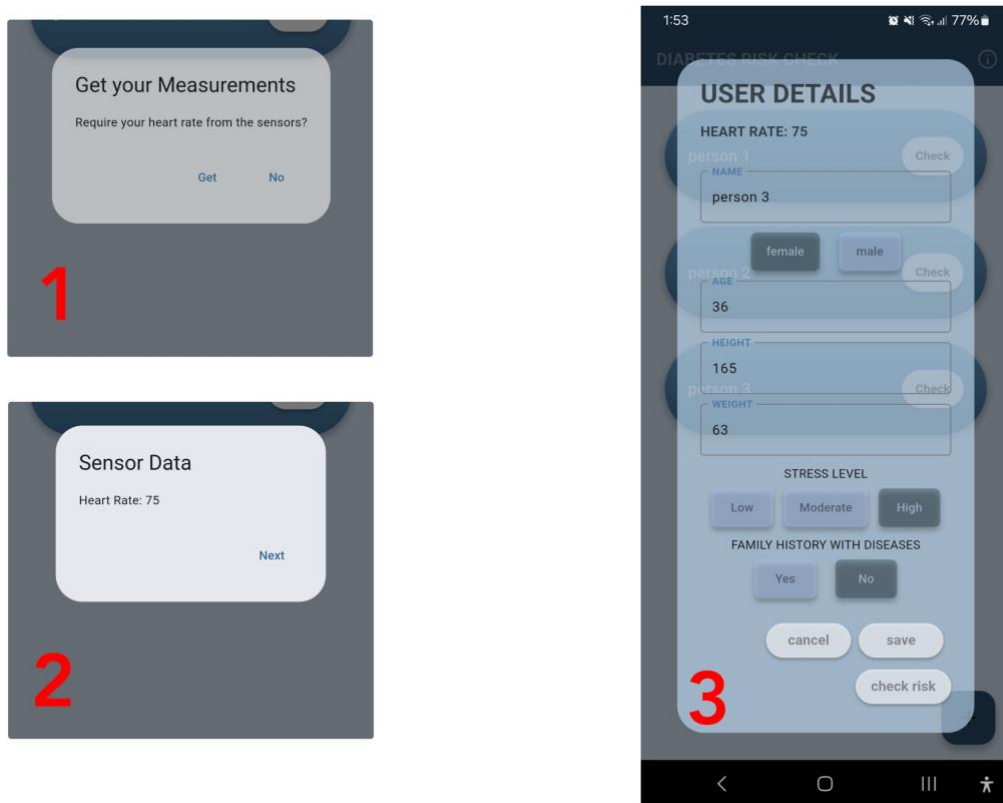


Figure 30. Page d'utilisateur

Page 3 : Résultat IA (Évaluation du risque)

Une fois la fiche enregistrée, l'utilisateur peut appuyer sur "Check" pour lancer une prédiction via le modèle IA.



Figure 31. Affichage de résultat dans l'application

III.6.2 Back-end de l'application

L'architecture logicielle de l'application est répartie en plusieurs composants Dart, chacun jouant un rôle précis dans le fonctionnement interne. Voici comment ils s'articulent :

➤ **Initialisation de l'application :**

Ce bloc configure l'environnement dès le lancement de l'application. Il initialise Firebase, Hive (pour la base de données locale), et charge les données initiales.

Correspondance : main.dart

➤ **Gestion de l'interface principale et logique utilisateur :**

C'est ici que toute l'interaction utilisateur prend forme :

- Affichage de la liste des utilisateurs enregistrés,
- Suppression/modification d'un profil,
- Ouverture des dialogues de saisie,
- Récupération de la fréquence cardiaque via Firebase,
- Déclenchement de l'évaluation par l'IA.

Correspondance : homepage.dart

➤ **Saisie des données utilisateur :**

Ce composant affiche le formulaire de saisie (ou modification) des données utilisateur. Il contient :

- Les champs pour nom, âge, sexe, taille, poids,
- La sélection du niveau de stress et des antécédents médicaux,
- Le calcul automatique de l'IMC.

