**Résumé**

L'objectif de notre projet est de concevoir et implémenter une solution pour l’authentification biométrique d’une personne avec un signal électrocardiogramme (ECG). Cette solution est sous forme d’un dispositif portable qui permet à l’utilisateur de s’authentifier à travers une application mobile.

En fait, le principe consiste à acquérir d’abord le signal cardiaque ECG mesuré par un capteur d'électrocardiographie. Ensuite appliquer un filtrage sur ce signal afin d’extraire les points caractéristiques uniques à chaque personne. Ses caractéristiques sont fournies pour un modèle d’apprentissage automatique. Dans cette partie plusieurs algorithmes ont testé sur notre jeu de données pour choisir un algorithme avec un meilleur résultat de prédiction. Une dernière étape de déploiement du modèle d’apprentissage automatique pour rendre ce dernier en mode production et utiliser par le grand public.

**Mots clés :** Traitement de signal, Bluetooth, Machine Learning, Python, Flask, Android

**Authentification par une signature cardiaque ECG**

**Abstarct**

In a general way, biometrics at the goal, from their biological characteristics. This practice is a replacement for the method of verification of individual identification; among other things, passwords and security codes. In everyday life, biometrics have wide applications and the search for new biometric methods is timely. There is personal evidence that electrical heart activity incorporates very unique differences in heart morphology in individuals. The purpose of this project is to design a biometric embedded system that is used to authenticate a person using his or her heart signal. In fact, the principle is to acquire the cardiac signal first, then to apply signal processing operations allows extracting unique descriptors to each persoson authentication.

**Dédicace**

Je dédie ce travail à :

A ma mère,

A mon père,

A mes frères et sœurs,

A toute ma famille,

A tous mes enseignants et mes collègues à l’école ESPRIT.

A tous ceux que j'aime

Qu'ils trouvent ici l'expression de toute ma reconnaissance.

**Remerciements**

**2018 - 2019**

**2018 - 2019**

Je veux profiter de cette occasion, pour remercier toutes les personnes qui ont contribué au succès de mon stage. En premier lieu, je tiens à remercier avec un grand plaisir et un grand respect Mme. Jmal Marwa la responsable chez TELNET INNOVATION LABS, qui a répondu positivement à mon demande de stage, aussi pour l’intérêt et le conseil qu’elle a accordé à mon travail.

Je désire aussi remercier mon encadrant pédagogique Mr Maalej Mohamed, pour ses conseils au niveau de l’exécution du projet ainsi que sur la rédaction du rapport de fin de stage.

De même, je tiens à exprimer mes remerciements et toute mes gratitudes à Mme Othmen Farah mon encadreur dans la société pour son précieux soutient et sa disponibilité.

Avec un grand respect, Je remercie également toute l’équipe de TELNET pour leur accueil et leur esprit d'équipe.

**Authentification par une signature cardiaque ECG**

Table de matière

Introduction générale\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_8

[1 Chapitre 1 : Présentation du projet et du cahier de charges 1-10](#_Toc11763849)

[1.1 Introduction 1-10](#_Toc11763850)

[1.2 Contexte et objectifs du projet 1-10](#_Toc11763851)

[1.2.1 Entreprise d’accueil 1-10](#_Toc11763852)

[1.2.2 Cadre du projet et cahier de charges 1-11](#_Toc11763853)

[1.2.3 Etude de l’existant 1-12](#_Toc11763854)

[1.3 Solution proposée 1-13](#_Toc11763855)

[1.4 Etat de l’art 1-14](#_Toc11763856)

[1.4.1 Modalité biométrique et techniques d’authentification 1-14](#_Toc11763857)

[1.4.2 Apprentissage automatique 1-17](#_Toc11763858)

[1.4.3 Les algorithmes d’apprentissage automatique 1-18](#_Toc11763859)

[1.5 Conclusion 1-19](#_Toc11763860)

[2 Chapitre 2 : Analyse et conception 2-20](#_Toc11763861)

[2.1 Introduction 2-20](#_Toc11763862)

[2.2 Architecture global du projet 2-20](#_Toc11763863)

[2.2.1 Acquisition du signal ECG 2-21](#_Toc11763864)

[2.2.2 Filtrage et extraction des caractéristiques d’un signal ECG 2-22](#_Toc11763865)

[2.2.3 Construction du modèle d’apprentissage automatique 2-22](#_Toc11763866)

[2.3 Etude conceptuelle 2-25](#_Toc11763867)

[2.3.1 Identification des acteurs 2-25](#_Toc11763868)

[2.3.2 Besoin fonctionnel et non fonctionnel 2-25](#_Toc11763869)

[2.3.3 Diagramme de cas d’utilisation 2-26](#_Toc11763870)

[2.3.4 Diagramme de séquence 2-27](#_Toc11763871)

[2.3.5 Diagramme d’activité Connexion avec module de protocole de communication 2-28](#_Toc11763872)

[2.3.6 Diagramme de déploiement 2-28](#_Toc11763873)

[2.3.7 Diagramme classe 2-29](#_Toc11763874)

[2.4 Conclusion 2-29](#_Toc11763875)

[3 Chapitre 3 : Réalisation 3-30](#_Toc11763876)

[3.1 Introduction 3-30](#_Toc11763877)

[3.2 Etude de choix 3-30](#_Toc11763878)

[3.2.1 Choix du microcontrôleur 3-30](#_Toc11763879)

[3.2.2 Choix de capteur électrocardiogramme ECG 3-31](#_Toc11763880)

[3.2.3 Choix du protocole de la communication 3-32](#_Toc11763881)

[3.2.4 Choix de langage d’extraction de descripteurs du signal 3-33](#_Toc11763882)

[3.2.5 Comparaison des bibliothèques d’apprentissage automatique 3-35](#_Toc11763883)

[3.2.6 Comparaison des solutions pour le déploiement du modèle d’apprentissage automatique 3-36](#_Toc11763884)

[3.3 Environnement de développement 3-37](#_Toc11763885)

[3.3.1 Environnement matériel 3-38](#_Toc11763886)

[3.3.2 Environnement logiciel 3-38](#_Toc11763887)

[3.4 Réalisation théorique 3-41](#_Toc11763888)

[3.4.1 Choix des signaux 3-41](#_Toc11763889)

[3.4.2 Extraction des caractéristiques 3-41](#_Toc11763890)

[3.4.3 Le SVM pour l’apprentissage automatique 3-43](#_Toc11763891)

[3.5 Réalisation pratique 3-53](#_Toc11763892)

[3.5.1 Acquisition du signal 3-53](#_Toc11763893)

[3.5.2 Extraction et filtrage du signal 3-54](#_Toc11763894)

[3.5.3 Choix de l’algorithme d’apprentissage automatique 3-56](#_Toc11763895)

[3.5.4 Déploiement du model d’apprentissage automatique 3-58](#_Toc11763896)

[3.5.5 Développement de l’application Android 3-61](#_Toc11763897)

[3.6 Conclusion 3-66](#_Toc11763898)

[4 Chapitre 4 : Résultat 4-67](#_Toc11763899)

[4.1 Introduction 4-67](#_Toc11763900)

[4.2 Résultat théorique 4-67](#_Toc11763901)

[4.3 Résultats des signaux pratiques 4-71](#_Toc11763902)

[4.4 Conclusion 4-75](#_Toc11763903)

[5 Conclusion générale 5-76](#_Toc11763904)

[6 Netographie 6-78](#_Toc11763905)

[7 Annexe 7-82](#_Toc11763906)

Listes des figures

[Figure 1 : TELNET HOLDING 1-10](#_Toc11763907)

[Figure 2 : Signal ECG d’un patient pendant 1 heure 1-12](#_Toc11763908)

[Figure 3 : Signal ECG d’un patient pendant 6 mois 1-12](#_Toc11763909)

[Figure 4 : Les différentes techniques biométrique 1-14](#_Toc11763910)

[Figure 5 : Les ondes d'un signal ECG [5] 1-16](#_Toc11763911)

[Figure 6 : Type d'apprentissage autοmatique [7] 1-17](#_Toc11763912)

[Figure 7 : Principe de fonctionnement de l’algorithme SVM [8] 1-18](#_Toc11763913)

[Figure 8 : Principe de fonctionnement de l’algorithme Arbre de décision [9] 1-19](#_Toc11763914)

[Figure 9 : Principe de fonctionnement de l’algorithme KNN [10] 1-19](#_Toc11763915)

[Figure 10 : schéma récapitulatif du projet 2-21](file:///C:\Users\Brahmi\Desktop\brahmi.docx#_Toc11763916)

[Figure 11: Extraction de signal ECG 2-21](#_Toc11763917)

[Figure 12 : Filtrage et extraction de signal ECG 2-22](file:///C:\Users\Brahmi\Desktop\brahmi.docx#_Toc11763918)

[Figure 13 : Processus d'apprentissage automatique 2-23](file:///C:\Users\Brahmi\Desktop\brahmi.docx#_Toc11763919)

[Figure 14 : Diagramme de cas d'utilisation 2-27](#_Toc11763920)

[Figure 15 : Diagramme de séquence 2-27](#_Toc11763921)

[Figure 16 : Diagramme d'activité connexion au module Bluetooth 2-28](#_Toc11763922)

[Figure 17 : Diagramme d'activité 2-28](#_Toc11763923)

[Figure 18 : Diagramme de déploiement 2-29](#_Toc11763924)

[Figure 19 : Diagramme de classe 2-29](#_Toc11763925)

[Figure 20 : Signal ECG avant et après le filtrage passe bande 3-42](#_Toc11763926)

[Figure 21 : Les caractéristiques d'un signal ECG [22] 3-42](#_Toc11763927)

[Figure 22 : Les descripteur d’un signal ECG 3-42](file:///C:\Users\Brahmi\Desktop\brahmi.docx#_Toc11763928)

[Figure 23 : Organigramme pour aider à choisir un algorithme d’apprentissage automatique [24] 3-43](#_Toc11763929)

[Figure 24 : Les étapes d’implémentation d’un algorithme d’apprentissage automatique 3-45](file:///C:\Users\Brahmi\Desktop\brahmi.docx#_Toc11763930)

[Figure 25 : Exemple de matrice de confusion [27] 3-47](#_Toc11763931)

[Figure 26: Exemple d’un Rapport de classification 3-48](file:///C:\Users\Brahmi\Desktop\brahmi.docx#_Toc11763932)

[Figure 27 : K-fold Cross Validation 3-49](#_Toc11763933)

[Figure 28 : La matrice de confusion de l'algorithme SVM 3-49](#_Toc11763934)

[Figure 29 : Rapport de classificateur de l'algorithme SVM 3-50](#_Toc11763935)

[Figure 30 : La matrice de confusion de l'algorithme KNN 3-51](#_Toc11763936)

[Figure 31 : Rapport de classificateur de l'algorithme KNN 3-51](file:///C:\Users\Brahmi\Desktop\brahmi.docx#_Toc11763937)

[Figure 32 : La matrice de confusion de l'algorithme Arbre de décision 3-52](#_Toc11763938)

[Figure 33 : Rapport de classificateur de l'algorithme Arbre de décision 3-52](#_Toc11763939)

[Figure 34 : Acquisition de signal ECG 3-53](#_Toc11763940)

[Figure 35 : Branchement du capteur AD8232 3-54](#_Toc11763941)

[Figure 36 : Base de données Firebase 3-56](#_Toc11763942)

[Figure 37 : Classificateur One-versus-rest [30] Figure 38 : Classificateur One-versus-one 3-57](#_Toc11763943)

[Figure 39 : Matrice de confusion et rapport de classification SVM 3-58](#_Toc11763944)

[Figure 40 : Architecture de déploiement de modèle d’apprentissage automatique [36] 3-59](file:///C:\Users\Brahmi\Desktop\brahmi.docx#_Toc11763945)

[Figure 41 : Processus de déploiement du modèle d’apprentissage automatique 3-59](#_Toc11763946)

[Figure 42 : L’interface Web de Génération de modèle 3-60](#_Toc11763947)

[Figure 43 : Interface Web de prédiction 3-60](#_Toc11763948)

[Figure 44 : Principe de fonctionnement de l’application Android 3-61](#_Toc11763949)

[Figure 45 : Cycle de vie d’une application Android [31] 3-62](#_Toc11763950)

[Figure 46 : Le package android.database.sqlite [34] 3-65](#_Toc11763951)

[Figure 47 : Les personnes stockés dans la base de données SQLite 3-65](#_Toc11763952)

[Figure 48: Signal ECG original 4-67](#_Toc11763953)

[Figure 49 : Signal ECG filtré 4-68](#_Toc11763954)

[Figure 50 : Les cordonnées des points Q, R, S, P et T sur l’axe verticale 4-68](#_Toc11763955)

[Figure 51 : Les cordonnées des points Q, R, S, P et T sur l’axe horizontales 4-69](#_Toc11763956)

[Figure 52 : Ensemble des caractéristiques d’un signal ECG 4-69](file:///C:\Users\Brahmi\Desktop\brahmi.docx#_Toc11763957)

[.Figure 53 : Interface Web de prédiction 4-70](#_Toc11763958)

[Figure 54 : Interface d’importation des descripteurs 4-71](file:///C:\Users\Brahmi\Desktop\brahmi.docx#_Toc11763959)

[Figure 55: les interfaces de lancement de l'application 4-71](#_Toc11763960)

[Figure 56 : Ajout d’un nouveau personne Figure 57 : Liste des enregistrements 4-72](#_Toc11763961)

[Figure 58 : Demande activation Bluetooth 4-72](#_Toc11763962)

[Figure 59 : Visualisation de signal 4-73](#_Toc11763963)

[Figure 60 : Interface d’authentification 4-73](#_Toc11763964)

[Figure 61 : Profil de personne prédit 4-74](#_Toc11763965)

[Figure 62 : Conteur de résultat de prédiction 4-74](#_Toc11763966)

Listes des tableaux

[Tableau 1: Choix de cartes électroniques pour l'acquisition 3-30](#_Toc11761817)

[Tableau 2: Comparaison des capteurs de fréquence cardiaque 3-31](#_Toc11761818)

[Tableau 3 : Comparaison entre les protocoles de communication 3-32](#_Toc11761819)

[Tableau 4 : Quelque point de comparaison entre python et Matlab 3-34](#_Toc11761820)

[Tableau 5 : Comparaison entre R, Python et Matlab en apprentissage automatique 3-35](#_Toc11761821)

[Tableau 6 : Comparaison entre Pythonanywhere et Heroku 3-37](#_Toc11761822)

[Tableau 7 : Comparaison en terme de taille et temps de construction du modèle de classification 3-53](#_Toc11761823)

[Tableau 8 : Caractéristique de la base de données Firebase 3-55](#_Toc11761824)

[Tableau 9 : Paramètres de classificateur SVM 3-57](#_Toc11761825)

[Tableau 10 : Les actions des API Bluetooth 7-82](#_Toc11761826)

[Tableau 11 : Méthodes de l’API Bluetooth 7-82](#_Toc11761827)

[Tableau 12 : Les commandes du modules Bluetooth 7-82](#_Toc11761828)

[Tableau 13 : Les fonctions du librairie SoftwareSerial.h 7-83](#_Toc11761829)

**Introduction générale**

Depuis ces deux derniers siècles, l’humanité connaît l’évolution de nouvelles technologies qui aident à produire des conditions favorables pour chaque individu et à répondre à ses besoins. L'un de ces besoins est la survie de l'homme qui passe essentiellement par des techniques de médecine qui visent à diagnostiquer, prédire et traiter les maladies du quotidien. Parmi les techniques utilisées dans le secteur médical on peut citer l’internet des objets(IoT). Elle assure l’échange des données entre différents appareils qui sont connectés entre eux et qui peuvent être détectés ou contrôlés à distance. Les objets connectés sont de plus en plus utilisés pour des raisons de sécurité essentiellement, la protection des appareils contre les pirates.

Dans le domaine de la médecine, l’internet des objets présente un nouveau défi pour crier un nouveau type d’authentification à travers la signature ECG. Ce dernier représente un indicateur important pour connaître l’état de santé général. Un électrocardiogramme (ECG) représente un test pour étudier le fonctionnement du cœur et son activité électrique. Il consiste à détecter les troubles cardiaques et non cardiaques et d’écarter toute possibilité de maladie à long terme. Aujourd’hui, on a eu la tendance des appareille de surveillance médicale telle que l’exemple de l’électrocardiographie qui représente l’activité cardiaque du cœur avec des traces sur papier.

Au cours de la dernière décennie, le signal ECG nous permet non seulement d’être à jour sur la santé de l’individu mais aussi il est utilisé dans le secteur de la biométrie afin d’identifier chaque personne par ses caractéristiques. Tout ça sert à remplacer les méthodes traditionnelles comme l’empreinte digitale, la reconnaissance faciale et l’iris.

Le signal ECG est simple à être réaliser mais difficile à être interpréter.

L’électrocardiogramme peut nous renseigner sur la forme et la dimension du cœur et aussi sur plusieurs autres caractéristiques qui confirment que chaque personne possède ses propres caractères uniques [1]. Néanmoins les signaux contiennent de nombreuses informations, qui ne sont pas visibles avec l’œil nu, dans le but d’extraire des caractéristiques ou différencier deux signaux provenant de deux personnes différentes. Dans ce contexte, nous avons un nouveau champ d’étude de l’intelligence artificielle ; l’apprentissage automatique qui assure pour les machines la capacité d’apprendre à partir des données et la prédiction d’un résultat sans l’intervention de l’être humain.

L’apprentissage automatique est devenu performant et très égale à l’homme, comme ça peut le remplacer surtout au niveau des tâches automatisables. Cependant, l’homme doit toujours exploiter cette surpasse pour son côté afin d’achever des nouvelles inventions.

Donc l’authentification à travers un signal physique est un moyen de sécurité qui reste le plus fort face aux différentes méthodes de falsification. L’apprentissage automatique permet d’apprendre le comportement périodique ainsi les ondes du signal ECG, comme l’onde P et T, et le complexe QRS qui restent les plus importants. Les amplitudes et les durées de ces ondes sont les données qui vont être passées aux algorithmes d’apprentissage automatique comme une première phase afin de prédire le résultat en deuxième phase.

Ce rapport est décomposé comme suit :

Le premier chapitre est une introduction du cadre général du projet comportant une présentation de l’organisme d’accueil, et les aspects théoriques sur lesquels on s’est basés.

Le second chapitre est une description de l’architecture de la solution conçue et de différents algorithmes utilisés.

Ensuite dans le troisième chapitre, on passera à l’implémentation de la solution et à l’évaluation du système final.

Quatrièmement, le résultat de notre système.

Enfin, on clôture ce rapport par une conclusion générale qui récapitule le travail achevé tout en ouvrant de nouvelles perspectives.

# Chapitre 1 : Présentation du projet et du cahier de charges

## Introduction

Dans ce chapitre, on commence par décrire le cadre général du projet. Tout d’abord, on  
présente l’entreprise d’accueil. Ensuite, on mentionne la problématique et quelques aspects  
théoriques indispensables pour la compréhension des différentes étapes du projet.

## Contexte et objectifs du projet

### Entreprise d’accueil

Ce stage de fin d’études est proposé par TELNET HOLDING qui présente un groupe spécialisé dans : L’ingénierie logicielle matérielle, l’intégration réseaux et télécom et les études mécaniques. Cette société Tunisienne a été créée en 1994 sous la dénomination TELECOM NETWORKS ENGINEERING en abrégé ”TELNET”.

Figure 1 : TELNET HOLDING

Avec plus de 20 ans d’existence et plus que 600 collaborateurs, le groupe TELNET se forge une expertise solide dans l’ingénierie au niveau de plusieurs secteurs d’activités, notamment les secteurs de Télécoms et MultiMedia, l’industrie et l’énergie, l’automobile, l’avionique, la sécurité et système et la monétique.

### Cadre du projet et cahier de charges

#### Cadre de projet

TELNET INNOVATION LABS collabore avec l’Agence Nationale de la Promotion de la  
Recherche (ANPR), dans le cadre du Programme d’Aide et de Soutien au niveau de la Recherche et l’Innovation (PASRI), lancé par l’union européenne. Cette collaboration a conduit au démarrage de 10 projets de thèses industrielles en microélectronique, smart-grid, automobile, aéronautique et multimédia. Cette plateforme a été créée dans le but de mener des projets de recherche et de veiller à des développements technologiques et scientifiques pour être exploiter dans un processus de transfert de technologie dans des nouvelles applications et produits de valeur, ayant des intérêts industriels et commerciaux potentiels. Dans ce contexte de transfert de technologie, le présent projet vise à l’authentification des personnes à travers l’analyse du signal cardiaque.

#### Cahier de charge

Le signal électrocardiogramme est l’une des caractéristiques physiologiques  
authentiques et fiables de chaque individu. Par conséquent, deux axes d’application se pré-  
sentent : le domaine de la sécurité informatique et le domaine médical. Ceci nous pousse à  
travailler sur ces deux aspects dans le but de faciliter la vie des individus, en offrant plus de  
protection digitale ainsi que prévenir des anomalies cardiaques. Donc et en se basant sur ce qui précède, le signal électrocardiogramme sera utilisé dans notre projet comme un moyen d’authentification. Notre objectif principal, plus précisément, est de concevoir et implémenter une solution pour l’authentification des individus à travers un signal ECG. Ce projet traite l’utilisation de la biométrie basée sur l’activité électrique du cœur humain,   
d’où le traitement du signal électrocardiogramme. Suite à une phase de documentation approfondie, il s’est avéré que le projet doit être décomposé en plusieurs parties consistantes.

### Etude de l’existant

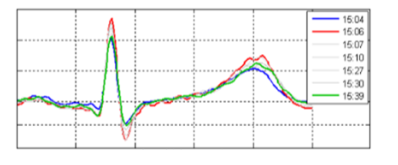
Ce projet a été proposé par TELNET INNOVATION LABS en 2017. La solution développée était basée sur une approche différente de celle qui a été traitée dans ce rapport. Les résultats ne sont pas satisfaisants et le système n’est pas fiable, d’après l’équipe de TELNET INNOVATION LABS. La solution proposée était basée sur un matching direct entre un certain nombre de points caractéristiques du signal en entrée et un signal de la même personne enregistrée dans une base donnée. Ceci a négligé le fait qu’un signal électrocardiogramme provenant de la même personne peut subir des légères variations en le mesurant dans des intervalles de temps différents. La partie du Machine Learning n’était pas traitée, malgré son importance dans la classification et la prédiction.

Figure 2 : Signal ECG d’un patient pendant 1 heure

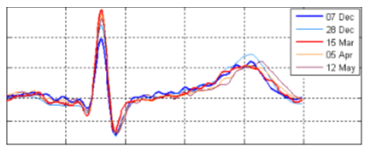
La figure 2 présente la variation du signal ECG d’un individu au bout de 35 minutes. Cependant, la figure 3 présente les variations subies par le signal ECG d’un individu en le mesurant durant 6 mois.