Correspondance : dialogue.dart

➤ **Organisation et persistance des données :**

Ce module gère le stockage local des profils utilisateurs via Hive. Il assure la sauvegarde, la récupération et la mise à jour des données dans une box locale.

Correspondance : databasebox.dart

➤ **Composants utilitaires (UI) :**

Plusieurs fichiers définissent des éléments d'interface réutilisables pour éviter la redondance :

- Boutons stylisés,
- Tuiles représentant chaque utilisateur,
- Actions rapides (bouton d'analyse IA, suppression...).

Correspondence: primary_button .dart, secondary_button.dart, tile.dart

III.7 Intégration du modèle IA

L'intégration du modèle d'intelligence artificielle dans l'application s'est faite en quatre étapes principales, en passant par une API intermédiaire développée avec Flask.

Étape 1 : Entraînement et exportation du modèle :

Après évaluation du model, il a été exporté en fichier .pkl (pickle) avec joblib.

```
[112]: import joblib  
  
joblib.dump(rf2, 'diabetes_rf_model.pkl')
```

```
[112]: ['diabetes_rf_model.pkl']
```

Étape 2 : Création de l'API Flask

Un petit serveur Flask (app.py) a été développé pour charger le modèle et exposer un endpoint /predict acceptant des requêtes POST avec les données du patient. Il retourne le résultat du modèle (0 ou 1) selon le risque de diabète.

```

model_api > app.py > predict
1  from flask import Flask, request, jsonify
2  import joblib
3  import numpy as np
4  from flask_cors import CORS
5
6  app = Flask(__name__)
7  CORS(app) # Allow cross-origin requests
8  #(important for Flutter to talk to Flask)
9  # Load the trained Random Forest model
10 model = joblib.load("diabetes_rf_model.pkl")
    
```

Étape 3 : Lancement de l'API localement

Le fichier app.py est exécuté sur la machine en local (via python app.py) ce qui lance un serveur à l'adresse http://127.0.0.1:5000. Ce serveur attend les requêtes depuis l'application mobile.

```

PS C:\Users\Merina\Desktop\appo\model_api> python app.py
* Serving Flask app 'app'
* Debug mode: on

* Running on http://127.0.0.1:5000 Adresse locale de l'API, utilisée dans l'application
Press CTRL+C to quit Flutter pour envoyer les données
* Restarting with stat
* Debugger is active!
* Debugger PIN: 117-300-697
    
```

Étape 4 : Communication entre Flutter et API

Depuis l'application Flutter, une fonction asynchrone a été créée pour envoyer une requête HTTP POST contenant les données du profil utilisateur au serveur Flask.

```

    appo1 > lib > pages > home_page.dart > _HomePageState > build
18  class _HomePageState extends State<HomePage> {
176 // AI connection
177 void runAICheck(int index) async {
178   final feature = db.features[index];
179
180   final inputData = {
181     "Gender": feature[4],
182     "Age": feature[1],
183     "Height": feature[2],
184     "Weight": feature[3],
185     "Stress_Level": feature[5],
186     "Family_History": feature[6],
187     "Heart_Rate": feature[7],
188     "BMI": feature[8],
189   };
190
191   final prediction = await PredictService().predictRisk(inputData);
192
193   _showAlert(
194     "Prediction Result",
195     prediction == 1 ? "⚠️ High Diabetes Risk" : "✅ Low Diabetes Risk",
196   );
197 }

```

Le résultat est ensuite affiché sous forme de popup à l'utilisateur.

III.8 Discussion :

Dans ce chapitre, nous avons introduit le processus de développement du prototype visant à évaluer le risque de diabète de type 2 à partir de la fréquence cardiaque.