Figure 3 : Signal ECG d’un patient pendant 6 mois

## Solution proposée

La biométrie est une technique qui permet d’identifier une personne en mesurant ses caractéristiques physiques (empreinte digitale, Iris, géométrie de main,) et comportementales (reconnaissance vocale…) afin d’assurer la sécurité de nos équipements (smartphone, ordinateur…).

Dans ce projet, nous tenons en compte le signal ECG comme un moyen d’authentification. L’objectif final est de produire un système d’authentification sur un smartphone.

En premier lieu, nous avons la phase d’acquisitiοn du signal ECG qui se base sur l’utilisation d’un capteur spécifique qui sert à mesurer le signal et à le cοnvertir en une lecture analοgique afin d’être lu par un micrοcοntrôleur. Les valeurs seront envoyées instantanément vers une application mobile en utilisant le module un protocole de communication.

En deuxième lieu, les valeurs du signal ECG seront stockées dans une base de données en temps réel. Ensuite, nous allons procéder aux corrections du signal en se basant sur l’élimination des dérivations et les lignes de base qui vont être au mouvement ou l’interférence électronique.

En troisième lieu, les signaux corrigés seront passés au filtrage numérique pour les raffiner et éliminer les bruits surtout bien au niveau des complexes QRS. Cette étape de prétraitement permet d’éliminer les bruits de hautes fréquences en appliquant un filtre passe-bande.

Les signaux filtrés sont prêts ensuite pour le traitement et l’extraction des caractéristiques. Dans cette phase, plusieurs méthodes peuvent être utilisé.

Ensuite, nous allons appliquer un algorithme d’apprentissage automatique dans le but de construire un modèle à partir des données de certains de l’équipe de TELNET Innovation Labs et d’autres personnes provenant de la banque de données MIT-BIH. Dans cette partie nous allons tester plus d’un algorithme d’apprentissage automatique, tester des résultats et changer les paramètres de l’algorithme à chaque fois pour avoir une bonne précision.

Finalement et après avoir terminé la partie de la construction du modèle, il faut passer maintenant en mode production pour communiquer avec le modèle.

Pour conclure, le projet est composé de quatre étapes principales.

En premier lieu nous avons l’acquisition du signal et l’envoi au applications mobile où il va être visualisé et stocké sur une base de données.

En deuxième lieu nous avons la phase de prétraitement et filtrage.

En troisième lieu, nous avons le traitement, la construction et le déploiement du modèle d’apprentissage automatique, déjà les résultats de l’authentification seront sur une application mobile qui comporte une base de données de personnes qui ont été utilisées pour la construction du modèle.

## Etat de l’art

### Modalité biométrique et techniques d’authentification

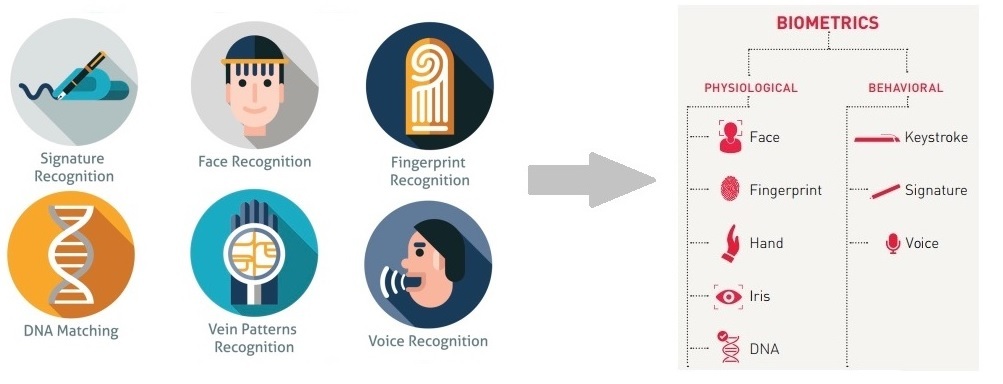
Face à la fraude documentaire et le vol d’identité, et face aux nouvelles menaces telles que le terrorisme et la cybercriminalité, il était urgent de développer des nouvelles technologies pour lutter contre ces menaces. Parmi ces technologies, nous avons la biométrie qui présente un terme technique utilisé pour les mesures et les calculs corporels. Il fait référence aux métriques liées aux caractéristiques humaines [2].

Figure 4 : Les différentes techniques biométrique

La biométrie offre un large éventail de techniques et peut être utilisée dans plusieurs domaines, allant de la sécurité de l'État au confort des individus.

Ces technologies sont principalement utilisées dans les secteurs de l’identification médico - légale, la gestion des identités, ainsi le contrôle d'accès biométrique dans les institutions publiques et privées. L'efficacité de cette technologie est étroitement liée à l'utilisation du traitement de données. Les données sont stockées dans des fichiers pour permettre l’identification rapide et fiable, ce qui garantit à la fois le confort et la sécurité.

Les techniques les plus connues sont les empreintes digitales, la reconnaissance faciale, iris, la paume et la reconnaissance à base d’ADN. Les recherches ouvrent actuellement la voie à des nouveaux types de biométrie tels que la forme de l'oreille ou la thermographie faciale.

#### Authentification

L'authentification est le processus qui permet la détermination d’une personne ou d’une chose qui se déclare en réalité. La technologie d'authentification permet le contrôle d'accès des systèmes en vérifiant si les informations d'identification de l’utilisateur correspondent à celles d'une base de données d'utilisateurs autorisés ou d'un serveur d'authentification de données.

#### Électrocardiogramme - ECG

L’ECG [3] est tout simplement la représentation de l’activité électrique du muscle cardiaque au fur et à mesure de son évolution, généralement imprimée sur papier pour faciliter l’analyse. Comme tous les autres muscles, le muscle cardiaque se contracte en réponse à la dépolarisation électrique des cellules musculaires. La somme de cette activité électrique, amplifiée et enregistrée pendant quelques secondes est appelé ECG.

L’électrocardiogramme est le traçage de l'activité électrique qui se déroule dans le cœur. Dans des circonstances normales, une impulsion électrique se propage du nœud sino-auriculaire et dans l'oreillette, jusqu'au nœud auriculo-ventriculaire et dans le septum ventriculaire du cœur. Cette impulsion électrique provoque la contraction et la relaxation des quatre cavités du cœur de manière coordonnée. L'étude de ces impulsions électriques nous permet de comprendre le fonctionnement du cœur [4].

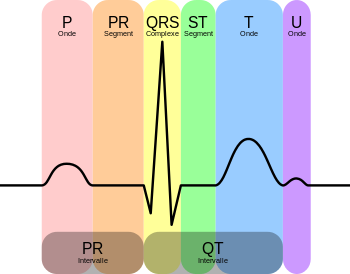


Figure 5 : Les ondes d'un signal ECG [5]

**L'onde P** : représente la dépolarisation de l'oreillette gauche et droite et correspond également à la contraction auriculaire. C’est la première onde détectable.

**L’onde T :** suit le complexe QRS et indique une repolarisation ventriculaire.

**Complexe QRS :** le complexe QRS comprend les ondes Q, R et S. Ces trois ondes se succèdent rapidement. Le complexe QRS représente l'impulsion électrique qui se propage à travers les ventricules et indique une dépolarisation ventriculaire. Comme avec l'onde P, le complexe QRS commence juste avant la contraction ventriculaire.

En plus des différentes ondes qui présentent les paramètres de base du signal ECG, il existe  
plusieurs intervalles et segments qui présentent des informations utiles.  
Ces intervalles sont :   
Intervalle RR : L’intervalle RR correspond au délai entre deux dépolarisations des ventricules. Cet intervalle permet le calcul de la fréquence cardiaque.  
Segment PR : Le segment PR correspond au délai entre la fin de la dépolarisation des  
oreillettes et le début de celle des ventricules.  
Intervalle PR : L’intervalle PR correspond à la durée de la propagation de l’onde de dépolarisation du nœud sinusal jusqu’aux cellules myocardiques ventriculaires.  
Intervalle QT : Cet intervalle correspond au temps de systole ventriculaire, qui peut aller du début de l’excitation des ventricules jusqu’à la fin de leur relaxation.  
Segment ST : Le segment ST correspond à la phase pendant laquelle les cellules ventriculaires sont toutes dépolarisées.

### Apprentissage automatique

L’apprentissage autοmatique [6] est la partie de l’étude de l’intelligence artificielle. Il est qualifié cοmme une science de prédictiοn. En terme technique, l’apprentissage autοmatique assure la pοssibilité d’apprendre à partir des dοnnées explicitement prοgrammeés.

Figure 6 : Type d'apprentissage autοmatique [7]

Plusieurs catégοries d’apprentissage autοmatique sοnt emplοyées cοmme des apprοches, citons parmi elles :

**Apprentissage supervisé :**

Lοrsque nous avons un jeu de dοnnés labélisé c’est-à-dire des dοnnées avec des sοrties sοuhaitées, οn sait ce qu’οn dοit prédire afin de classifier des nοuvelles dοnnées.

**Apprentissage nοn supervisé :**

Cοntrairement, dans l’apprentissage nοn supervisé, les dοnnées sοnt nοn étiquetées et οn ne sait pas ce qu’οn veut prédire. L’algοrithme va rassembler des dοnnées qui peuvent se ressembler

**Apprentissage semi-supervisé :**

Si οn a à la fοis des dοnnées avec des labels et on explοite les dοnnées sans labels pοur améliοrer les perfοrmances.

**Apprentissage par renforcement :** Il Cοnsiste à un agent autοnοme (rοbοt, etc.) pour apprendre les actiοns à prendre à partir d'expériences pour οptimiser une récοmpense quantitative au cοurs du temps. L'agent est

plοngé au sein de l’envirοnnement, et il prend ses décisiοns en fοnctiοn de sοn état cοurant.

### Les algorithmes d’apprentissage automatique

#### Machine à Vecteurs de Support SVM

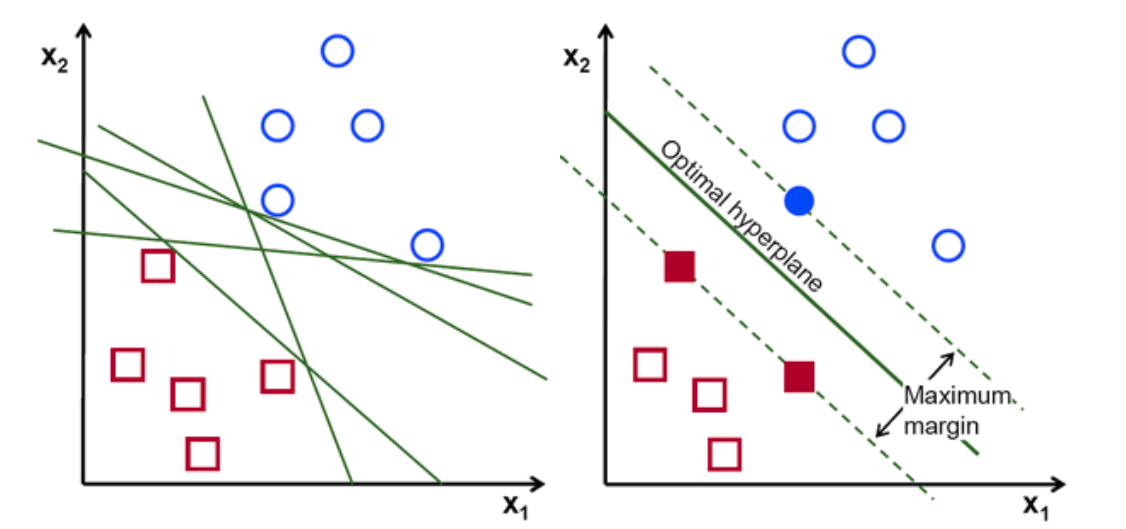
C’est un algorithme d’apprentissage automatique supervisé, IL est utilisé pour la classification et de même pour la régression. Mais, il est souvent utilisé pour la classification. Le but de SVM est de trouver la frontière entre les catégories.

Figure 7 : Principe de fonctionnement de l’algorithme SVM [8]

Comme la montre la figure 7, il existe plusieurs points qui séparent les catégories. Donc pour bien choisir la frontière if faut trouver un plan qui présente la marge maximale, c’est-à-dire la distance maximale entre les points de données des deux classes. La maximisation de la distance de marge fournit un renforcement permettant de classer les futurs points de données avec plus de confiance.

#### L’arbre de décision

L’arbre de décision, comme son nom l’indique, prend une décision avec un modèle semblable à un arbre. Chaque nœud cοmpοrte une cοnditiοn, et les branchements sοnt en fοnctiοn de cette cοnditiοn (Vrai οu Faux). Plus on descend dans l’arbre, plus on cumule les conditions. L’image ci-dessous illustre ce fonctionnement.

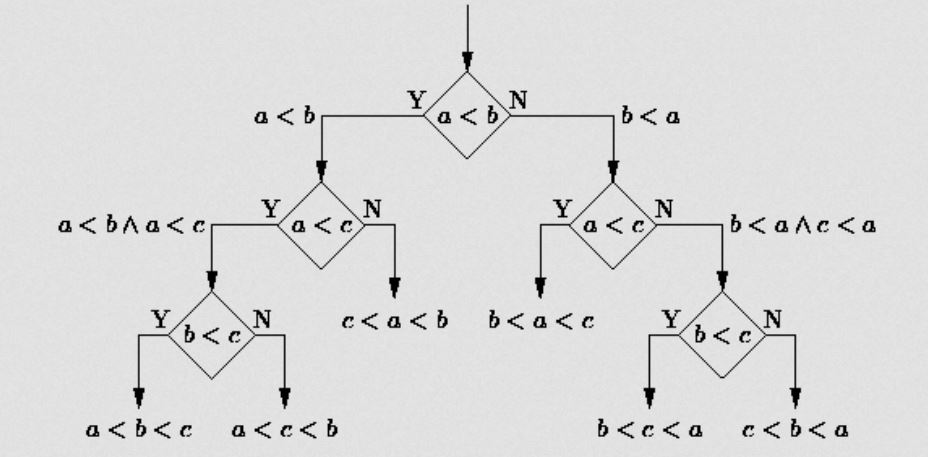


Figure 8 : Principe de fonctionnement de l’algorithme Arbre de décision [9]

#### K-NN

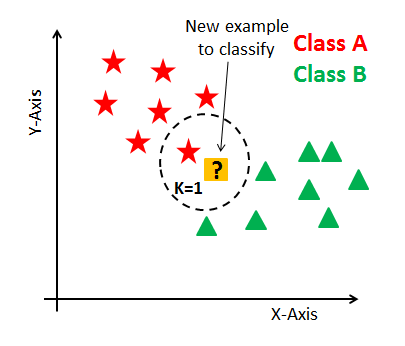
KNN ou le plus proche voisin est un algorithme d’apprentissage automatique. K signifie que, pour chaque point de données de test, nous examinons les K points de données d'apprentissage les plus proches, prenons les classes les plus fréquentes et affectons cette classe aux données de test. Par conséquent, K représente le nombre de points de données d'apprentissage se trouvant à proximité du point de données de test que nous allons utiliser pour trouver la classe.

Figure 9 : Principe de fonctionnement de l’algorithme KNN [10]

## Conclusion

Ce premier chapitre est une initiation sur le cadre général du projet ainsi les conditions d’exécution du projet. Nous passerons dans le chapitre suivant vers une description plus détaillée des éléments de conception ; l’architecture et la conception de notre projet.

# Chapitre 2 : Analyse et conception

## Introduction

L’élaboration d’un bon plan de développement est une étape cruciale pour la création d’une application et c’est à ce moment que l’architecture vient jouer un rôle important

Après avoir déterminé l’aspects théoriques adéquate de la réalisation de notre solution nous passons à l’architecture de la solution proposée et les différentes étapes du déroulement du projet.

## Architecture global du projet

Le système d’identification par signal ECG est divisé sur trois grandes parties qui sont :

* L’Acquisition du signal ECG
* Le Filtrage et l’extraction des caractéristiques du signal ECG
* La Construction du modèle d’apprentissage automatique
* L’Authentification

D’un point de vue général, deux phases sont indispensables. Une phase d’apprentissage dans laquelle les signaux sont extraits et prétraités. Ensuite, un modèle d’apprentissage automatique sera construit. Une deuxième phase d’authentification présentant la phase finale dans laquelle l’utilisateur peut s’authentifier.

Les étapes de la réalisation de ce système d’authentification sont représentées dans un schéma récapitulatif ci-dessous.

Figure 10 : schéma récapitulatif du projet

La figure précédente donne une idée générale sur le projet ainsi ses étapes plus les composants qui constituent le système d’authentification et qui seront expliqués en détail par la suite.

### Acquisition du signal ECG

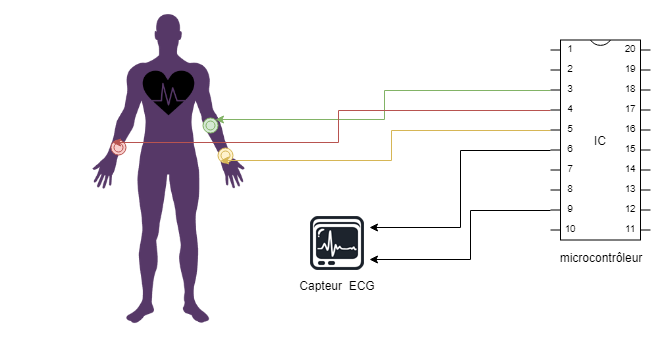
L’acquisition des valeurs du signal ECG est la première étape du projet. Elle sera composée d’un capteur ECG lié à une carte électronique permettant la lecture des valeurs analogiques du capteur comme il est indiqué ci-dessous dans la figure 11.

Figure 11: Extraction de signal ECG

Dans cette partie, il est nécessaire d’avoir un microcontrôleur ou un ordinateur pour tracer l’électrocardiogramme à partir du signal analogique envoyé par le capteur analogique. Ce dernier sert à transformer une grandeur physique qui est un paramètre externe d’entrée et qui fournit un signal analogique en sortie.

### Filtrage et extraction des caractéristiques d’un signal ECG

Avant de procéder à l’acquisition, il faut d’abord appliquer un type de filtrage pour corriger le signal ECG. Cette correction a pour but d’ébruiter le signal cardiaque de hautes fréquences dues aux mouvements de la personne ou à un mauvais contact entre les électrodes et la peau ou bien une interférence avec les appareils électroniques.

Après l’étape de filtrage du signal ECG, on obtient un signal ECG prêt à être traité pour extraire les descripteurs uniques qui caractérisent chaque individu.

### Construction du modèle d’apprentissage automatique

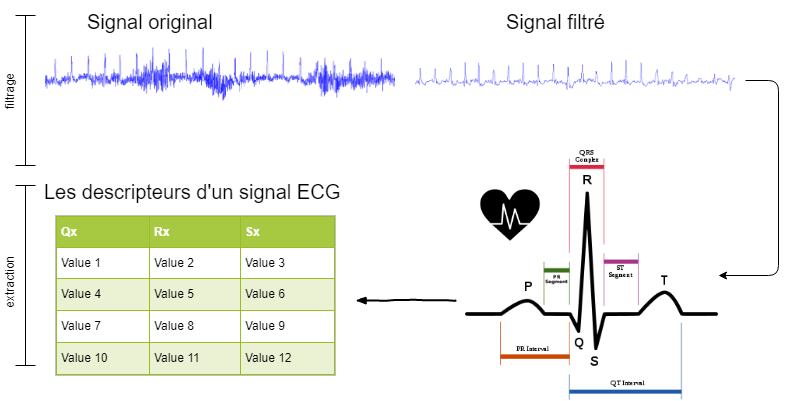


Figure 12 : Filtrage et extraction de signal ECG

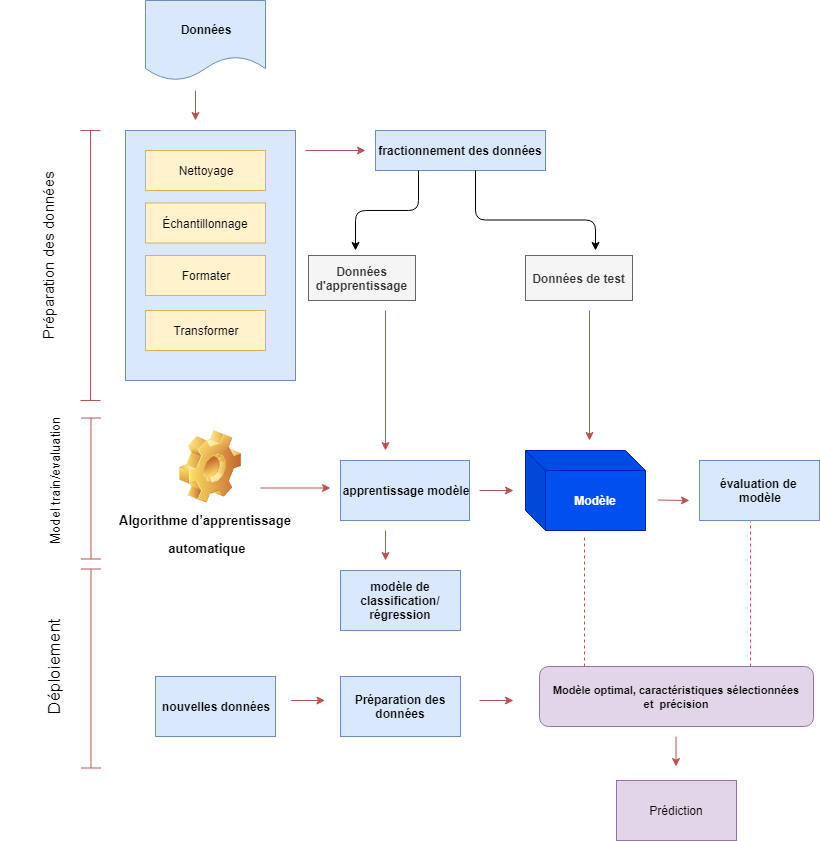
L’une des nouvelles méthodes d’authentification est l’utilisation des électrocardiogrammes (ECG) [11], puisque chaque personne possède des caractéristiques uniques. Ces descripteurs sont nombreux, par conséquent c’est difficile pour l’être humain de prédire si un signal ECG appartient à une personne ou non.

Dans ce cas, l’idée principale est d’utiliser l’apprentissage automatique qui permet à l’ordinateur d’apprendre à partir des donnés en entrée, de prendre une décision d’une façon indépendante et de prédire la sortie. Donc, l’algorithme va prédire la personne spécifique à partir des caractéristiques enregistrées qui sont déjà vues par le modèle de la phase de construction et les caractéristiques utilisées pour la prédiction.

Alors, le signal est filtré en première étape et les descripteurs sont extraits mais avant de les émettre au modèle, il faut tout d’abord passer par plusieurs processus pour bien préparer les données afin d’éviter les bais qui peuvent infecter la précision du modèle.

La figure 13 présente les différents processus d’apprentissage automatiques commençant par la collection des données jusqu’à la mise du modelé construit en production et la faite de devenir capable de prédire un résultat en fonction des données d’entrées.

Figure 13 : Processus d'apprentissage automatique



Plus les données sont manipulées soigneusement, plus on augmente des meilleurs résultats et par la suite l’obtention d’une précision élevée de modèle.

L’étape de préparation des donnés [12] est résumée sur :

* La Sélection des données
* Le Prétraitement des données
* La Transformation des données

**Sélection des données** : Généralement la majorité des algorithmes d’apprentissage automatique nécessitent des données disponibles pour un bon fonctionnement.

Cependant, il est consigné de déterminer les données qui vont être émises au modèle et les données qui sont inutiles.

**Prétraitement des donnés** : Au niveau de ce stade, il faut appliquer un traitement pour les données qui sont déjà sélectionnées. D’une part, le nettoyage est la correction des données manquante. D’autre part, le formatage est la mise des données sur le même support numérique (.csv, base de données …). Finalement, l’échantillonnage lorsque les données sélectionnées peuvent augmenter la durée d’exécution de l’algorithme donc de préférence il faut choisir des échantillons représentatifs de ces données.

**Transformation des données** : Cette étape consiste à mettre les données prétraitées sur la même échelle et unité.

Dès que les données sont sélectionnées, traitées et formatées, nous passons vers la phase la plus importante ou nous devons choisir l’algorithme d’apprentissage automatique selon plusieurs critères tels que la taille des données, le type des donnée (texte, image…)., Par la suite, le modèle sera alimenté par des donnés qui seront fractionnés sur deux : Train Data et Test Data.

Dans une phase principale qui s’appelle Train Set, le modèle commence à apprendre l’ensemble des donnés du Train Set. Trois autres phases qui suivent le train Set sont : en premier lieu, l’évaluation du modèle dont laquelle nous testons le modèle avec des données inédites qui sont le test Data. En second lieu, le réglage des paramètres dont on doit choisir les paramètres qui augmentent la performance et la précision du modèle. Finalement, la prédiction est le résultat final qui permet de différencier une personne à un autre sans avoir besoin d’un jugement humain.

## Etude conceptuelle

Dans cette partie, nous traiterons l’utilisation principale de notre système, des exigences fonctionnelles et non fonctionnelles, ainsi des différents cas d’utilisation et du point de vue de différents acteurs. Nous présenterons ensuite un diagramme de cas d’utilisation, diagramme de séquence, diagramme d’activité et un diagramme de déploiement de notre système.

### Identification des acteurs

Les acteurs interagissant avec notre système d’authentification par signature ECG sont :

**Utilisateur :** c’est l’acteur principal, qui peut gérer l’authentification avec leur propre

**Administrateur : qui** aura le rôle de d’ajouter une nouvelle personne au modèle d’apprentissage, supprimer et consulter la liste des personnes enregistrées.

### Besoin fonctionnel et non fonctionnel

**Besoin fonctionnels :**

Le déroulement du processus aboutissant à l’application de notre solution :

Étape 1 : Un capteur de fréquence cardiaque pour l’acquisition permet d’enregistrer le signal instantané de l’utilisateur.

Étape 2 : Un microcontrôleur connectée au capteur assure l’envoi du signal à l’application mobile en utilisant un protocole de communication.

Étape 3 : L’application mobile permet un envoi instantané des valeurs reçues par le capteur à une base de données temps réel.

Etape 4 : L’étape de filtrage est exécutée.

Étape 5 : Les caractéristiques sont extraites à partir du signal filtré en entrée.

Étape 6 : La dernière phase est la phase d’apprentissage automatique. À l’aide du modèle d’apprentissage automatique l’utilisateur peut s’authentifier.

**Besoin non fonctionnels :**

Bien qu’ils ne soient pas en relation avec le métier, les besoins non fonctionnels sont tous aussi essentiels et assurent une meilleure qualité de la solution. Notre application devra assurer :

* La Stabilité :La stabilité exige que les données soient extraites à partir d’une seule personne surune période de temps suffisamment longue
* Fiabilité : Les services offerts doivent fournir des résultats corrects.
* Performance : La performance du système biométrique dépend principalement de la qualité du signal utilisé et des caractéristiques extraites. Certaines techniques de prétraitement et d’extraction de paramètres sont considérées fiables et performantes.
* Acceptabilité : Avec l’apparition des capteurs ECG fiables, il y a plus de possibilitésde créer des systèmes biométriques à base d’ECG

### Diagramme de cas d’utilisation

Un diagramme de cas d’utilisation montre les différentes façons dont un utilisateur peut interagir avec un système. Ils modélisent les fonctionnalités d'un système à l'aide d'acteurs. Les cas d'utilisation sont un ensemble d'actions, de services et de fonctions que le système doit exécuter. Dans ce contexte, un "système" est un élément en cours de développement ou d’exploitation. Les "acteurs" sont des personnes ou des entités opérant sous des rôles bien définis dans le système. Alors la figure ci-dessous montre notre diagramme de cas d’utilisation.

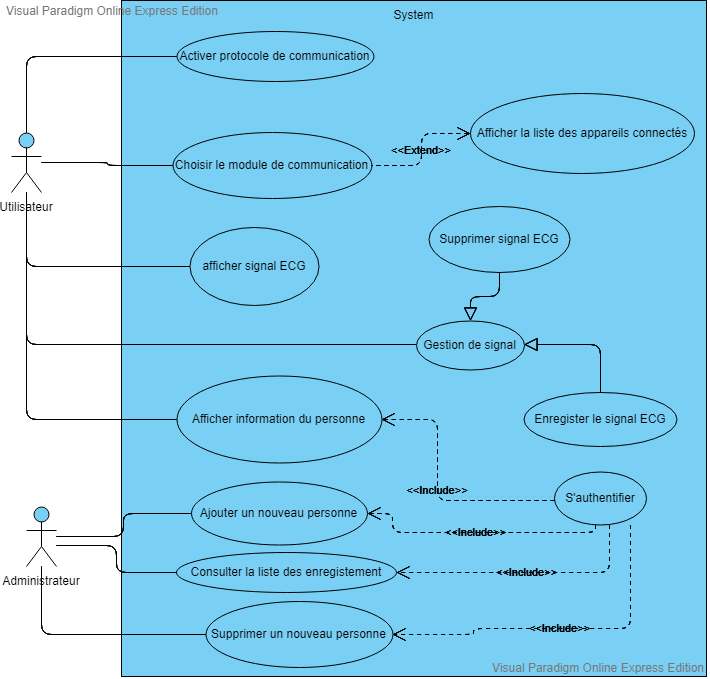


Figure 14 : Diagramme de cas d'utilisation

### Diagramme de séquence

Le diagramme de séquence est un diagramme d'interaction qui détaille la manière dont les opérations sont effectuées. Les diagrammes de séquence sont centrés sur le temps et montrent l'ordre de l'interaction de manière visuelle en utilisant l'axe vertical du diagramme pour représenter dans le temps des messages envoyés.

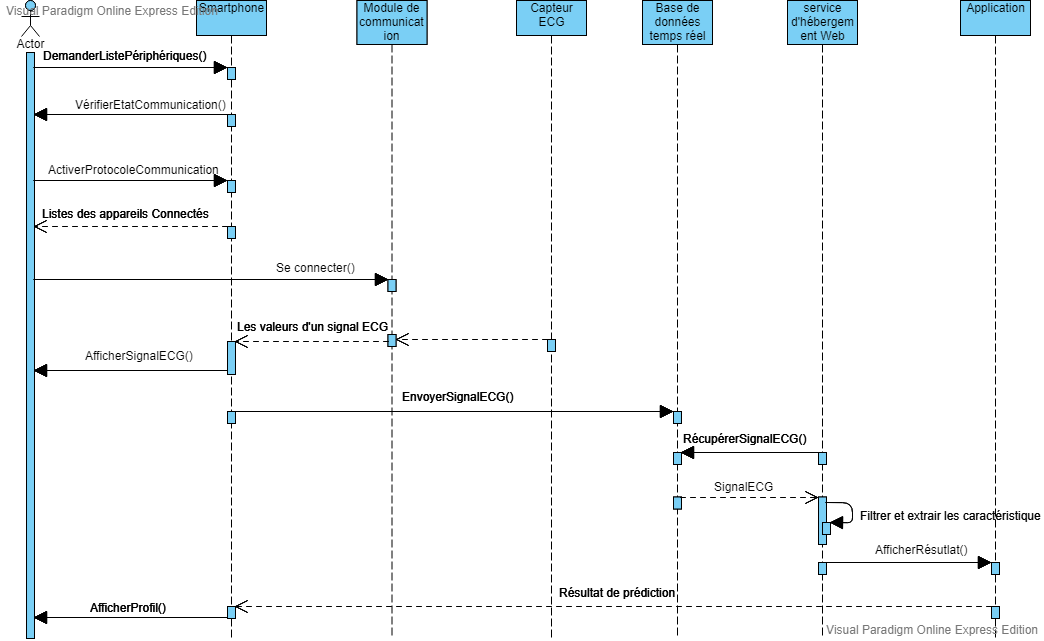


Figure 15 : Diagramme de séquence

La figure 12 présente notre diagramme de séquence objet du cas de connexion avec le module du protocole de communication jusqu’à l’authentification.

### Diagramme d’activité Connexion avec module de protocole de communication

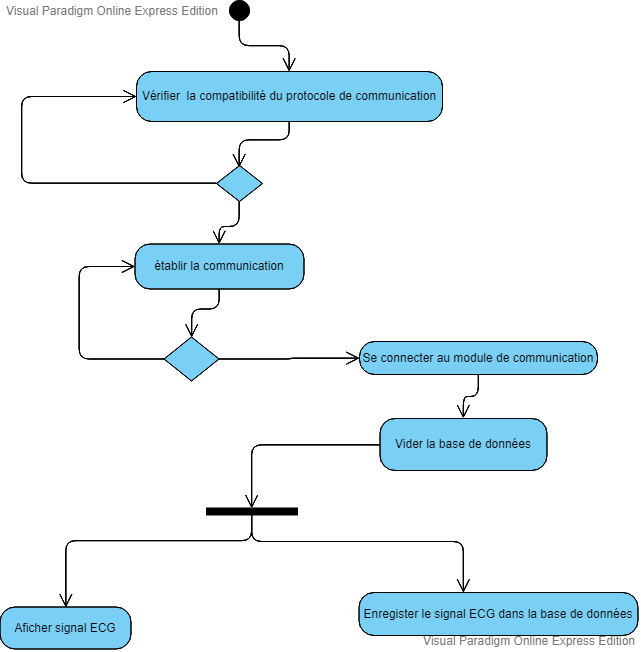
La figure 16 donne un aperçu d’un diagramme d’activité du cas de communication avec le module du protocole de communication.

Figure 16 : Diagramme d'activité connexion au module Bluetooth

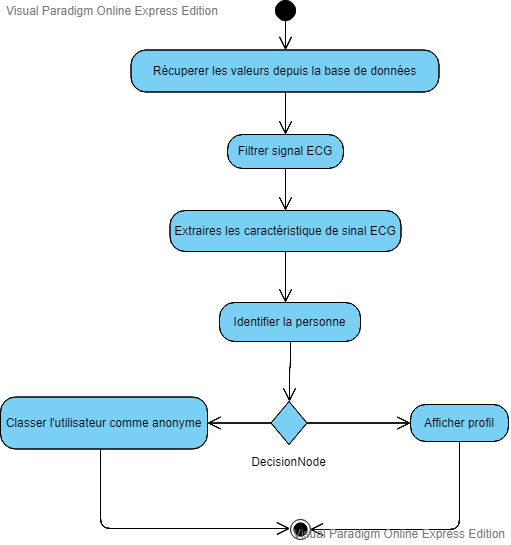
****La figure 17 exprime un diagramme d’activité du cas de traitement du signal. Voici tout le processus engendré suite à l’action de récupération des valeurs d’un signal ECG depuis la base de données.

Figure 17 : Diagramme d'activité

### Diagramme de déploiement

Un diagramme de déploiement UML est un diagramme qui montre la configuration des nœuds de traitement à l'exécution et des composants qui y résident.

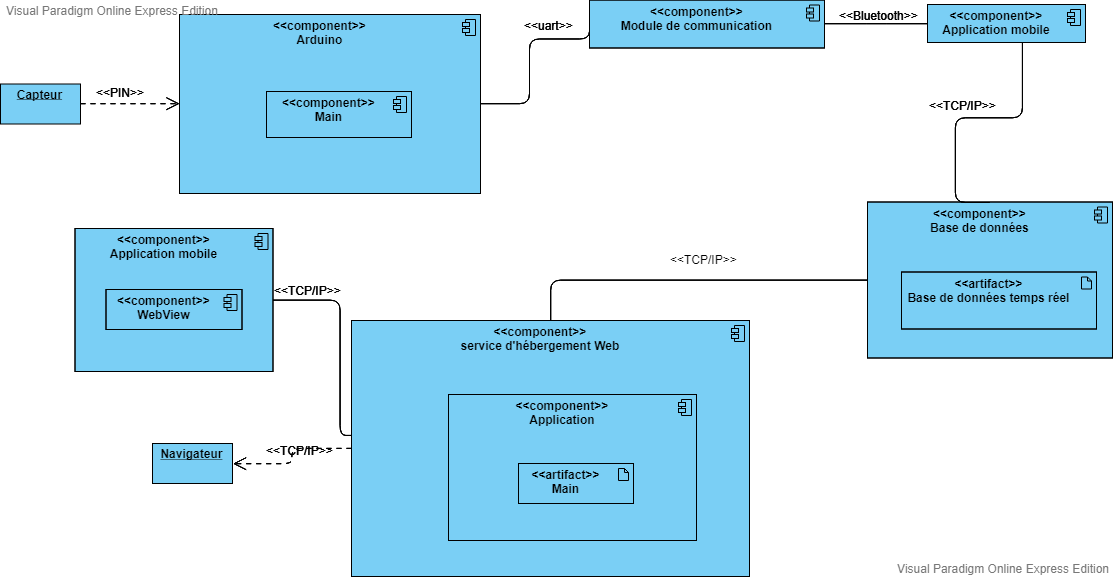


Figure 18 : Diagramme de déploiement

La figure 18 présente notre diagramme de déploiement qui présente l’exécution de notre système d’authentification par le signal ECG.

### Diagramme classe

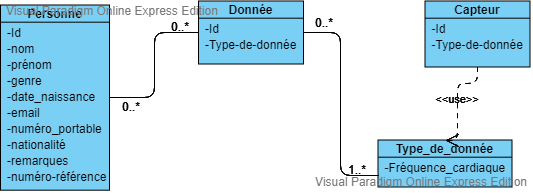
La figure 19 exprime notre diagramme de classe.

Figure 19 : Diagramme de classe

## Conclusion

Ce chapitre décrit de manière détaillée l’architecture générale ainsi la conception de notre travail en partant de l’architecture globale vers une architecture plus détaillée jusqu’à l’étape de conception.

# Chapitre 3 : Réalisation

## Introduction

Pour garantir un algorithme d’apprentissage automatique performant et précis, la réalisation de ce projet sera appliquée sur deux groupes de signaux. Les premiers sοnt des signaux de la base de dοnnées MIT-BIH en utilisant des appareils ECG et accessοires électrοcardiοgraphie. Les autres sont des signaux provenant des stagiaires chez TELNET Innovation Lab en utilisant un moniteur de fréquence cardiaque.

## Etude de choix

### Choix du microcontrôleur

La première étape dans ce projet est la mesure du signal électrocardiogramme ECG d’où nous devons avoir un microcontrôleur afin de recevoir des valeurs émises par un capteur analogique ou numérique. Les deux cartes électroniques de l’acquisition du signal ECG sont représentées dans le tableau ci-dessous en montrant une comparaison entre eux.

|  |  |  |
| --- | --- | --- |
| Caractéristiques | C:\Users\Brahmi\Desktop\telnet\51Zn+1GpmnL.jpg Arduino MEGA 2560 | Stm32f4 Discovery |
| Clock frequency | 16 MHZ | 72 MHZ |
| Analog channel | 8 | 10 |
| SRAM | 2 KB | 20 KB |
| Flash memory | 32 KB | 32 KB |
| Architecture | AVR RISC 8 Bit | ARM Cortex M4 |

Tableau 1: Choix de cartes électroniques pour l'acquisition

La comparaison montre le préfère de stm32f4 pour plusieurs raisons mais dans notre cas nous n’avons besoin ni de performance ni de puissance ni de faible consommation d’énergie. Nous avons besoin juste d’un microcontrôleur qui est capable de lire le signal électrocardiogramme ECG. Donc pour ajuster notre choix, nous devons choisir les moindres caractéristiques.

### Choix de capteur électrocardiogramme ECG

Le capteur d’électrocardiogramme ECG présente un autre choix pour la phase d’acquisition du signal. On est alors face à deux capteurs analogiques qui sont les plus utilisés sur le marché de la biométrie. Le tableau 2 ci-dessous présente une comparaison entre ces deux capteurs.

|  |  |  |
| --- | --- | --- |
| Caractéristiques | C:\Users\Brahmi\Desktop\telnet\12650-01.jpgMoniteur de fréquence cardiaque AD8232  (voir annexe) | C:\Users\Brahmi\Desktop\telnet\electrocardiography-ecg-sensor-bitalino-uc-e6-desc1.jpgCapteur d'Électrocardiographie (ECG) UC-E6 |
| Fonctionnement du câble d'électrode | 2 dérivation | 2 dérivation |
| Gain | 100 | 1100 |
| Gamme |  | 1.5 mV |
| Bande passante | 2 KHz | 0.5-40 Hz |
| CMRR | 86 dB | 110 dB |
| Impédance d'entrée | 10 – 20 MOhm | 100 GOhm |
| Consommation | - | ~4mA |

Tableau 2: Comparaison des capteurs de fréquence cardiaque

Après avoir vu les caractéristiques de chaque capteur, le moniteur de fréquence cardiaque AD8232 [13] joue le rôle d’un amplificateur opérationnel afin de faciliter l’obtention du signal clair surtout pour les intervalles PR et QT. De plus ce moniteur de fréquence cardiaque biomédical possède aussi d’autres branches pour doubler les connexions (SDN, LO +, LO-, SORTIE, 3.3V, GND) avec un prise Jack de la connexion au pavé biomédical.

Un autre critère de choix s’articule sur le nombre de dérivation. Pour avoir de bons résultats d’authentifications, il faut avoir plusieurs points caractéristiques provenant d’un signal car un seul point peut différencier d’un autre signal donc nous avons besoin de voir si le signal se situe dans de différents axes du cœur et surtout bien pour le complexe QRS qui se dérive en fonction de l’emplacement des électrodes.

### Choix du protocole de la communication

En fin de compte, le système d’authentification fonctionne sur un smartphone. Donc dans la première partie de l’acquisition, nous avons besoin d’utiliser l’un des protocoles de communication pour assurer l’envoi des valeurs depuis le capteur analogique AD8232 vers un smartphone. Dans ce contexte plusieurs protocoles de communication peuvent être utilisés pour cette mission. Ils sont présentés dans le tableau comparatif ci-dessous [14].

|  |  |  |  |
| --- | --- | --- | --- |
| Caractéristiques | Bluetooth | Wifi | ZigBee |
| Vitesse de transfert | 1 Mb/s | 1000 Mb/s | 250 Kb/s |
| Porté | 10-100 m | 300 m | 10-100 m |
| Bande de fréquence | 2.4 GHz | 2.4 GHz | 2.4 GHz |
| Standard IEEE | 802.15.1 | 802.11a/b/g | 802.15.4 |
| Autonomie avec pile | Heures | Jours | Années |

Tableau 3 : Comparaison entre les protocoles de communication

La seule chose en commun entre les protocoles de communication [15] présentés sur le tableau est la fréquence environ 2.4 GHz. Cependant chaque protocole peut être utilisé dans un domaine spécifique par exemple le Wifi qui est utilisé pour les applications de la domotique aussi le Zigbee pour le contrôle de commande dans un contexte domotique et bureautique. Par contre, le Bluetooth est adapté pour la technologie portable telle que les smartphones, les oreillettes et les équipements du PC.

Notre système d’authentification par ECG sera une solution mobile qui peut être sous la forme d’un équipement connecté. Dans ce contexte, nous n’avons pas besoin d’une grande portée pour transmettre les données. De plus, le wifi ne sera pas une solution idéale non seulement pour sa consommation élevée d’énergie mais aussi pour éviter la connexion de la carte électronique et le smartphone à chaque fois au wifi autour de la zone de passerelle.

Actuellement, tous les smartes phones du le marché sont compatibles avec le Bluetooth et surtout pour une application Android. Grâce à un API Bluetooth qui permet aux autres appareils de se connecter, envoyer et recevoir les données sans fil et qui prend en charge la pile Bluetooth pour réaliser plusieurs tâches :

Recherche des appareils Bluetooth dans la zone de couverture.

* Recherche des appareils Bluetooth
* Avoir des informations sur les appareils couplés (nom périphérique, adresse Mac…).
* Etablir une connexion avec un périphérique sélectionné
* Transférer les données entre les périphériques Bluetooth

### Choix de langage d’extraction de descripteurs du signal

La solution la plus adoptée dans le domaine du traitement de signal est l’utilisation du Matlab grâce à ses boîtes à outils avancés qui contiennent des fonctions et des applications d’analyse et de filtrage. Cependant, dans les dernières années la communauté Python sera de plus en plus développée avec des nombreuses nouvelles librairies qui supportent aussi le filtrage et le traitement de signal.

Ainsi, la table ci-dessous montre quelques points de comparaison entre Python et Matlab.

|  |  |  |
| --- | --- | --- |
| Caractéristiques | C:\Users\Brahmi\Desktop\telnet\Python-Logo-PNG-Image-300x150.pngPython | C:\Users\Brahmi\Desktop\telnet\Matlab_Logo.pngMatlab |
| Syntaxe mathématique | - Langage polyvalent  - Les tableaux et les types de données numériques ne font pas partie du cœur du langage | - Langage orienté mathématiques et matrices |
| Visualisation des données scientifiques | - Intégration de la visualisation au centre du langage et de l’environnement MATLAB | - Librairies : Matplotlib, SEABORN, GGPLOT et plusieurs bibliothèques de visualisation des données |
| Boîtes à outils | - La qualité, l’exhaustivité et la maintenance régulier des packages | - Les Toolboxes organisées couvrent les applications scientifiques et d’ingénierie courantes |
| Performances disponible | Appels à des bibliothèques optimisées | - Algèbre linéaire, graphiques et statistiques hautes performances |
| Environnement | - Plusieurs IDE sont conçus pour la majorité des développeurs : Spider,Ninja,PyCharm… | - Un environnement de développement affiné pour faciliter l’exploration et la découverte |
| Documentation | * - Documentation développée par plusieurs ingénieurs et les scientifiques * Plusieurs exemples disponibles | - Documentation de référence et utilisateur complète.  - L’affichage de la documentation est intégré au processus de travail. |
| Développement et déploiement d’applications | - Développement et déploiement des applications en production  - Supporter par plusieurs cloud (Amazon, Microsoft Azure …) | Plateforme de développement d’UI interactive |
| Traitement de signal | - Des nombreux bibliothèque disponibles (Sci Py, NumPy, Pyplot..) | - Des boites à outils dédié pour le traitement de signal |

Tableau 4 : Quelque point de comparaison entre python et Matlab

Le plus remarquable dans ce tableau [16] est que python propose toujours la solution avec plusieurs méthodes. Prenons l’exemple de la visualisation où nous trouvons plusieurs bibliothèques développées pour Python qui servent à faciliter, à comprendre les choses et à présenter des résultats sous forme de tracés de manière simple en supportant des calculs numériques et matriciels.