À travers la réalisation du capteur MAX30102, la carte ESP32, la base de données Firebase, un modèle d'intelligence artificielle Random Forest, et une application mobile Flutter, nous avons réussi à valider la faisabilité de la chaîne complète : mesure, transmission, prédiction et affichage. Malgré que les paramètres HRV et PA n'aient pas été implémentés dans cette version du système, leur étude en phase de conception permet d'avancer vers une extension fonctionnelle et finale à l'avenir.

Conclusion

Notre travail vise à réaliser un prototype connecté, non invasif et facile à utiliser destiné à la prédiction du risque de diabète de type 2 en utilisant l'intelligence artificielle. Dans une première étape, nous avons énoncé les fondements théoriques du diabète, des paramètres vitaux (HR, HRV, PA) et des approches IA utilisés en médecine. Dans une seconde étape, nous avons proposé une architecture technique basé sur une carte ESP32 et l'intelligence artificielle (Random Forest). Nous avons développé une application mobile sous Flutter qui permet de piloter le système. De plus, nous avons réalisé l'implémentation d'un prototype fonctionnel capable de mesurer la fréquence cardiaque, de transmettre les données, de prédire le risque, d'afficher le résultat sur l'application mobile.

Bien que limité à la seule mesure de la fréquence cardiaque, notre prototype constitue une base solide pour envisager l'évolution vers un système plus abouti. Cependant, l'absence de traitement en temps réel, que ce soit dans la réception des données depuis le capteur ou dans la transmission du signal de l'application vers Firebase, constitue une limite importante. Cela est principalement dû à la dépendance au Wi-Fi pour la transmission. Par conséquent, la mesure de la fréquence cardiaque met un certain temps à être transmise, ce qui rend le processus moins fluide.

Nous pouvons également intégrer d'autres fonctionnalités afin d'aboutir à une solution plus complète, par exemple :

- Intégration de la variabilité de la fréquence cardiaque (HRV) et de la pression artérielle (PA) pour enrichir la précision du modèle.
- Ajout de capteurs non invasifs permettant d'estimer la glycémie sans piqûre.
- Entraînement du modèle IA sur une base de données réelle (datasets) pour une meilleure robustesse.

Notre projet ouvre un chemin vers la création d'un dispositif intelligent, évolutif et accessible pour la prévention du diabète, qui pourrait à terme contribuer à une prise en charge précocement, personnalisée et efficace de cette pathologie chronique.