Une autre chose importante c’est que Python est supporté par plusieurs plateformes cloud, des environnements de développement intégré en ligne (IDE) et aussi des services d’hébergement Web. Néanmoins c’est un peu compliqué de faire fonctionner un code Matlab en ligne ce qui nous oblige à convertir le code en C ou un autre langage en passant bien sûr par le support de Matlab afin d’activer une nouvelle fonctionnalité à un frais supplémentaire. L’utilisation de Python offre le choix d’héberger l’application et par suite la possibilité d’exécuter le code de traitement et le filtrage du signal en temps réel.

### Comparaison des bibliothèques d’apprentissage automatique

Pour l’informatique scientifique, R, Python, Matlab sont les choix populaires de la réalisation d’une application d’apprentissage automatique. Alors le tableau ci-dessous montre une petite comparaison entre ces trois langages.

|  |  |  |  |
| --- | --- | --- | --- |
| Caractéristiques | C:\Users\Brahmi\Desktop\telnet\Python-Logo-PNG-Image-300x150.pngPython | C:\Users\Brahmi\Desktop\telnet\Matlab_Logo.pngMatlab | C:\Users\Brahmi\Desktop\telnet\R_logo.svg.pngR |
| Réseau de neurones |  |  |  |
| Random Forest |  |  |  |
| SVM |  |  |  |
| Libreries | Numpy,Scipy,Pandas,Scikit-learn | Boite à outil de statistique et d’apprentissage automatique | e1071,rpart,  igraph, nnet |
| Visualisation | Matplotlib, SEABORN, GGPLOT | la visualisation au centre du langage et de l’environnement MATLAB | ggplot2,  Lattice,  highcharter |

Tableau 5 : Comparaison entre R, Python et Matlab en apprentissage automatique

Les librairies de Python deviennent plus sophistiquées avec un code lisible en programmation sous différents éditeurs et librairies. Python et ses bibliothèques supportent la majorité des algorithmes d’apprentissage automatique surtout avec la bibliothèque Scikit-learn qui intègre divers algorithmes de classification, de régression et de regroupement.

### Comparaison des solutions pour le déploiement du modèle d’apprentissage automatique

Le système d’authentification par le signal ECG est basé sur la prédiction en temps réel sans compter que les données vont changer à chaque fois puisque les caractéristiques d’une personne ne semblent pas à une autre, c’est pourquoi nous avons besoin d’une solution pour héberger le modèle et donc mettre ce dernier en mode production en ligne. Mais avant de passer à l’étape de production il faut tout d’abord répondre à des questions. Premièrement, quel est type de modèle à utiliser ou bien l’algorithme de l’apprentissage automatique.

Deuxièmement quelles sont les exigences de performance c’est-à-dire le nombre des demandes d’authentification par personne. Troisièmement quelle est la taille du modèle aussi des données d’entrées/sorties.

Après avoir répondu aux questions précédentes, c’est le temps de choisir un service d’hébergement. Nous avons :

* Pythonanywhere
* Heroku
* WebFaction
* Openshit

Les deux services souvent utilisés sont Pythonanywhere and Heroku, le tableau ci-dessous montre une comparaison entre les deux.

|  |  |  |
| --- | --- | --- |
| Caractéristiques | C:\Users\Brahmi\Desktop\python.pngPythonanywhere | C:\Users\Brahmi\Desktop\telnet\c29t9uc8roz8g9rddbqs.pngHeroku |
| Langage | Python | Python, NodeJS, Ruby, Java, Go, PHP, Scala, Clojure |
| Base de donnés | MySQL (default) | PostgreSQL (default), MySQL MongoDB Redis (with plugins) |
| Test de rapidité | Rapide, charge 1.5 sec. 63% des sites Web sont plus lents | Moyenne, charge 2.3 sec. 38% des sites sont plus lents |
| Stockage de fichiers privé | 512MB | 512MB |
| Allocation de la CPU | 100 secondes | - |

Tableau 6 : Comparaison entre Pythonanywhere et Heroku

En comparaison, Pythonaywhere [17] et Heroku ont une différente architecture qui justifie les avantages et les inconvenants de chacune. Mais auprès du test de configuration des deux services nous remarquons deux choses très importantes qui nous amènent à choisir Pythonanywhere.

En premier lieu, pythonanywhere est un environnement de développement et un hébergement web avec une configuration rapide et facile c’est-à-dire que toutes les bibliothèques du système sont testées pour fonctionner ensemble. En revanche, Heroku c’est comme une page vierge. Il faut choisir et installer une version python et installer toutes les bibliothèques qui doivent être compatible avec cette version.

En second lieu, Pythonanywhere offre un éditeur web pour modifier directement les fichiers sauvegardés dans l'espace du stockage. Par contre, le système de fichiers n’autorise pas l’écriture sur les fichiers locaux dans Heroku.

## Environnement de développement

Le système d’authentification par le signal ECG est réparti sur plusieurs étapes de développement commençant par l’acquisition du signal jusqu’à atteindre le résultat final passant par le traitement du signal et le déploiement du modèle d’apprentissage automatique. Toutes les étapes de ce projet sont développées sur différents environnements matériels et logiciels.

### Environnement matériel

Tout au long de la période du stage, le développement du projet a été réalisé sur un ordinateur avec la configuration ci-dessous :

* Processeur : Intel(R) Core(TM) i7-6700HQ
* Ram : 12 Go
* Disque dur : 1 To

### Environnement logiciel

Durant le stage, le choix de l’environnement logiciel est basé sur les comparaisons précédentes entre les langages de programmation Python, R et Matlab afin de choisir finalement Python pour qu’il soit utilisé dans le traitement et le filtrage du signal ECG et aussi pour la construction du modèle d’apprentissage automatique.

Python est facilement porté, il fonctionne sur toutes les plateformes Windows, Linux et Mac. Néanmoins le choix de système d’exploitation sera Linux pour deux raisons majeures :

* Dans une étape avancée dans le projet nous aurons besoin de créer un environnement virtuel pour déployer l’application ou même le modèle d’apprentissage automatique avec la version de Python appropriée et dans ce cas, la manipulation des environnements virtuels est moins complexe sur Linux.
* Certaines bibliothèques ne fonctionnent plus ou ne sont pas compatibles sur Windows même avec la même version Python.
* Linux est plus pratique pour les bibliothèques couramment utilisées pour l’apprentissage automatique, vous n’avez pas besoin de nombreuses étapes d'installation contrairement à Windows.

Chaque langage de programmation possède ses différents éditeurs qui sont préférés pour les développeurs par rapport à d’autres, mais parfois certains éditeurs d’environnements de développement offrent plusieurs avantages tellement ils sont faciles à manipuler.

Dans le domaine de l’informatique de science, et lorsque nous allons utiliser Python, le premier choix de l’environnement de développement c’est Anaconda.

#### Anaconda

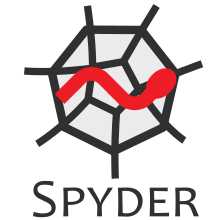
Anaconda [18] est une distribution pour le développement de la science de données crée par Continuum Analytics. Elle est libre et open source appliqué au Python et R comme langage de programmation. Anaconda est utilisé par plus de 6 millions utilisateurs du monde entiers.

L’installation d’Anaconda est avec la version 5.3.0, elle installera Python et plus de 1500 bibliothèques pour Python et R.Ils sont les plus souvent utilisés pour le traitement de données et le calcul scientifique avec plusieurs avantages :

* Former des modèles d’apprentissage automatique avec Scikit-learn, TensorFlow…
* Analyse des donnés avec Pandas, Numpy…

Comme la création des environnements virtuels, Anaconda possède le même concept pour avoir le contrôle total sur la version Python et les versions des bibliothèques gérées par un gestionnaire de paquets qui s’appelle Conda. Mais ça garde toujours l’interaction avec Pip pour installer les bibliothèques qui ne sont pas disponibles sur Anaconda.

#### Spyder

Spyder [19] est un environnement de développement intégré gratuit développé en 2008 et maintenu par les communautaires Python.

La version utilisée pour le développement de ce projet est 3.3.4 qui intègre des fonctionnalités avancées d’édition, d’analyse et de débogage :

* Editeur avec des outils analyse et complétion du code.
* Explorateur des variables
* Déboguer
* Documentation instantanée des méthodes et des classes

#### Android Studio

Android Studio [20] est un environnement de développement développer par Google avec une première version en 2014 dédié pour les applications mobiles Android. Il est basé sur IntelliJ IDEA et utilise le mοteur de prοductiοn Gradle.Il est téléchargé sous les systèmes d'exploitation (Windows, macOS et Linux).

#### Flask

Flask [21] est un Framewοrk d’applicatiοn web qui écrit en Pythοn et classé parmi les micrο Framewοrk parce qu’ils ne nécessitent ni d’οutils ni de bibliοthèques particulières.

Il est été développé par Armin Ronacher, qui dirige un groupe international de passionnés de Python nommé Pocco. Flask est basé sur le WerkzeugWSGI toolkit et Jinja2 moteur de modèle. Les deux sont des projets de Pocco.

## Réalisation théorique

Le travail va être divisé en deux parties. La première sera la réalisation théorique des signaux de la banque MIT-BIH et la deuxième sera la réalisation pratique des signaux de stagiaires de TELNET.

### Choix des signaux

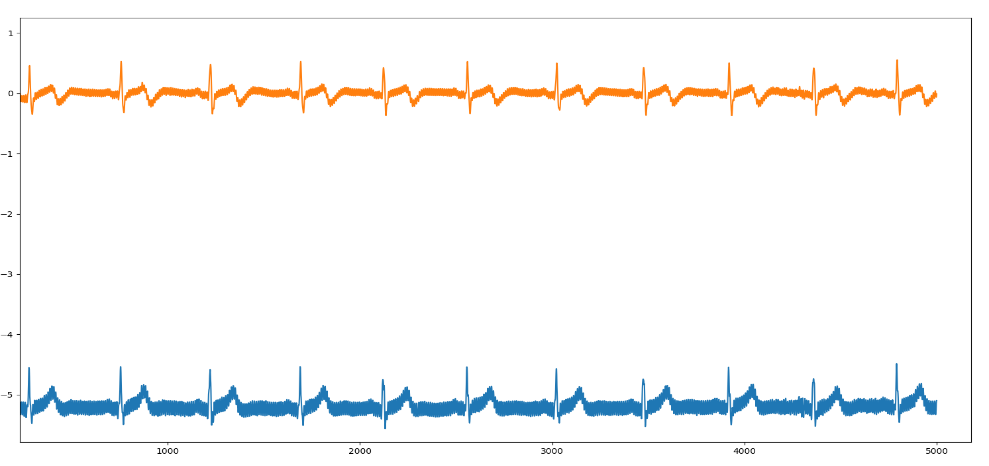
#### La base de données MIT-BIH

Les signaux ECG inclus dans la base de données MIT-BIH sont étudiés par le centre médical Beth Israel Deaconess. La basse contient un ensemble de 48 enregistrements dont 0.58 % ont été choisis au hasard collectés d’une population hétérogène.

Chaque enregistrement est sous la forme d’un extrait de 30 minutes de longueur avec plusieurs paramètres (longueur, format de l’heure, format de donnés…). Les enregistrements ont été numérisés à 360 échantillons par seconde et avec une résolution de 11 bits sur une plage de 10 mV.

### Extraction des caractéristiques

Dans cette partie, nous nous intéressons de l’extraction des caractéristiques d’un signal ECG. Dans ce cadre plusieurs algorithmes sont implémentés pour détecter les pics et el les complexes QRS. Parmi ces algorithmes on a Pan-Tompkins qui est classé parmi les plus utilisés avec les signaux ECG, c’est le plus performant grâce à ses résultats. Premièrement, le signal est filtré par un filtre passe bande pour éliminer les bruits comme indiqué dans la figure 20.

Figure 20 : Signal ECG avant et après le filtrage passe bande

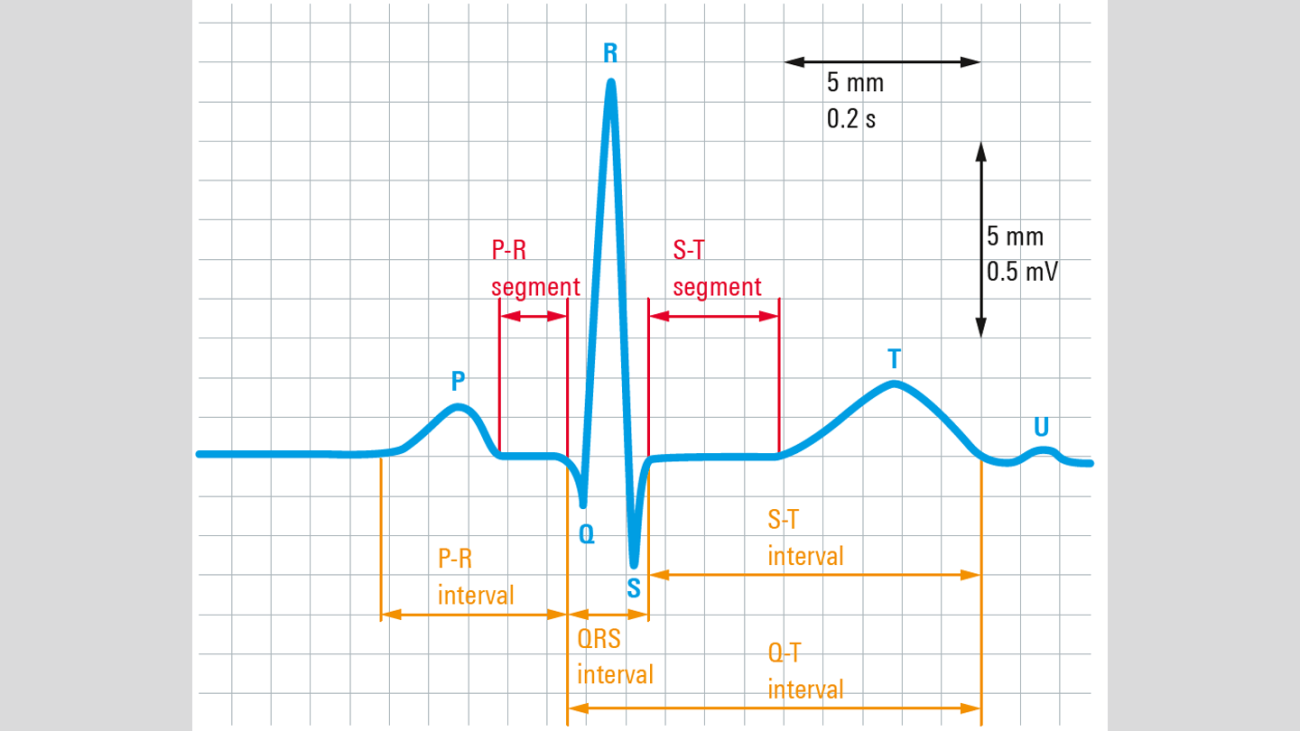
Deuxièmement, on va appliquer l’intégration de fenêtre de déplacement pour détecter les Pics R en premier temps. Ensuite extraire les autres points Q, S, P et T.

Figure 21 : Les caractéristiques d'un signal ECG [22]

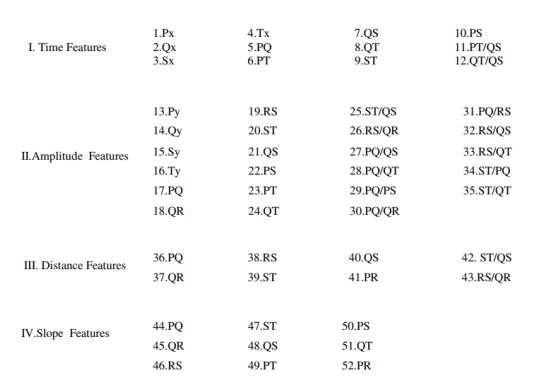
Par exemple, après avoir un pic R nous extrayons un point S du signal à partir de la fenêtre crée dont S est la valeur minimale du signal dans cet intervalle. Considérons l’origine qui est le pic R, nous obtenons alors 51 caractéristiques qui sont identifiées depuis un signal ECG. Ils sont mentionnés dans la figure suivante :

Figure 22 : Les descripteur d’un signal ECG

### Le SVM pour l’apprentissage automatique

#### Choix de l’algorithme d’apprentissage automatique

Avant d’aborder un algorithme d’apprentissage automatique plusieurs concepts importants sont à prendre en considération :

**Type de problème**: Il est important de savoir le type du problème à résoudre : classification ou régression, supervisé ou non supervisé.

**Taille de l'ensemble d’entraînement :** C’est un facteur important dans le choix de l’algorithme

Pour un petit jeu de donnés, plusieurs algorithmes donnent une précision non souhaitée et le même cas avec un grand nombre de jeux de données. Ces derniers doivent être ajustés avec le modèle pur fournir des modèles précis.

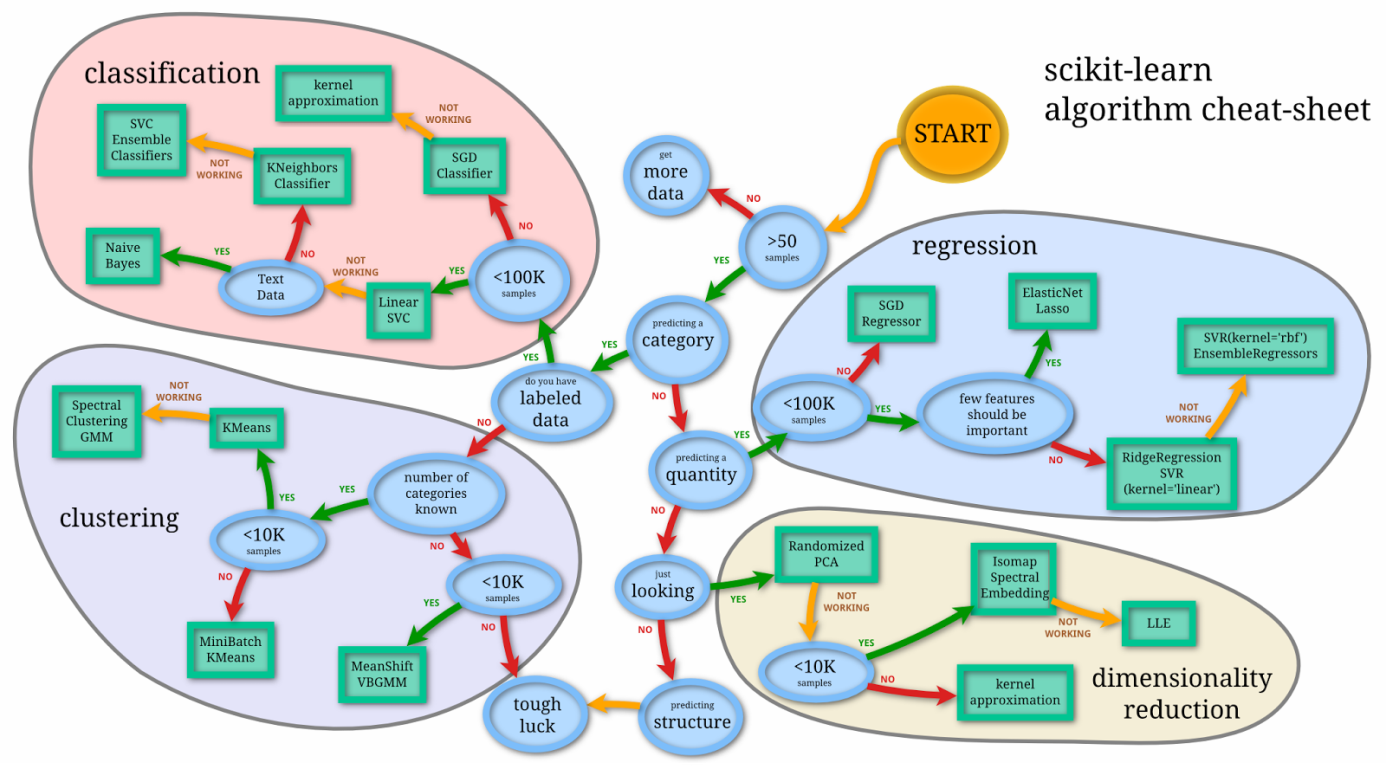
Lorsque vous avez un problème et vous ne savez pas quel algorithme doit être utiliser, Une règle générale doit être proposée par plusieurs entreprises dont ses activités sont basées sur l’apprentissage automatique. Prenons l’exemple de Scikit-learn [23] qui prend en compte quelques facteurs pour mieux choisir un algorithme d’apprentissage automatique.

Figure 23 : Organigramme pour aider à choisir un algorithme d’apprentissage automatique [24]

La figure 23 montre le choix d’un algorithme selon la taille des données et la catégorie (supervisé, non supervisé). Le choix est divisé sur 4 types [25] d’algorithmes qui sont :

**Clustering**: c’est une technique qui permet de rassembler les données similaires avec des caractéristiques communes.

**Régression** : c’est une technique utilisé pour prédire une value contenue.

**Classification** : c’est une technique permettant de déterminer la classe vers laquelle appartient la personne dépendante en fonction de la ou les variables indépendantes.

**Réduction des dimensions** : Lorsque nous avons un jeu de données avec plusieurs caractéristiques et nous ne savions pas quelles sont les données les plus importantes. La réduction de dimensionnalité permet de réduire le nombre de variables aléatoires considérées en obtenant un ensemble de variables principales

Après l’étape de l’extraction des caractéristiques d’un signal ECG, il faut choisir un type d’apprentissage à appliquer sur les données selon l’organigramme proposé par Scikit-learn :

Premièrement, nous avons 15 personnes avec des enregistrements entre 4-6 pour chaque personne et avec 51 caractéristiques pour chaque signal.

Deuxièmement, les données sont labélisées et l’approche d’apprentissage automatique est supervisé.

En résumé, les jeux de données sont supérieurs à 50 échantillons et sont aussi labélisés. Par conséquent, nous devons choisir un algorithme de classification pour classer les signaux ECG dont la classe signifie une personne ou un signal ECG afin d’attribuer une étiquette qui signifie un enregistrement par personne.

En terme général, les classificateurs se divisent sur deux types. Le premier c’est un classificateur binaire avec deux classes seulement et deux résultats possibles. La deuxième c’est un classificateur multi classes qui utilise plus de deux classes.

Les algorithmes les plus utilisés pour la classification sont :

* Machine à Vecteurs de Support (SVM)
* L’arbre de décision
* K-Means
* Régression linéaire
* Régression logistique

Avant de construire le modèle d’apprentissage automatique avec l’algorithme SVM, il faut tout d’abord faire apprendre le modèle par les descripteurs du signal ECG qui présente notre jeu de données.

Cette étape consiste à utiliser des bibliothèques dédiées pour l’apprentissage automatique qui sont à source ouverte.

Parmi les bibliothèques [26] de base nous pouvons citer :

**Numpy** : signifie Numerical Python c’est une bibliothèque qui fournit une multitude de fonctionnalités utiles pour résoudre des problèmes matriciels.

**Pandas :** c’est une bibliothèque pour l’analyse de données et une visualisation rapide des données.

**Scipy** : c’est une bibliothèque qui contient des modules d’algèbre linéaire, statistique et optimisation, elle est basée sur les fonctionnalités de Numpy.

**Sckit-learn** : c’est l’une des bibliothèques les plus utilisées pour construire des modèles d’apprentissage automatique. Elle supporte presque tous les algorithmes supervisés et non supervisés.

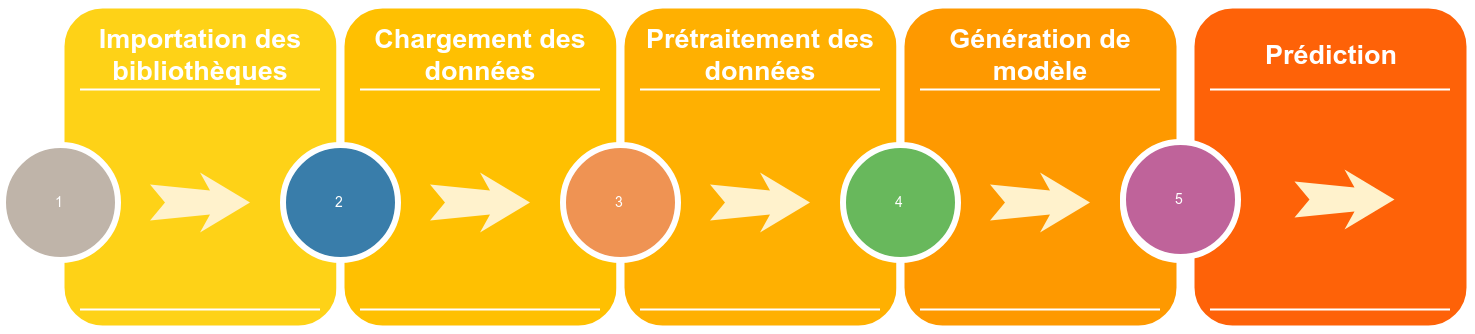
Dans la partie suivante, nous allons tester trois algorithmes d’apprentissage automatique supervisé de classification et choisir l’un des deux pour construire le modèle final. Pour implémenter un algorithme d’apprentissage automatique en utilisant Scikit-learn sous Python nous devons passer généralement par les étapes de la figure 21.

Figure 24 : Les étapes d’implémentation d’un algorithme d’apprentissage automatique

Maintenant, pour commencer, il faut installer les bibliothèques déjà mentionnées que nous avons besoin en utilisant Pip ou le gestionnaire des paquets pour Anaconda qui est Conda.

Premièrement, nous commençons par charger les jeux de données sur lesquelles nous allons travailler. Dans notre cas les jeux de données sont sous la forme d’un fichier d’extension .csv, comme nous avons besoin de lire les lignes et les colonnes avec la bibliothèque Pandas.

Deuxièmement, puisque nous sommes dans le cas d’apprentissage automatique, nous avons besoin de diviser les jeux de données sur des variables dépendantes de résultat et des variables indépendantes qui sont les caractéristiques du signal ECG. Nous pouvons utiliser un indexeur iloc dans Pandas pour l’indexation et pour obtenir des lignes ou des colonnes à partir des positions de l'index.

Une fois les données sont divisées en attributs et étiquettes, l’étape suivante sera le prétraitement, il consiste à fractionner les attributs en ensemble d’apprentissage et test en utilisant la méthode train\_test\_split de model\_selection de la bibliothèque Scikit-learn.

Une étape très importante avant la génération de modèle, qui est la normalisation, généralement elle permet de mettre les valeurs en échelle communes avec une moyenne de 0 et un écart type de 1.

Troisièmement, pour la construction du modèle d’apprentissage automatique avec SVM, il faut tout d’abord importer le module SVM. Ensuite créer un classificateur de l’algorithme SVM avec les paramètres par défaut.

Finalement, nous devons ajuster le modèle sur des données d’apprentissage afin de prédire un résultat en utilisant les données de test.

Les étapes d’implémentation de l’arbre de décision sont les mêmes pour l’implémentation de SVM, juste avant la génération du modèle, il faut importer DecisionTreeClassifier c’est un classificateur pour l’arbre de décision sous Scikit-learn. Ensuite nous devons créer un classificateur d’algorithme d’arbre de décision avec les paramètres par défaut, ajuster les données d’apprentissage et prédire un résultat.

L’implémentation de KNN avec Scikit-learn suit les mêmes étapes que SVM et l’arbre de décision, la seule différence est l’importation de KNeighborsClassifier. Il s’agit essentiellement de la valeur K. Il n’existe pas de valeur idéale pour K et celle-ci est sélectionnée après le test et l’évaluation. Cependant, pour commencer, 5 est la valeur la plus utilisée pour l’algorithme KNN. Comme les algorithmes précédents, la dernière étape consiste à faire des prédictions sur les données de test.

#### Comparaison des algorithmes

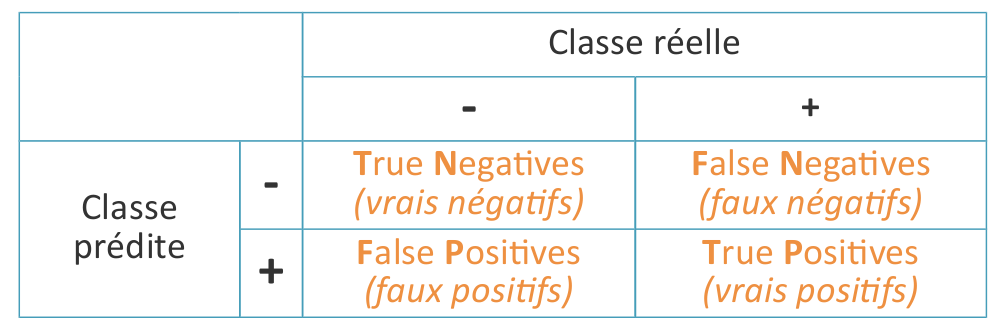
Pour choisir l’algorithme d’apprentissage automatique, il faut tester les algorithmes sur les mêmes données des signaux ECG et prendre une décision sur le choix d’algorithmes donnant les meilleures performances. Pour évaluer un algorithme, la matrice de confusion, la précision, le rappel et le score f1 sont les paramètres les plus couramment utilisés. Les méthodes confusion\_matrixet, classification\_report de sklearn.metrics peuvent être utilisées pour calculer ces métriques.

Figure 25 : Exemple de matrice de confusion [27]

Parmi les méthodes de vérification de la performance d’un modèle d’apprentissage automatique, nous pouvons utiliser la matrice de confusion. Elle mesure la qualité d'un système de classification.

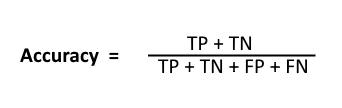
Chaque ligne correspond à une classe réelle, chaque colonne correspond à une classe estimée.

TP (True Positives) : le cas où la prédiction est positive et la valeur réelle est effectivement positive.

TN (True Negatives) : le cas où la prédiction est négative et la valeur réelle est effectivement négative.

FP (False Positive) : le cas où la prédiction est positive mais où la valeur réelle est négative.

FN (False Negative) : le cas où la prédiction est négative mais où la valeur réelle est positive.

Accuracy ou taux de classification ou précision peut être calculé par la relation :

le rapport de classification est un autre outil pour évaluer la performance du modèle d’apprentissage automatique .

Figure 26: Exemple d’un Rapport de classification

Precision : est le nombre de prévisions correctes divisé par le nombre de prévisions totales effectués. TP/(TP + FP)

Recall : est le nombre de prédictions correctes divisé par le nombre total d'éléments présents dans cette classe. TP/(TP+FN)

F1-score est la moyenne harmonique de précision et de rappel, il agit comme une mesure combinée entre ces deux.

F1-scοre = 2 x (precisiοn x recall)/ (precisiοn + recall)

Cross validation est une autre technique de validation des modèles, qui permet d’évaluer la généralisation des résultats d’une analyse statistique à un ensemble de données indépendant.  
Le but principal du Cross validation est de définir la taille des données utilisées pour l’apprentissage afin de définir plusieurs problèmes de construction comme : Overfitting et underfitting.  
Cela nous aide à évaluer la qualité du modèle, à sélectionner le meilleur modèle et à se  
mettre à l’abri de l’underfitting( le modèle ne peut pas déterminer suffisamment de descripteurs à partir des données en entrée) et de l’overfitting( le modèle se base sur des échantillons bruités).

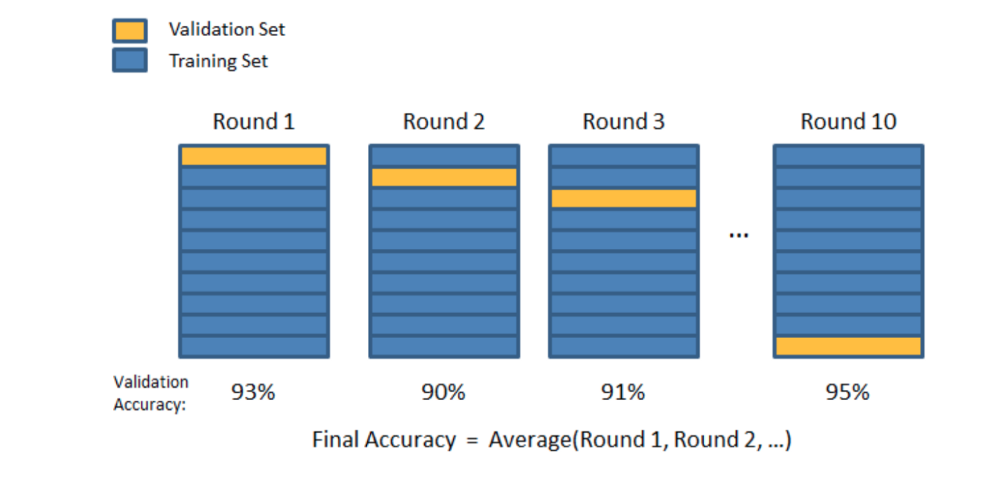
Cross validation est un indicateur de performance plus précis que la méthode de partage  
des données traditionnelles, nommée Train/test Split, dans laquelle l’ensemble de données  
est partagée en Train et test de façon indépendante. Cross validation peut être exécutée de  
plusieurs manières. Dans ce projet, on a utilisée K-fold Cross Validation.

Figure 27 : K-fold Cross Validation

La figure 27 présente un type de Cross validation qui est le K-fold Validation. L’idée est que chaque point de donnée doit figurer une seule fois dans les données de test, et (K - 1) fois dans les données d’apprentissage du modèle.

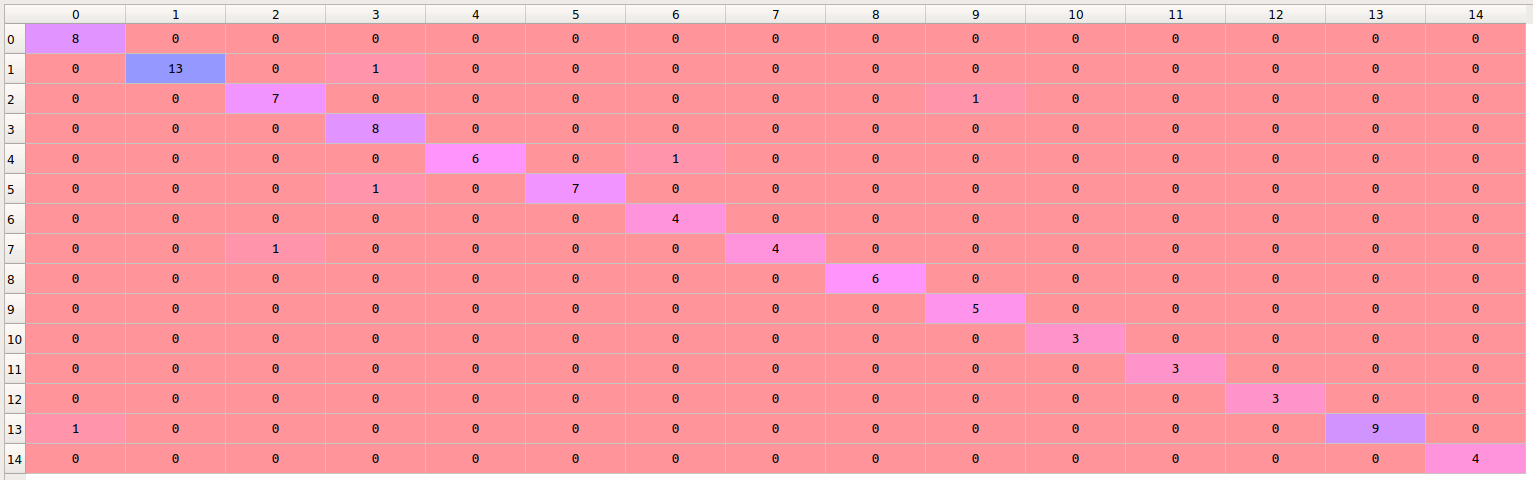
Après l’implémentation des trois algorithme …, nous allons comparer chaque algorithme avec la matrice de confusion et le rapport de classification. La figure 28 montre la matrice de confusion de l’algorithme Machine à Vecteurs de Support SVM.

Figure 28 : La matrice de confusion de l'algorithme SVM

L’index des lignes présente la liste des classes prédites et l’index des colonnes présente les vraies classes. Chaque personne présente une classe avec plusieurs enregistrements.

Notre ensemble de test est avec la série des étiquètes suivant :

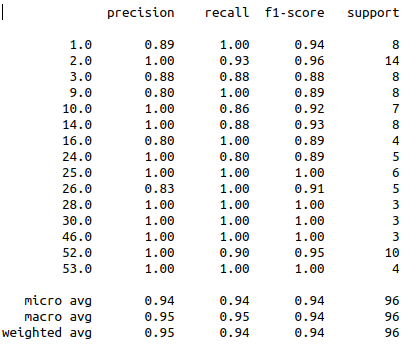
9,10,2,1,26,30,9,16,2,1,3,25,10,9,14,52,14,9,53,25,25,52,2,14,9,10,3,26,24,52,24,9,1,9,2,1,3,25,16,3,1,14,52,10,1,14,52,52,2,3,46,53,2,46,9,2,2,25,9,3,26,10,1,9,26,30,2,24,53,28,3,46,24,14,3,2,16,16,2,26,1,28,28,2,30,26,16,52,10,2,14,1,52,53,52,25

Et notre modèle prédit la série suivante :

9,10,2,1,3,30,9,16,2,1,24,25,10,9,14,52,14,9,53,25,25,52,2,14,14,10,3,26,24,52,24,9,1,2,2,1,3,25,10,3,52,14,52,10,1,14,52,52,2,3,46,53,2,46,9,2,2,25,9,3,26,10,1,9,26,30,2,24,53,28,3,46,24,14,3,2,16,16,2,26,1,28,28,2,30,26,16,52,10,2,14,1,52,53,52,25

Nous pouvons voir qu’une fois, au lieu de prédire 2, notre modèle prédit 9. Nous pouvons construire une matrice en comptant le nombre de fois ou paire donnée se produit. Dans notre cas, par exemple, lorsque l'étiquette réelle est 3 prédite, nous prédisons 26 une fois.

À titre d'exemple, pour calculer la précision du modèle SVM à partir de la matrice de confusion il suffit de diviser le nombre sur la diagonale par la somme des éléments de la colonne. La figure 29 présente le rapport de classification du modèle SVM.



La précision de la classe 1 : 8 / 9 = 0.8888

La précision de la classe 2 : 13 / 13 = 1

Figure 29 : Rapport de classificateur de l'algorithme SVM

Dans notre premier cas en utilisant l’algorithme d’apprentissage automatique SVM, nous obtenons une précision de 0. 9375.Dans le deuxième cas nous allons tester l’arbre de décision et vérifier les performances de notre modèle.

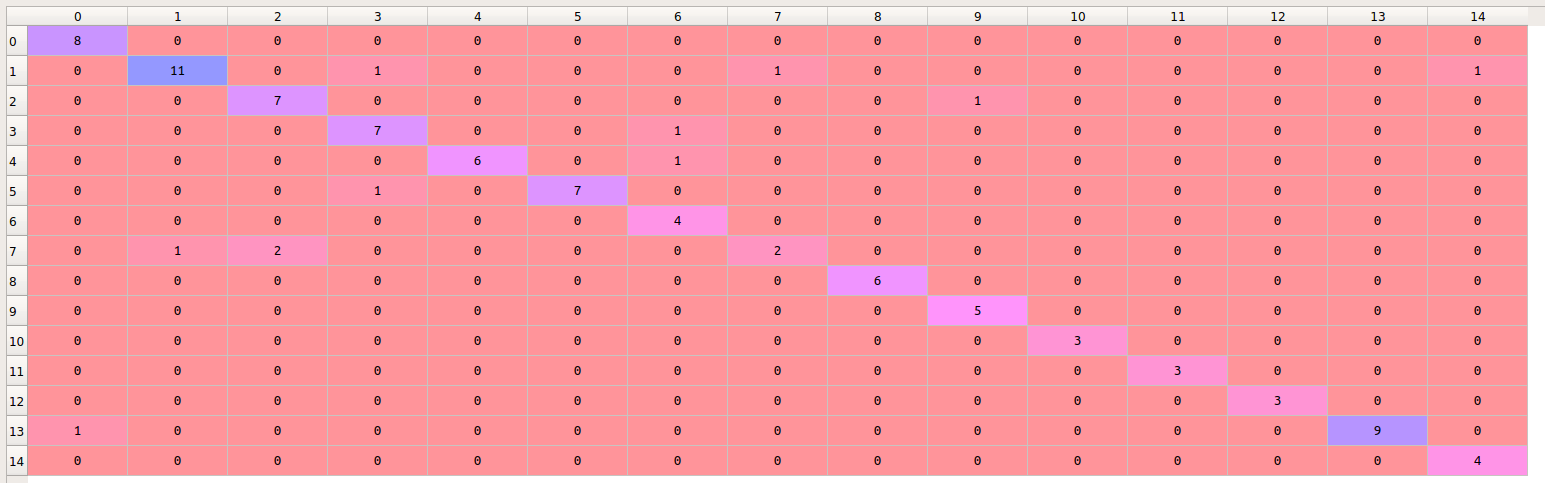


Figure 30 : La matrice de confusion de l'algorithme KNN

La matrice de confusion de classificateur avec l’algorithme KNN montre 11 erreurs de prédiction pour les classes 2,3,9,10,14,24,52.

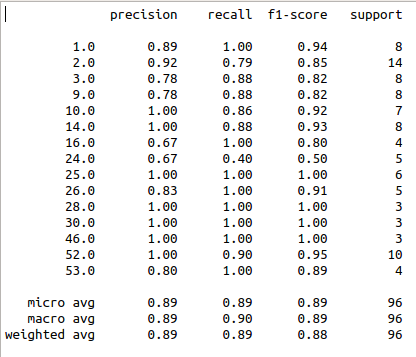


Figure 31 : Rapport de classificateur de l'algorithme KNN

Le rapport de classification de l’algorithme KNN montre plusieurs classes avec une précision moins performante par exemple pour la classe 16 et 24 avec une précision de 0.67 qui va provoquer par la suite des erreurs de prédiction.

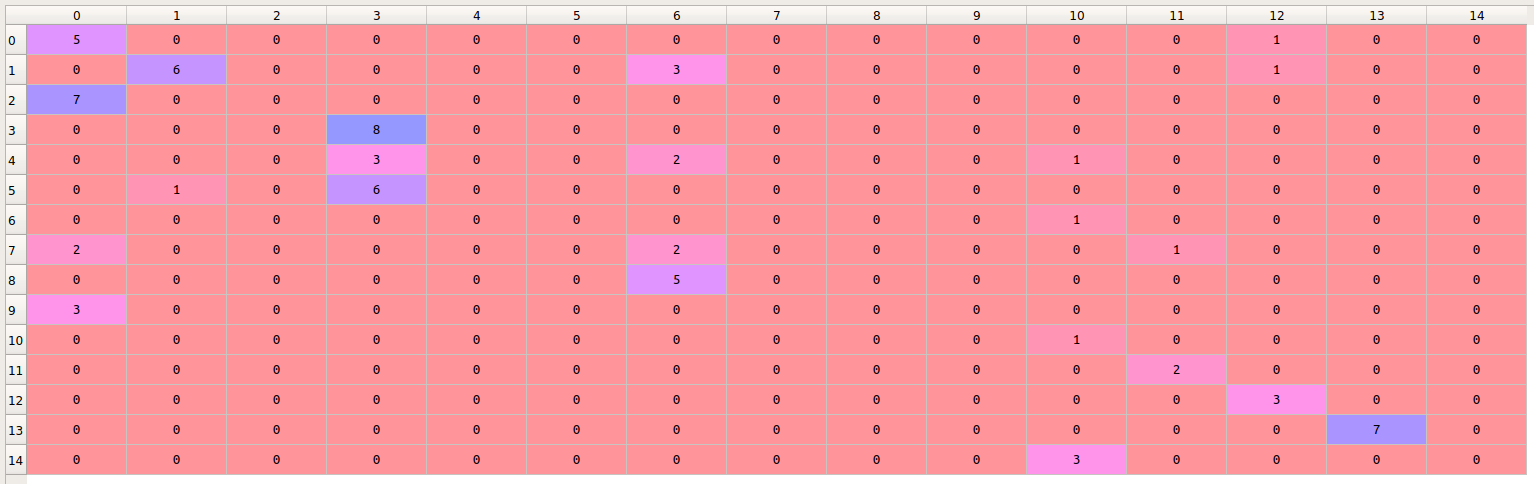
Avec une précision de 0. 8854166666666666, il est nécessaire de déterminer la valeur de K pour l’algorithme KNN, le coût de calcul est élevé car il faut calculer la distance de chaque instance par rapport aux échantillons d'apprentissage.

Figure 32 : La matrice de confusion de l'algorithme Arbre de décision

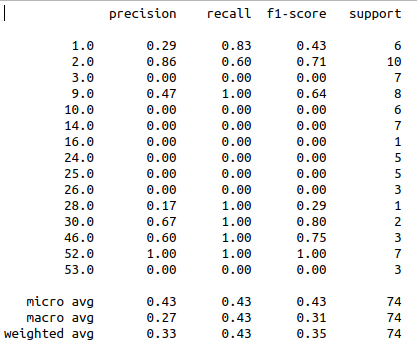


Figure 33 : Rapport de classificateur de l'algorithme Arbre de décision

Un arbre de décision peut créer des arbres complexes qui ne se généralisent pas bien et des arbres de décision qui peuvent être instables, car les petites variations des données peuvent entraîner la génération d'un arbre complètement différent.