Bibliographie

1. Organisation mondiale de la Santé. *Diabète*. [En ligne] 2024. [Citation : 15 Mars 2025.] <https://www.who.int/fr/news-room/fact-sheets/detail/diabetes>.
2. La Fédération Française des Diabétiques. *Qu'est-ce que le diabète ?* [En ligne] [Citation : 15 Mars 2025.] <https://www.federationdesdiabetiques.org/information/diabete>.
3. Vidal. *Diabète de type 2*. [En ligne] 06 Mai 2021. [Citation : 15 Mars 2025.] <https://www.vidal.fr/maladies/metabolisme-diabete/diabete-type-2.html>.
4. JOURNAL DES FEMMES. *Diabète de type 1 : symptômes, âge, espérance de vie*. [En ligne] [Citation : 15 Mars 2025.]
5. JOURNAL DES FEMMES. *Diabète de type 2 : signes, cause, traitement, héréditaire ?* [En ligne] [Citation : 15 Mars 2025.]
6. PMC . *Impact of Diabetes on Cardiovascular Disease*. [En ligne] [Citation : 22 Mai 2025.] <https://pmc.ncbi.nlm.nih.gov/articles/PMC3603160/>.
7. LE MANUEL MSD. *Présentation des troubles du rythme cardiaque*. [En ligne] [Citation : 26 Mars 2025.] <https://www.msmanuals.com/fr/accueil/troubles-cardiaques-et-vasculaires/troubles-du-rythme-cardiaque/pr%C3%A9sentation-des-troubles-du-rythme-cardiaque>.
8. PubMed. *Reste cardiaque et risque de diabète de type 2: une étude de cohorte prospective et une méta-analyse*. [En ligne] <https://pubmed.ncbi.nlm.nih.gov/30378246/>.
9. PubMed. *Temporal Changes in Resting Heart Rate and Risk of Diabetes Mellitus*. [En ligne] <https://pubmed.ncbi.nlm.nih.gov/38310876/>.
10. Wikipédia l'encyclopédie libre. *Variabilité de la fréquence cardiaque*. [En ligne] [https://fr.wikipedia.org/wiki/Variabilit%C3%A9_de_la_fr%C3%A9quence_cardiaque#:~:text=La%20variabilit%C3%A9%20de%20la%20fr%C3%A9quence%20cardiaque%20\(VFC\)%20correspond%20%C3%A0%20la,externe%20de%20la%20fr%C3%A9quence%20cardiaque..](https://fr.wikipedia.org/wiki/Variabilit%C3%A9_de_la_fr%C3%A9quence_cardiaque#:~:text=La%20variabilit%C3%A9%20de%20la%20fr%C3%A9quence%20cardiaque%20(VFC)%20correspond%20%C3%A0%20la,externe%20de%20la%20fr%C3%A9quence%20cardiaque..)
11. PubMed. *Decreased heart rate variability as a predictor for diabetes-A prospective study of the Brazilian longitudinal study of adult health*. [En ligne] <https://pubmed.ncbi.nlm.nih.gov/31062488/>.
12. [En ligne] <https://pmc.ncbi.nlm.nih.gov/articles/PMC10505535/#dgad200-s7>.
13. Assia., Nait Si Mohand Assma et Nait Si Mohand. *Réalisation d'un dispositif dédié à la mesure des*. 2018/2019.
14. ViDAL. *la tension artérielle* . [En ligne] [https://www.vidal.fr/maladies/coeur-circulation-veines/hypertension-arterielle.html#:~:text=La%20pression%20art%C3%A9rielle%20\(ou%20tension,est%20%C3%A9valu%C3%A9e%20par%20deux%20valeurs..](https://www.vidal.fr/maladies/coeur-circulation-veines/hypertension-arterielle.html#:~:text=La%20pression%20art%C3%A9rielle%20(ou%20tension,est%20%C3%A9valu%C3%A9e%20par%20deux%20valeurs..)