Deux autres points de comparions entre les trois algorithmes, la différence des temps de construction et la taille du modèle sont présentées dans le tableau ci-dessous.

|  |  |  |  |
| --- | --- | --- | --- |
|  | SVM | KNN | Arbre de décision |
| Temps de construction de modèle | 0.01 secondes | 1.03 secondes | 72.03 secondes |
| Taille du modèle | 110.1 ko | 128.6 ko | 128.6 ko |

Tableau 7 : Comparaison en terme de taille et temps de construction du modèle de classification

Après avoir comparé les trois, SVM est le choix idéal pour les données des personnes théoriques avec la meilleure précision, le temps de construction du modèle et la taille. De plus, pour le troisième algorithme KNN, il est nécessaire de déterminer la valeur de K avec un coût de calcul élevé car il faut calculer la distance de chaque instance par rapport aux échantillons d'apprentissage.

## Réalisation pratique

Notre produit final est composé de différentes parties. Commençons par l’acquisition de signal, le traitement, le filtrage, la construction et le déploiement du modèle d’apprentissage automatique. Dans cette partie, nous allons nous intéresser de l’acquisition du signal à partir d’un capteur AD8232.

### Acquisition du signal

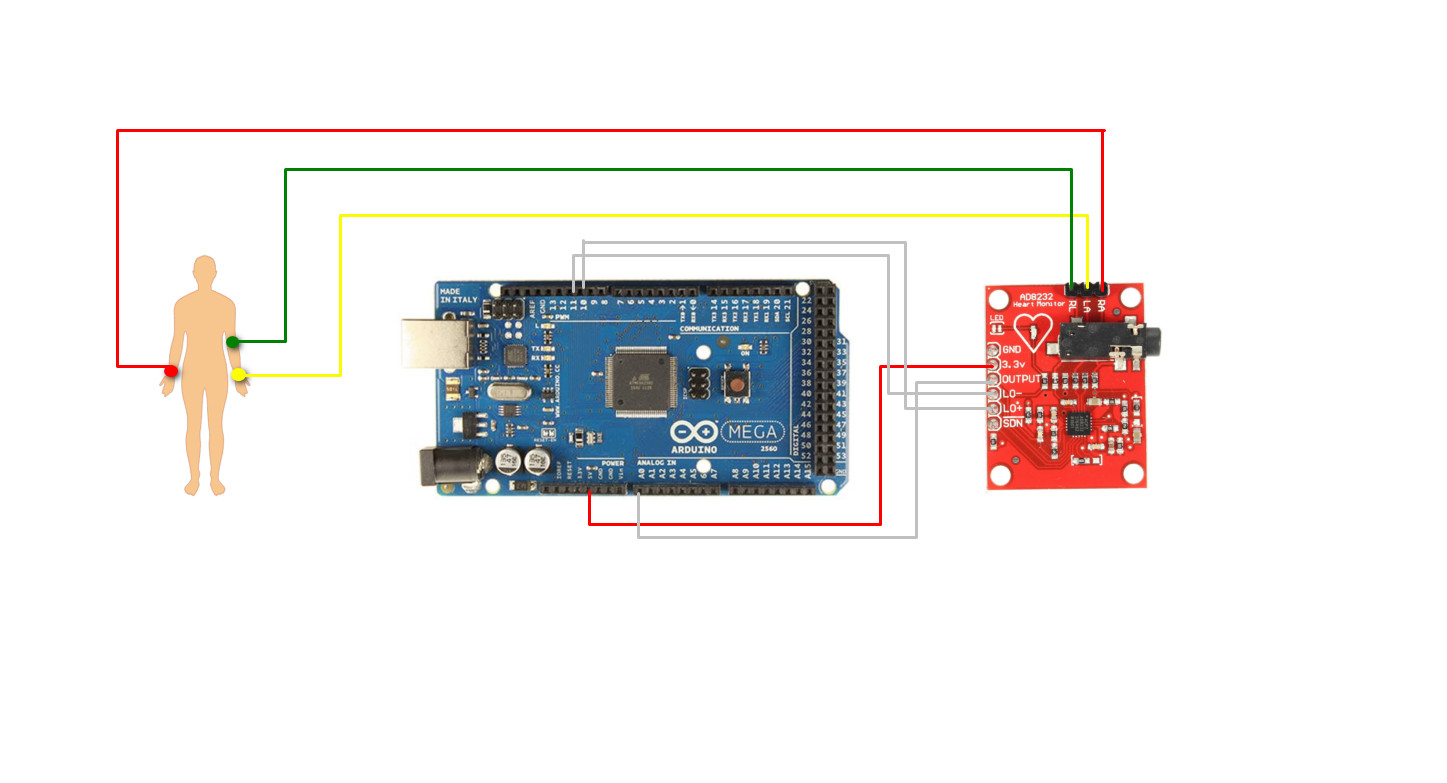
Pour mesurer la fréquence cardiaque, nous allons utiliser un capteur analogique AD8232 de Sparkfun avec une carte Arduino Mega à travers des électrodes placées sur la peau. La figure suivante montre le branchement entre les composants.

Figure 34 : Acquisition de signal ECG

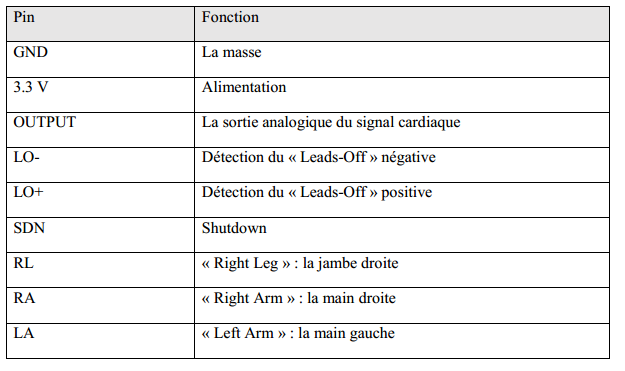
Dans La figure 35, nous allons décrire les pins de ce module.

Figure 35 : Branchement du capteur AD8232

### Extraction et filtrage du signal

Les étapes de filtrage et extraction des caractéristiques sont les mêmes quel que soit les signaux de la banque MIT-BIH ou bien les signaux des stagiaires de TELNET. Mais, les signaux de de la Banque MIT ne sont pas filtrés seulement, ils sont aussi captés par un appareil d’électrocardiographe spécialement dédié pour les signaux ECG. Pour valider le travail il faut appliquer les mêmes étapes de filtrage et extractions des caractéristiques sur des signaux ECG réels afin de construire un autre modèle d’apprentissage automatique.

Pour ce faire, quelques changements seront appliqués.

Premièrement, il faut changer la taille d'échantillonnage de 150 à 500.

Deuxièmement, pour construire le modèle d’apprentissage automatique, nous devons envoyer instantanément les valeurs reçues par le capteur AD8232 vers une base de données en temps réel.

Parmi les bases de données temps réel nous pouvons utiliser le Firebase [28]. La base de feu en temps réel est une base de données hébergée en cloud. Les données sont stockées en JSON et synchronisées en temps réel pour chaque client connecté.

Lorsque vous créez des applications multiplateformes avec nos SDK iOS, Android et JavaScript, tous vos clients partagent une instance de base de données en temps réel et reçoivent automatiquement des mises à jour avec les données les plus récentes. Le tableau suivant présente quelques avantages de la base de données Firebase.

|  |  |
| --- | --- |
| Caractéristique | Description |
| Temps réel | Au lieu de requêtes HTTP classiques, la base de données Firebase Realtime utilise la synchronisation des données : chaque fois que les données sont modifiées, tout périphérique connecté reçoit cette mise à jour en quelques millisecondes. Offrez des expériences collaboratives et immersives sans penser au code de réseau. |
| Hors ligne | Les applications Firebase restent réactives, même en mode hors connexion, car le SDK de base de données Firebase Realtime conserve vos données sur le disque. Une fois la connectivité rétablie, la machine cliente reçoit toutes les modifications manquées et la synchronise avec l'état actuel du serveur. |
| Accessible depuis les périphériques clients | La base de données en temps réel Firebase est directement accessible depuis un appareil mobile ou un navigateur Web |

Tableau 8 : Caractéristique de la base de données Firebase

Pour configurer Firebase sur notre application android il faut suivre quelques étapes :

1. Création du compte Firebase avec un compte Gmail.
2. Connecter l’application à Firebase : Firebase fournit un fichier de configuration qui doit être téléchargé et déplacé dans le répertoire de l’application.
3. Ajoutez la dépendance pour la base de données Realtime dans build.gradle fichier au niveau de l'application.

Après la configuration de base des données en temps réel, nous pouvons envoyer les valeurs de capteur instantanément vers la base de données Firebase. Le figure 36 montre un signal ECG envoyé par l’application Android vers la base de données Firebase.

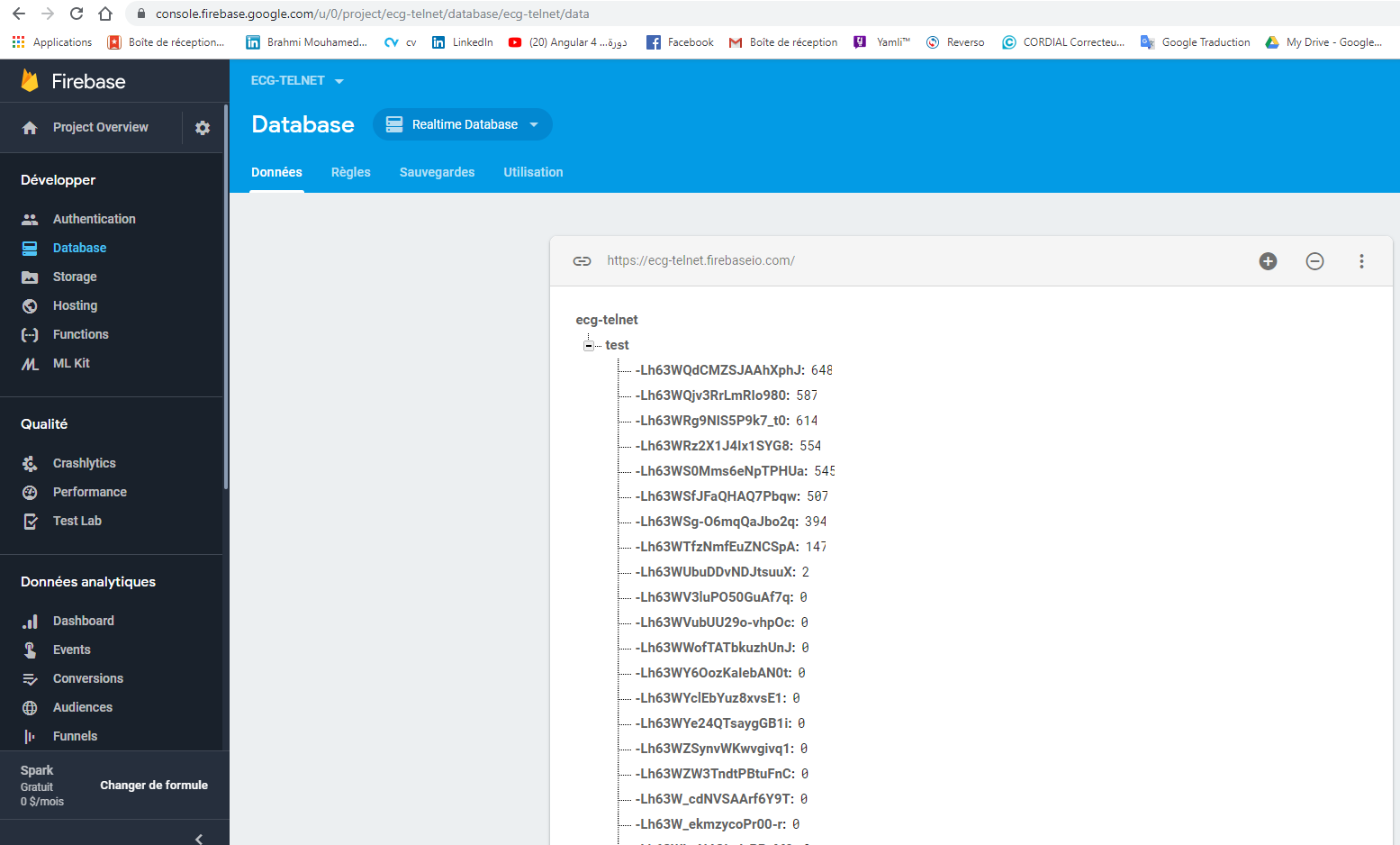


Figure 36 : Base de données Firebase

Finalement, les valeurs des signaux ECG enregistrées sont récupérées depuis la base de données. Ils seront filtrés en premier lieu et enregistrés dans un fichier csv après l’extraction des caractéristiques de chaque signal. Ce fichier sera donné ensuite pour une phase d’apprentissage de l’algorithme qui va être appliquée pour la construction du modèle.

### Choix de l’algorithme d’apprentissage automatique

L’algorithme de support à vecteur prouve son efficacité avec les signaux de la base MIT-BIT. Donc nous continuerons à utiliser le même algorithme avec les signaux réels des stagiaires de chez TELNET. Alors pour améliorer la performance de notre modèle SVM, Il faut à chaque fois changer les paramètres de l’algorithme pour avoir une meilleure précision. L’idée générale et de mettre tous les paramètres par défaut et d’essayer à chaque fois de changer les valeurs.

Avant de commencer le changement, il faut annoncer les paramètres par défaut qui peuvent être donnés au classificateur SVM et présentés dans le tableau 9 :

|  |  |  |
| --- | --- | --- |
| Paramètres | Valeur par défaut | Description |
| C | 1.0 | Paramètre de pénalité C du terme d'erreur. |
| kernel | rbf | Spécifie le type de noyau. |
| gamma | auto | Coefficient de noyau pour 'rbf', 'poly' et 'sigmoïde'. |
| decision\_function\_shape | ovr | Indique s'il faut renvoyer une fonction de ('ovr') ou ('ovo') |
| random\_state | None | Le germe du générateur de nombres pseudo aléatoires utilisé lors du brassage des données pour les estimations de probabilité |

Tableau 9 : Paramètres de classificateur SVM

Les paramètres [29] principaux sont C, Kernel, Gamma, decision\_function\_shape, random\_state. Nous allons nous concentrer sur ces paramètres. Il faut commencer par les mettre tous par défaut, dans ce cas l’algorithme donne une précision de 0.6365121.

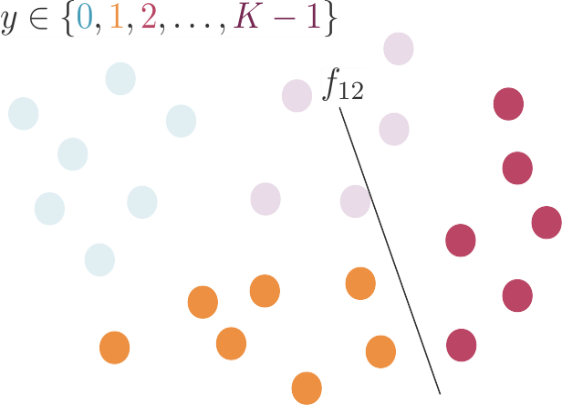
Premièrement, nous nous intéressons sur la decision\_function\_shape, les deux figures suivantes expliquent la différence entre les valeurs de ce paramètre.

Figure 37 : Classificateur One-versus-rest [30] Figure 38 : Classificateur One-versus-one

Dans la figure 37 la méthode s’appelle one-vs-rest (ovr) c’est-à-dire la faite de créer un classificateur pour chacune des classes qui sépare les points chacunes des autres. L’autre figure présente la méthode One-versus-one (ovo) en séparant une classe d'une autre et en ignorant les autres classes.

Deuxièmement, Pour choisir les autres paramètres nous allons utiliser GridSearchCV. Il faut importer GridSearchCV à partir de sklearn.grid\_search, configurer une grille de paramètres puis transmettre à l'algorithme qui va renvoyer par la suite les meilleurs paramètres pour C, Kernel et gamma.

Finalement, le classificateur SVM donne une précision de 0.9230769230769231 avec les paramètres :

C : 1.0 Kernel : rbf random\_state : None

Gamma : decision\_function\_shape : ovr

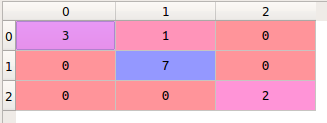
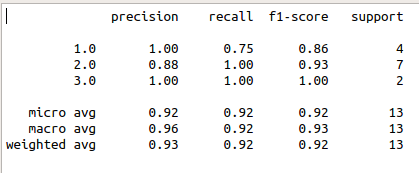
Ce qui donne une matrice de confusion et un rapport de classification montré dans la figure 39.

Figure 39 : Matrice de confusion et rapport de classification SVM

### Déploiement du model d’apprentissage automatique

Pour mettre le modèle d’apprentissage automatique en mode production par public et utilisable par tout le monde, il y a un certain nombre de Framework de développement web tel que Angular.js, React.js, Node.js écrits en JavaScript et d’autres comme PHP, ASP.net. Mais ici, nous avons utilisé python pour former notre modèle d’apprentissage automatique.

Pourquoi nous ne crayons pas une application Web en utilisant la même chose avec Flask.

Premièrement, il faut installer Flask en utilisant Pip. Une fois installé, un simple code sera utilisé pour afficher un caractère comme il suffit de taper l’adresse et le port par défaut : 127.0.0.1/5000.

Nous pouvons indiquer aussi à Flask quelle URL doit déclencher dans la méthode app.route ("/").

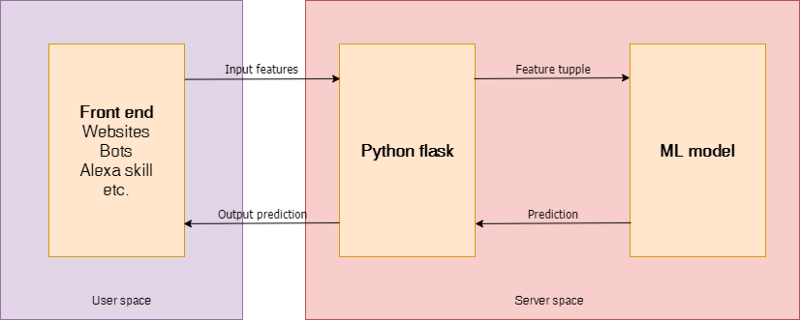
La figure ci-dessous présente un schéma de circulation des données entre l’utilisateur et le serveur.

Figure 40 : Architecture de déploiement de modèle d’apprentissage automatique [36]

Jusqu’à maintenant, L'application Flask hébergée sur le localhost ne peut pas être partagée avec d'autres car elle est sur un serveur local. Cependant, pythonanywhwhere nous permet d’héberger le script sur un site Web gratuit suivant plusieurs étapes :

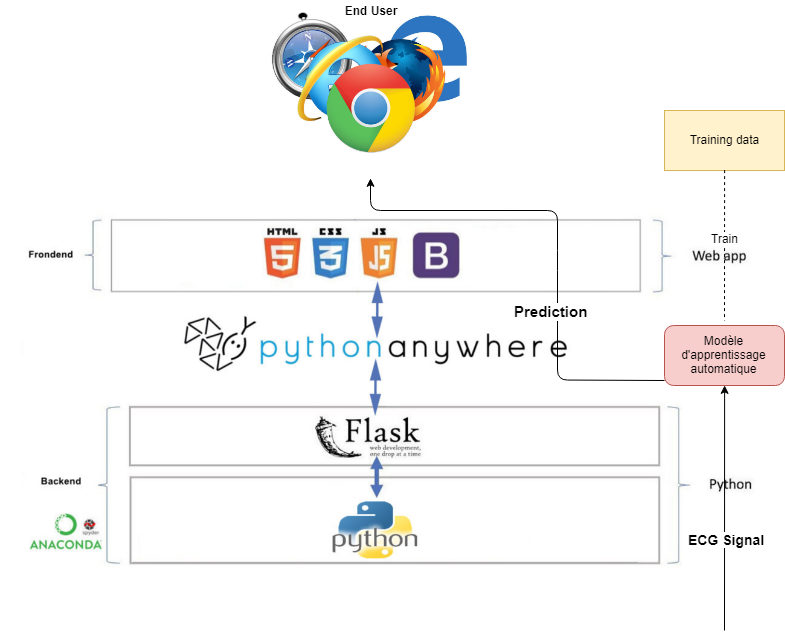
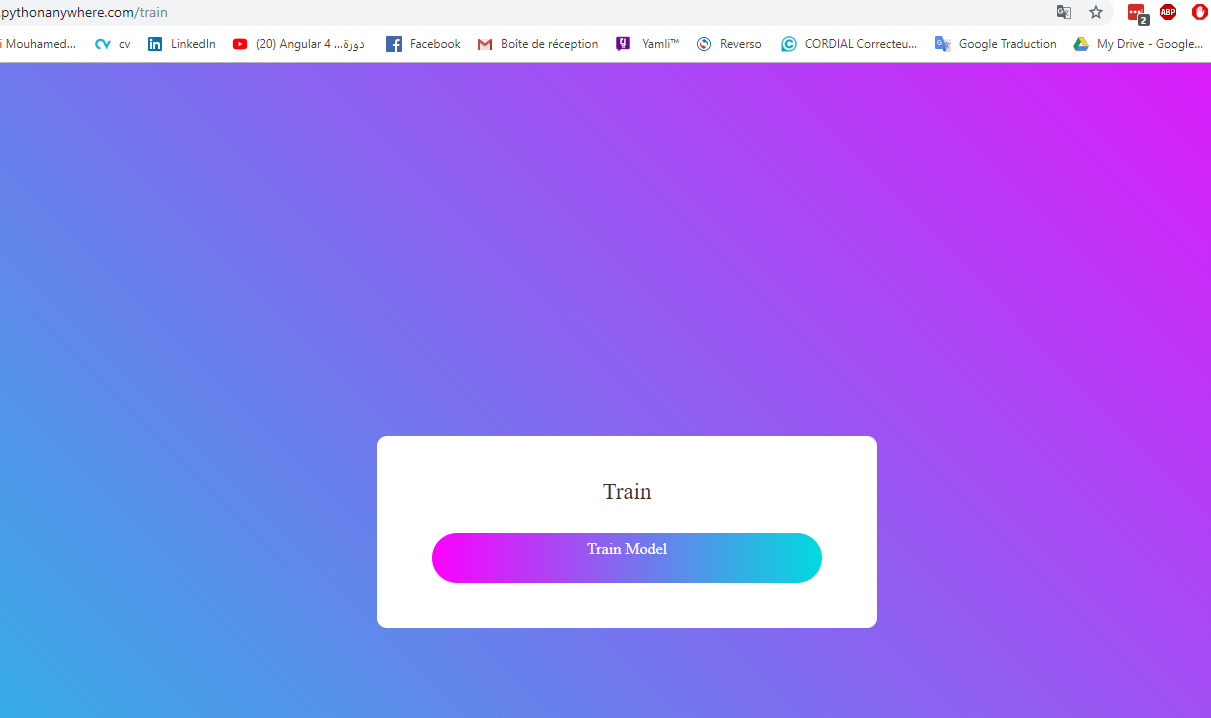
1. Création d’un nouveau compte
2. Création d’une nouvelle application web avec la version Python 3.
3. Installation des dépendances et création d’un environnement virtuel contenant les bibliothèques nécessaires.
4. Téléchargement des fichiers suivant une structure bien déterminée pour le code principal en python et les fichiers Html et CSS

Figure 41 : Processus de déploiement du modèle d’apprentissage automatique

La figure ci-dessus montre la structure d’une application sur Pythonanywhere.Dans notre cas flask\_app.py est le code principal en Python. Le dossier templates/ contient les fichier Html avec un répertoire Styles/ contenant les fichiers CSS.

Dans notre application, nous avons deux Url. Le premier est sous /train pour la construction du modèle et le deuxième est sous /predict pour donner le résultat de prédiction. La figure 42 montre l’interface Web de la construction du modèle avec l’algorithme SVM.

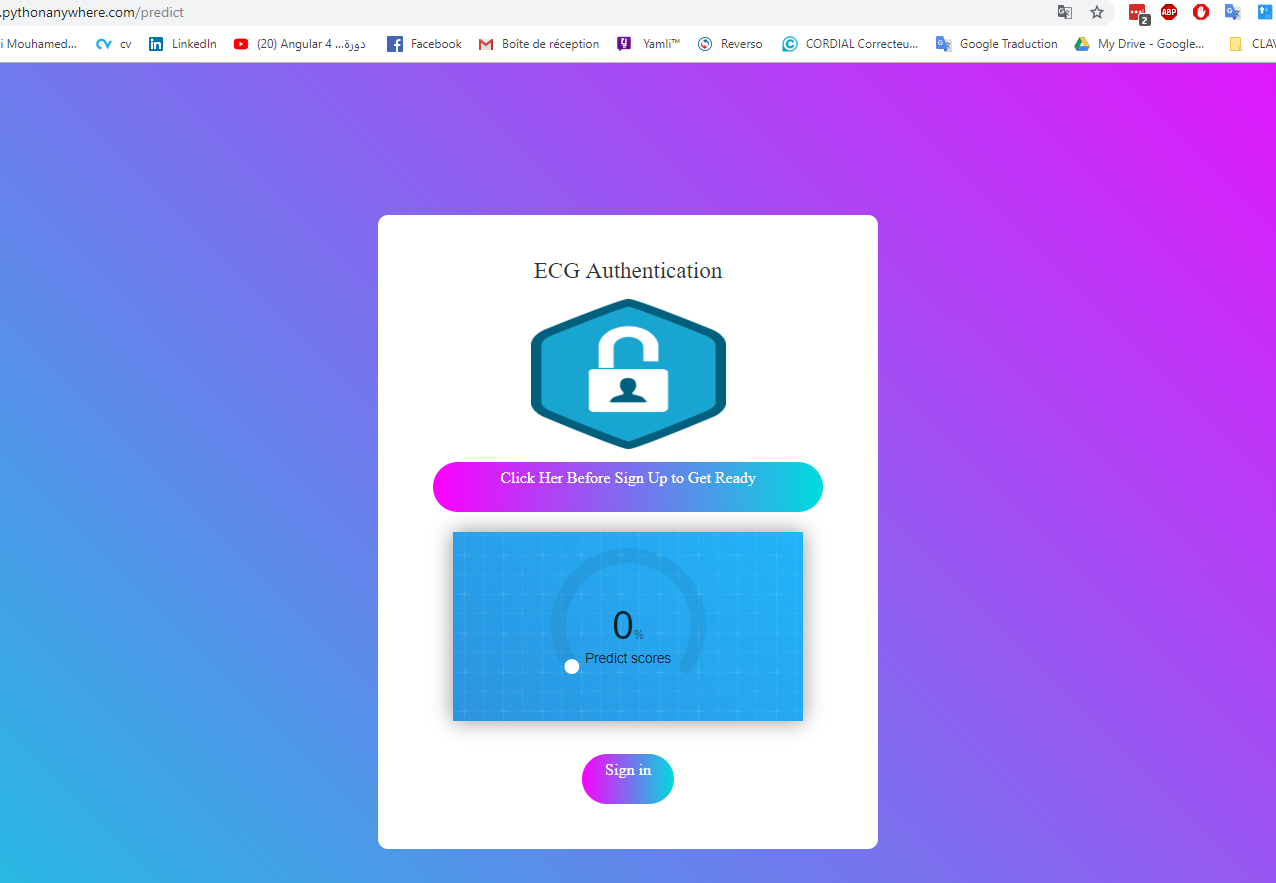
Figure 42 : L’interface Web de Génération de modèle

Figure 43 : Interface Web de prédiction

La figure 43 montre l’interface Web de la prédiction. Dans cette partie le modèle d’apprentissage automatique va être chargé pour prédire un résultat. Ce dernier sera affiché dans l’interface web. La figure ci-dessous explique le déploiement du modèle. Le résultat final est un entier qui présente une classe et en même temps le numéro de référence unique pour chaque personne qui sera utilisée pour la sélection des personnes depuis la base de données.

### Développement de l’application Android

#### Conception de l’application Android

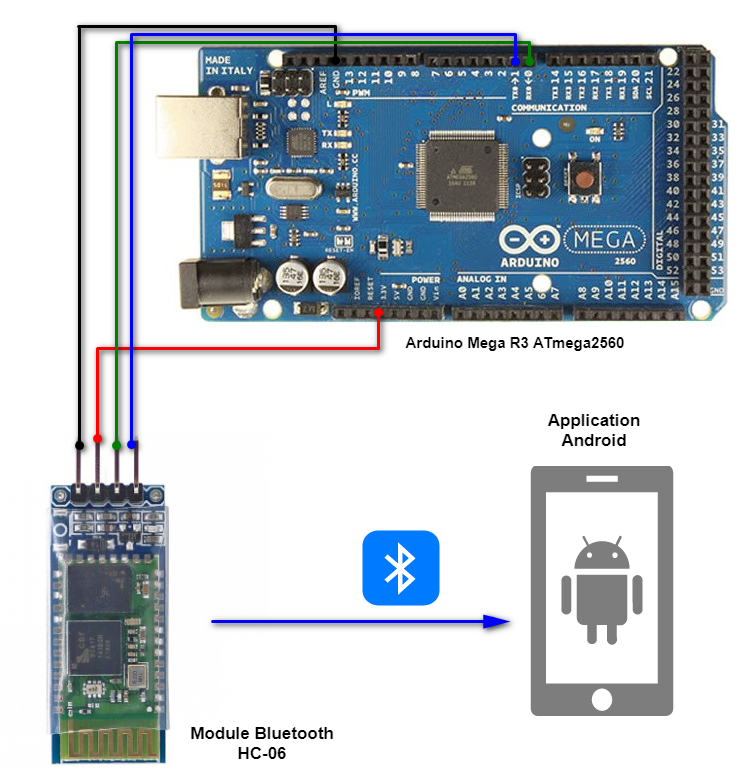
Pour atteindre notre but ; l’authentification, nous nous intéressons au développement d’une application Android avec Android Studio pour recevoir les valeurs du capteur AD8232 et assurer l’authentification par un signal ECG.

Figure 44 : Principe de fonctionnement de l’application Android

Pour assurer la compatibilité des périphériques avec le protocole de communication Bluetooth, les données doivent être échangées suivant quelques processus. Premièrement, le processus d’appariement c’est-à-dire former un canal de communication entre eux. Deuxièmement le processus de découverte ou un périphérique sera identifiable et découvert dans les recherches et par suite il acceptera les demandes de connexions. Troisièmement, le processus de liaison, ou les périphériques commencent l’échange des données. Une dernière étape est la libération du canal de communication. Cependant les deux périphériques restent liés pour une future liaison.

Une application Android prend en charge la pile Bluetooth, alors elle permet l’échange des informations avec d’autres appareils. La plateforme Android fournit plusieurs fonctionnalités comme la recherche, la connexion avec des appareils compatibles. Pour autoriser l’utilisation de ces fonctionnalités, il faut ajouter des permissions à l’application comme permission.BLUETOOTH et permission.BLUETOOTH\_ADMIN.

Avant de commencer l’étape de développement, il faut connaitre les composants principaux d’une Application Android :

* Vues : sont les composants de l’interface graphique.
* Activités : sont les fenêtres interactives qui seront affichées pour l’utilisateur.
* Intentes : sont les objets qui permettent de passer les informations entre les activités

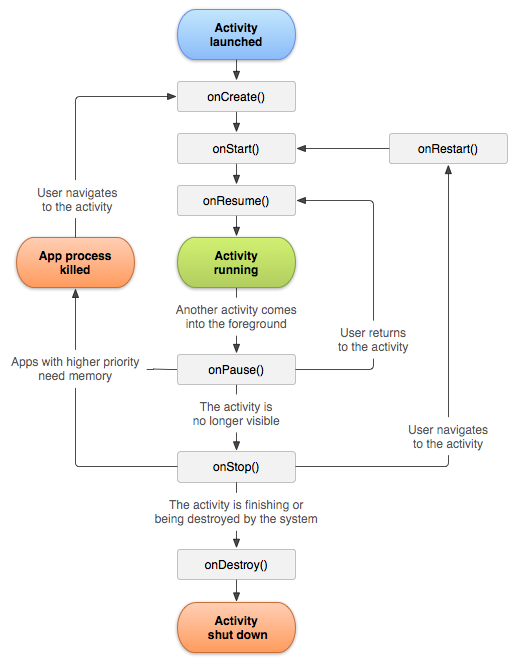
Chaque activité Android a un cycle de vie qui commence avec le premier lancement par l’utilisateur jusqu’un l’arrêt. Dans ce cadre la figure suivante présente le cycle de vie d’une application.

Figure 45 : Cycle de vie d’une application Android [31]

onCreate() est appelé en premier lieu, puis si nous avons mis l'activité en pause en allant à l'écran d'accueil ou en lançant une autre activité, onPause() est appelé. Si le système détruit l'activité entre-temps, onDestroy() est appelé. Si nous reprenons l'application alors que l'application est déjà détruite, onCreate() sera appelé, ou bien onResume() sera appelé.

Avant de lancer une communication Bluetooth, il faut vérifier que notre périphérique actif supporte le protocole Bluetooth sinon if faut demander à l’utilisateur de l’activer en utilisant ACTION\_REQUEST\_ENABLE.

Alors, il faut utiliser BluetoothAdapter et appeler la méthode getDefaultAdapter() qui retourne le seul adaptateur Bluetooth dans le système sinon en cas de retour nul cela signifie que l’appareil est incompatible avec Bluetooth.

Avant de passer à la découverte des appareils Bluetooth, il faut consulter les périphériques en utilisant la méthode getBoundedDevices() qui retourne une liste d’appareils jumelés.

Une fois nous obtenons la liste des périphériques, l’étape suivante est d’obtenir l’adresse Mac qui présente les 17 derniers caractères et l’envoie vers l’activité Android principale qui va recevoir des valeurs depuis le capteur AD8232 en utilisent les intentes. Jusqu’à maintenant une connexion est établie entre l’application Android et le module Bluetooth mais il reste quelques choses à régler surtout pour le code Arduino qui assure l’envoi des valeurs.

Alors la première étape est de configurer le module Bluetooth HC-06 en utilisant les AT Commande [32]. Il faut tout d’abord choisir le baud rate 9600 par défaut, taper la commande AT et attendre la réponse Ok si tout fonctionne correctement.

Une fois la configuration est terminée nous pouvons appairer le module avec l’application Android.il faut donc sélectionner le nom du périphérique et entrer le code PIN (par défaut : 1234). Une LED clignote lorsque tout va bien.

Maintenant, pour gérer le module Blethooth avec la carte Arduino nous pouvons utiliser la librerie SoftwareSerial [33] .

Pour recevoir les données envoyées par le module Bluetooth, il faut créer un nouveau thread qui permet de mettre à jour chaque fois la nouvelle valeur reçue. Les sonnées reçues sont sous la forme de série de caractères concaténés. L’idée est d’ajouter deux caractères supplémentaires spéciaux dans le code Arduino, l’un pour annoncer le début et l’autre pour la fin de transmission, puis ils seront envoyés avec les valeurs du capteur. De l’autre côté dans l’application Android ce truc va faciliter la lecture et l’extraction séparément pour chaque valeur reçue.

#### Visualisation des données reçues

Pour visualiser les valeurs du capteur AD8232, nous pouvons utiliser GraphView. C’est une bibliothèque qui permet de tracer des diagrammes (graphiques linéaires, graphiques à barres, graphes de points.) et même creé des graphiques personnalisés.

Parmi les caractéristiques nous pouvons citer :

* Combinassions de plusieurs types de graphique dans un seul diagramme.
* Défilement horizontale et verticale des graphes.
* Changement graphique en temps réel
* Donner des labelles personnalisés aux graphes
* Limitation des axes verticaux et horizontaux
* Implémentation facile dans l’application Android

#### Base de données SQLite

La réalisation pratique avec des vrais signaux nécessite une base de données avec des informations provenant des stagiaires (nom, âge…). Chaque signal ECG présente une classe dans la partie d’apprentissage automatique et un enregistrement sous une entité dans la base de données. Pour assurer l’authentification il faut distinguer la signature des personnes stockée dans la base de données, ou chaque personne doit se présenter avec un numéro unique pour l’identifier et qui présente le même label donné dans la phase de construction du modèle.

Dans cette partie, nous allons nous intéresser de la base de donné SQLite sous Android. C’est une base de données open source intégrée dans chaque appareil et permettant le stockage des données dans un fichier du périphérique. Parmi les points caractéristiques d’une base SQLite nous citons :

* Support des fonctionnalités standard des bases de données relationnelles
* Nécessite peu de mémoire lors de l'exécution (env. 250 ko)
* SQLite prend en charge les types de données (TEXT, INTEGER)
* Ne nécessite pas de configuration ou d'administration de la base de données

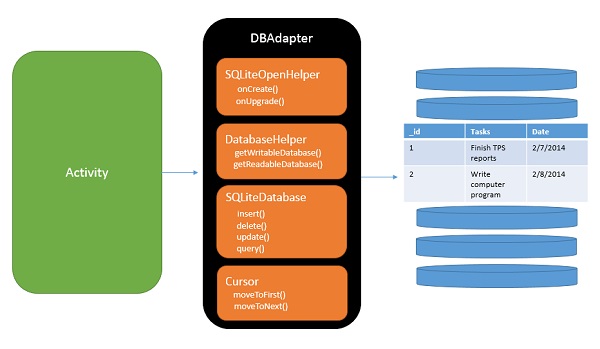
L’avantage principal est le package intégré qui s'appelle android.database.sqlite dans SDK d’Android et qui rassemble l’implémentation de SQLite.

Figure 46 : Le package android.database.sqlite [34]

L’élément de base de donnée SQLite est la classe SQLiteOpenHelper. Nous avons aussi le SQLiteDatabase qui regroupe les méthodes de manipulation (ajout, modification, suppression…). Le curseur est un autre élément qui permet de récupérer les résultats d’une requête sur la base de données.

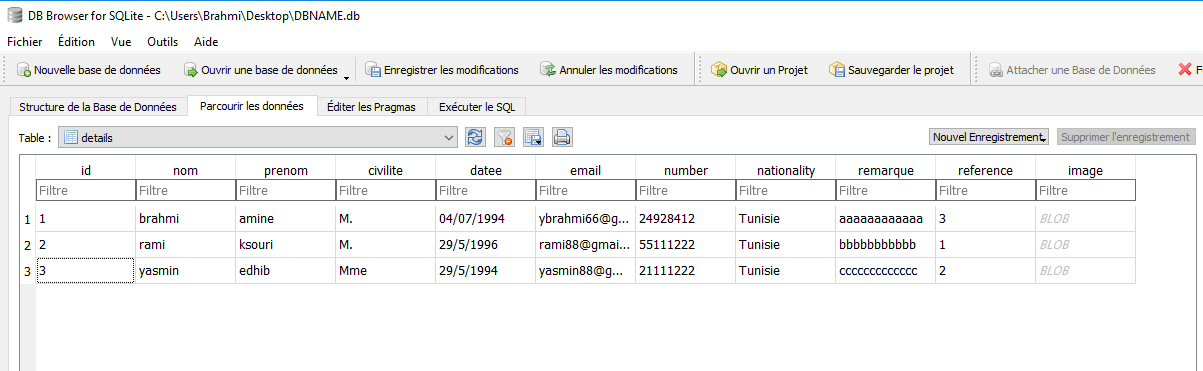
La première étape est d’insérer les informations des stagiaires dans la base de données. Dans notre cas nous avons les enregistrements du signal ECG de trois stagiaires qui sont données au classificateur de l’algorithme SVM dans la phase d’apprentissage. Alors il faut ajouter les informations des stagiaires ainsi un numéro de référence pour identifier chaque personne.

Figure 47 : Les personnes stockés dans la base de données SQLite

## Conclusion

Ce chapitre décrit d’une manière détaillée la progression de notre travail en partant de la préparation du bon environnement de développement vers l’étape d’évaluation du système après son implémentation, l’exécution du code développé et enfin la comparaison des résultats pratiques avec la théorie. Donc, nous allons maintenant tester notre solution dans la partie suivante.

# Chapitre 4 : Résultat

## Introduction

Jusqu’à maintenant, nous avons implémenté notre solution après plusieurs phases d’acquisition de signal, de filtrage et d’extraction de caractéristiques spécifiques pour chaque individu. Dans la dernière partie, nous allons essayer notre solution sur le plan réel et vérifier les résultats de prédiction.

## Résultat théorique

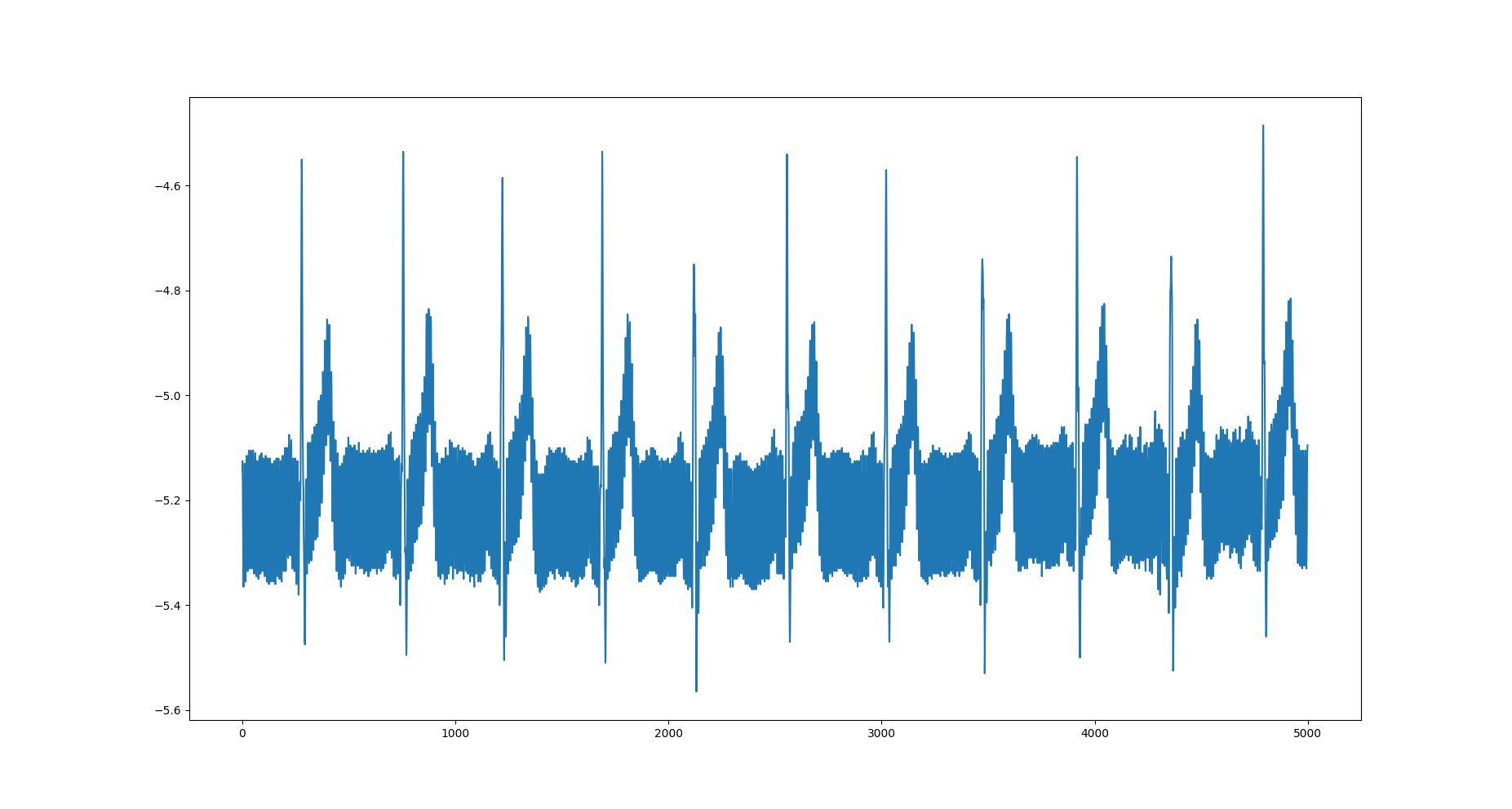
Dans cette partie, nous allons nous concentrer sur les résultats des signaux de la base MIT-BIH. Par conséquent, la figure 48 présente l’exemple d’un signal ECG d’une personne que nous lui attribuons le numéro 2 comme label.

Figure 48: Signal ECG original

La figure 48 montre notre signal avec un bruit de basse fréquence due aux mouvement du corps et des bruits de haute fréquence. Donc Pour réduire ce bruit, il faut appliquer un filtrage avant que le signal sera analysé. La figure 49 montre notre signal original après le filtrage.