15. Women's Health Study. *Blood pressure and risk of developing type 2 diabetes mellitus*. [En ligne] 2007. <https://pubmed.ncbi.nlm.nih.gov/17925342/>.
16. Diabetes.co.uk. *Obesity and Iabetes*. [En ligne] <https://www.diabetes.co.uk/diabetes-and-obesity.html#:~:text=This%20condition%2C%20known%20as%20insulin,with%20a%20BM I%20under%2022..>
17. La Fédération Française des Diabétiques. *Stress et diabète de type 2 : un lien confirmé*. [En ligne] [Citation : 11 Juin 2025.] <https://www.federationdesdiabetiques.org/information/recherche-innovations-diabete/actualites/stress-et-diabete-de-type-2-un-lien-confirme#:~:text=Les%20scientifiques%20ont%20ainsi%20pu,statut%20ou%20l'inactivit%C3%A9%20physique..>
18. Assma, : Nait Si Mohand. *Réalisation d'un dispositif dédié à la mesure des*. 2019.
19. Meriem, Bennaoum. *MESURE INDIRECTE DE LA TENSION*. 2017.
20. [En ligne] <https://www.fmfpro.org/lia-en-medecine-revolution-ou-evolution/>.
21. Inserm. *Intelligence artificielle : va-t-elle remplacer le diagnostic médical ?* [En ligne] 13 Juillet 2023. <https://www.inserm.fr/actualite/intelligence-artificielle-va-t-elle-remplacer-le-diagnostic-medical/#:~:text=En%20radiologie%2C%20en%20dermatologie%2C%20ou,orienter%20plus%20rapidement%20les%20patients..>
22. *L'INTELLIGENCE ARTIFICIELLE DANS LA PRÉDICTION ET LA PRÉVENTION DU DIABÈTE*. [En ligne] 26 Decembre 2024. <https://www.le-diabete-dans-tous-ses-etats.precidiab.org/series/lintelligence-artificielle-dans-la-prediction-et-la-prevention-du-diabete/>.
23. postDICOM. *avantage de l'intelligence artificiel*. [En ligne] <https://www.postdicom.com/fr/blog/benefits-of-artificial-intelligence-in-healthcare#:~:text=L'un%20des%20avantages%20de,le%20nombre%20%C3%A9lev%C3%A9%20de%20cas..>
24. Wikipédia l'encyclopédie libre. *ESP32*. [En ligne] [Citation : 05 Mai 2025.] <https://fr.wikipedia.org/wiki/ESP32#:~:text=ESP32%20est%20une%20s%C3%A9rie%20de,est%20une%20%C3%A9volution%20d'ESP8266..>
25. Bouhouch, Hamza. MOUSSASOFT. *Comment utiliser MAX30102 module capteur d'Oxygène et rythme cardiaque avec Arduino*. [En ligne] 10 Juillet 2023. [Citation : 19 Mai 2025.] <https://www.moussasoft.com/max30102-module-capteur-avec-arduino/>.
26. Campbell, Scott. *Circuit Basics. Basics of the I2C Communication Protocol*. [En ligne] [Citation : 19 Mai 2025.] <https://www.circuitbasics.com/basics-of-the-i2c-communication-protocol/>.
27. firebase. *Cloud Firestore*. [En ligne] [Citation : 22 Mai 2025.] <https://firebase.google.com/docs/firestore?hl=fr>.
28. Wikipédia l'encyclopédie libre. *Flutter*. [En ligne] [Citation : 22 Mai 2025.] [https://fr.wikipedia.org/wiki/Flutter_\(logiciel\)](https://fr.wikipedia.org/wiki/Flutter_(logiciel)).

29. [En ligne] [Citation : 30 Mai 2025.] <https://arduino.blaiseascal.fr/presentation/logiciel/>.
30. blent. *Random Forest : comment ça fonctionne ?* [En ligne] [Citation : 2 mai 2025.] <https://blent.ai/blog/a/random-forest-comment-ca-marche>.
31. Medium. *Random Forest In-depth*. [En ligne] [Citation : 5 juin 2025.] <https://medium.com/@fraidoonomarzai99/random-forest-in-depth-f0556817c40b>.
32. Wikipédia|encyclopédie libre. *Python*. [En ligne] [https://fr.wikipedia.org/wiki/Python_\(langage\)](https://fr.wikipedia.org/wiki/Python_(langage)).

Accès aux Fichiers du Projet sous Github :

Dans ce référentiel GitHub :

https://github.com/merinZen/PFE_diabetes_prediction

Nous avons regroupé tous les fichiers liés à notre système de prédiction du diabète.

- Application Flutter

Dossier app01/ : contient l'application mobile permettant de mesurer la fréquence cardiaque, saisir les données utilisateur et afficher le résultat de la prédiction.

- Code Arduino

Dossier arduino/ : code destiné à l'ESP32 pour lire la fréquence cardiaque avec le capteur MAX30105 et communiquer avec Firebase.

- API Flask

Dossier model_api/ : API de prédiction utilisant un modèle Random Forest. Le fichier .joblib n'est pas inclus (taille >100 Mo) mais peut être ajouté manuellement.

- Entraînement du Modèle

Dossier model_training/ : contient un notebook Jupyter avec les étapes de préparation des données, l'entraînement et l'export du modèle.

- Jeu de Données

Fichier diabetes_dataset.csv : données médicales utilisées pour entraîner le modèle (glucose, BMI, âge, etc.).

GitHub permet de centraliser le code, suivre les modifications, collaborer efficacement et partager toutes les ressources du projet. Le lien ci-dessus donne accès à l'ensemble des fichiers nécessaires à la reproduction et à l'analyse de notre système.