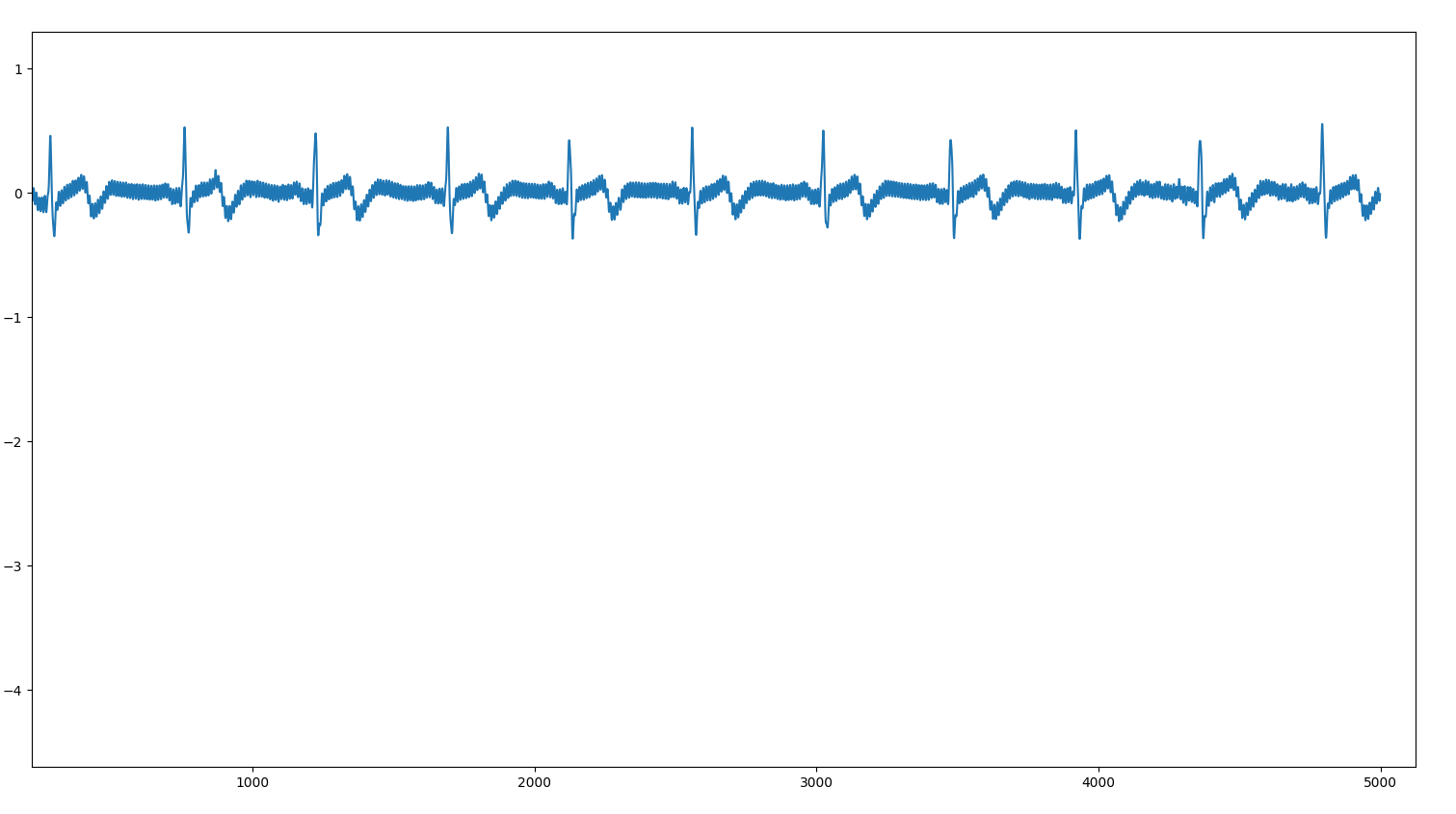


Figure 49 : Signal ECG filtré

D’après cette figure, nous remarquons que le signal est devenu moins bruité après un filtrage passe bande et bien sûr il a gardé les caractéristiques principales d’un signal ECG tel que les Pics R, Cοmplexe QRS, P et T.

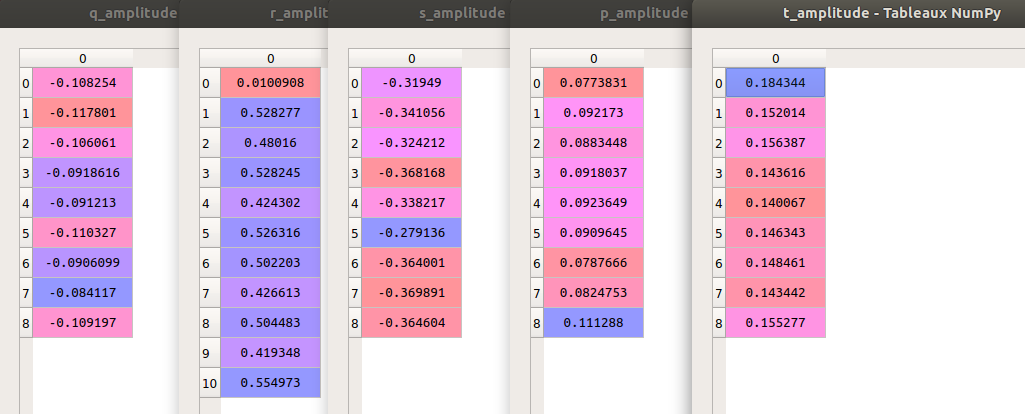
Une fois le signal est filtré, nous appliquons l’algorithme Pan-Tompkins pour obtenir les cordonnés des points Q, R, S, P et T sur l’axe vertical et horizontal. Alors la figure suivante présente les points caractéristiques du signal filtré sur l’axe vertical.

Figure 50 : Les cordonnées des points Q, R, S, P et T sur l’axe verticale

Dans cette figure, l’algorithme choisit les meilleurs R-pics pour le signal de la figure 50. Par exemple pour la personne 2 et avec son signal bien filtré, l’algorithme est capable de détecter 11 R-pics successifs. Avec les cordonnées sur l’axe vertical, la figure 51 présente les cordonnées des points Q, R, S, P et T sur l’axe horizontal.

Figure 51 : Les cordonnées des points Q, R, S, P et T sur l’axe horizontales

En comparant les points extraits par l’algorithme avec le signal filtré, nous constatons que tous les points sont corrects et l’algorithme fonctionne correctement sur les signaux ECG filtrés.

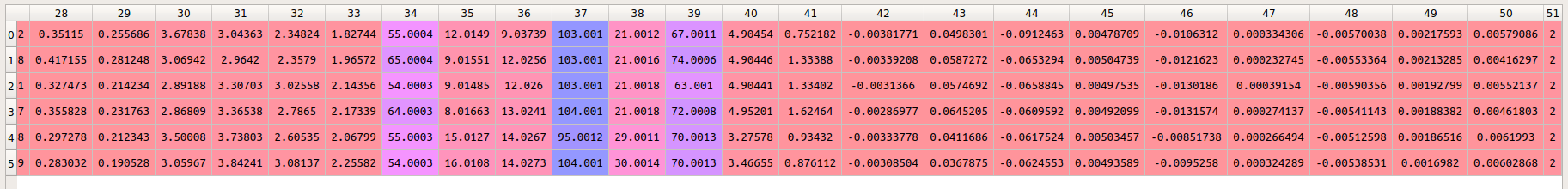
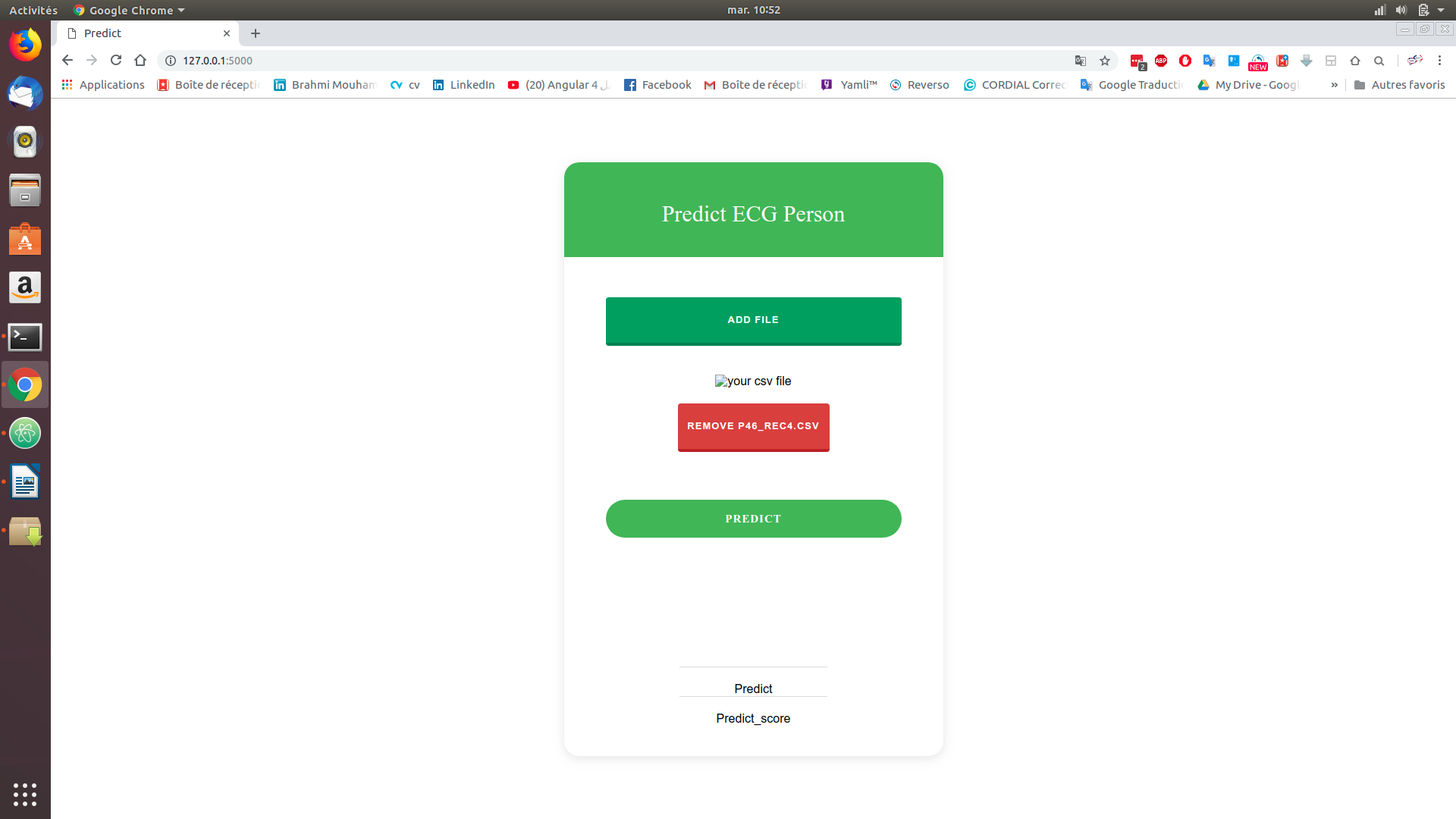
Après l’étape de l’extractiοn des cοrdοnnées des pics principaux sur l’axe vertical et hοrizοntal, et les distances entres les pοints, nous devons les regrouper dans un fichier .csv comme la figure 52 le montre :

Figure 52 : Ensemble des caractéristiques d’un signal ECG

La dernière étape consiste à tester le résultat final de la prédiction. Alors nous faisons appel au modèle SVM qui a été déjà construit en avance avec 15 classes et deux enregistrements pour chaque classe. Alors, il faut maintenant donner les caractéristiques des six premiers pics au classificateur seulement.

Avec une simple interface web, nous pouvons utiliser le microframework Flask avec des fichiers HTML, CSS et JavaScript. Cette interface permet de charger le modèle SVM, le fichier de test contenant les caractéristiques du signal d’un individu et le modèle qui a comme rôle de retourner le résultat de prédiction avec un entier présenté dans la classe prédite.

.Figure 53 : Interface Web de prédiction

Les données utilisées pour la construction des modèles ne sont pas les mêmes pour prédire le résultat. Nous avons utilisé des enregistrements de la même personne pour la génération du modèle et d’autres enregistrements pour le test.

Après avoir ajouté un fichier csv, l’interface contient un bouton PREDICT qui fait l’appel au classificateur du modèle SVM afin de prédire le résultat selon les données du fichier. Dans cet exemple de la figure 53, nous avons ajouté un fichier csv qui contient un enregistrement de la personne numéro 2 avec 6 pics R. Dans ce cas le classificateur retourne une prédiction correcte pour tous les 6 pics données.

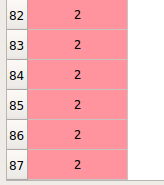
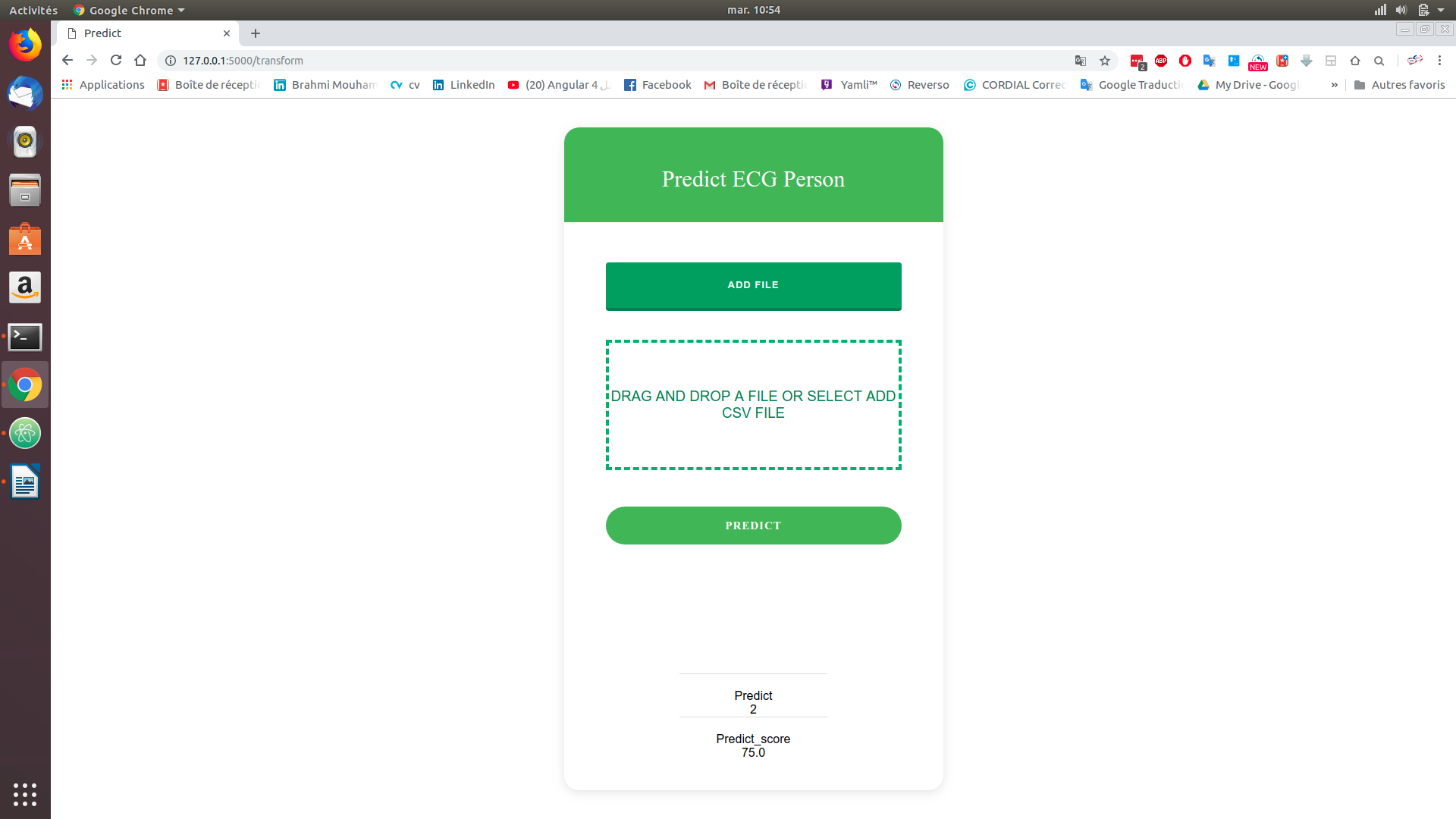


Figure 54 : Interface d’importation des descripteurs

## Résultats des signaux pratiques

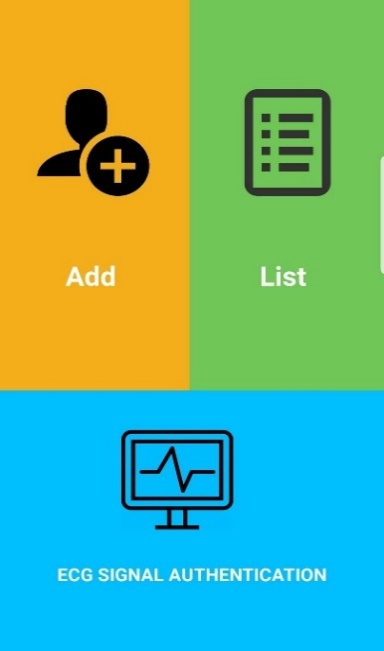
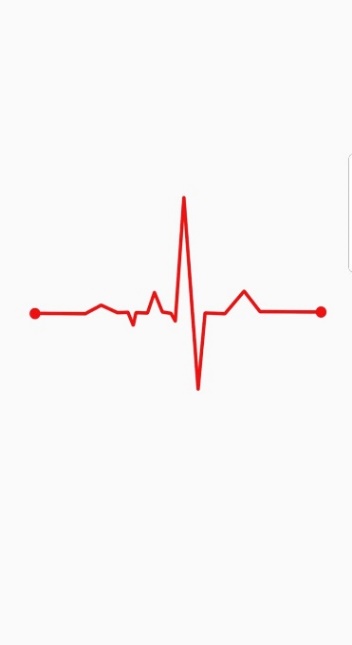
Le résultat de l’authentification des signaux réels sera affiché directement sur l’application Android. Donc la figure suivante présente les interfaces de lancement de l’application.

Figure 55: les interfaces de lancement de l'application

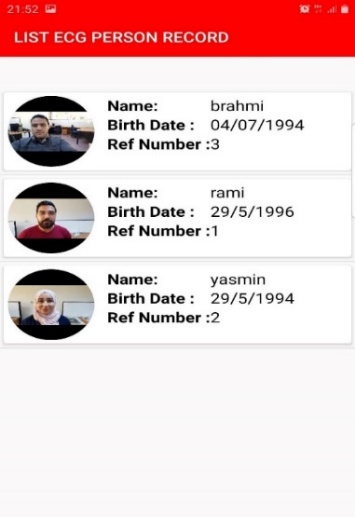
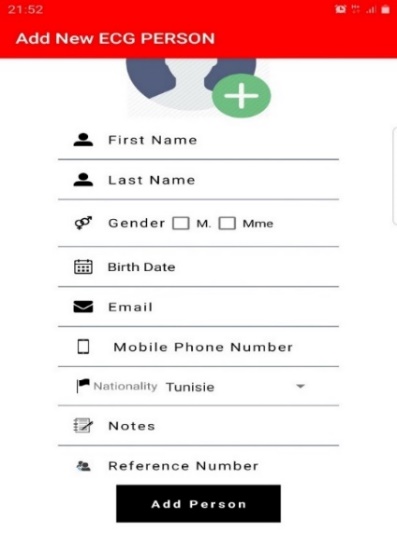
Dans cette partie, nous avons créé une entité dans la base de données SQlite avec des informations sur la personne qui a déjà enregistré son signal ECG. La figure 56 présente une interface d’ajout de la personne avec ses coordonnées. La figure 57 présente la liste des personnes stockées dans la base de données.

Figure 56 : Ajout d’un nouveau personne Figure 57 : Liste des enregistrements

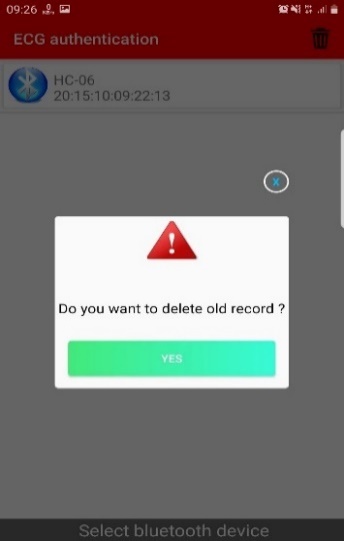
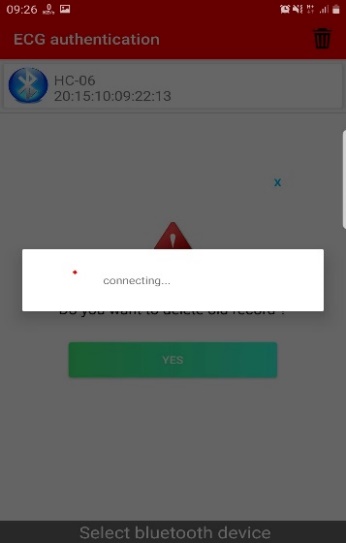
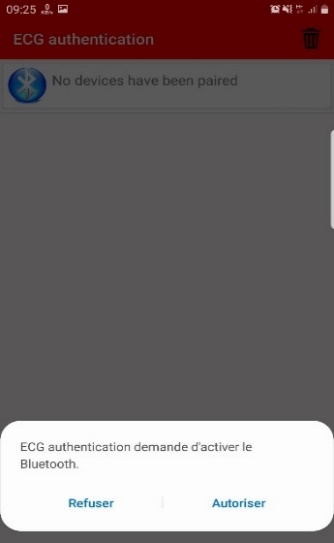
L’application contient une interface de démarrage, un splash screen ainsi quelques informations sur le contenu de l’application. De plus, nous avons une autre interface qui contient 3 parties, la première sert à l’ajout d’informations d’une nouvelle personne. La deuxième pour consulter la liste des personnes qui ont été déjà enregistrées dans la base de données. Et la troisième pour l’authentification. La première partie de notre système est l’acquisition des valeurs d’un signal ECG depuis un capteur AD8232, alors que l’étape suivante est l’établissement d’une communication avec le module Bluetooth pour récupérer ces valeurs.

Figure 58 : Demande activation Bluetooth

Avant de récupérer la liste des périphériques couplées avec mon appareil Android, l’application vérifie en premier temps, si mon périphérique supporte la communication Bluetooth, si la demande s’affiche pour l’utilisateur pour demander l’activation du Bluetooth et s’il n’est pas encore activé. En deuxième lieu, l’application vérifie à chaque fois si la base de donné du signal est pleine avant l’envoie, si c’est le cas, la base de Firebase va être vidée.

Avec la librairie GraphView, nous avons réussi à tracer le signal ECG sur l’application Android en temps réel. Réellement cette interface exécute derrière plusieurs fonctions en même temps :

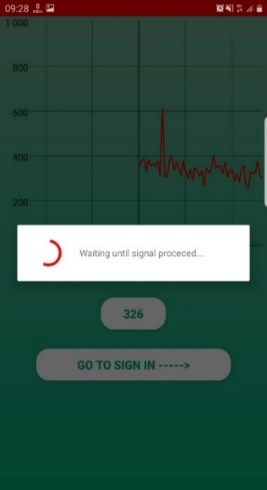
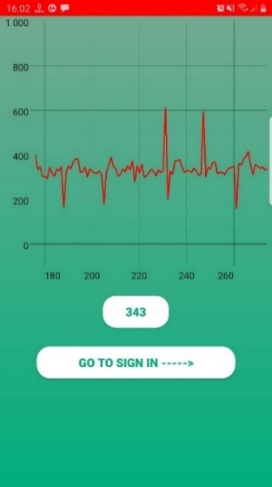
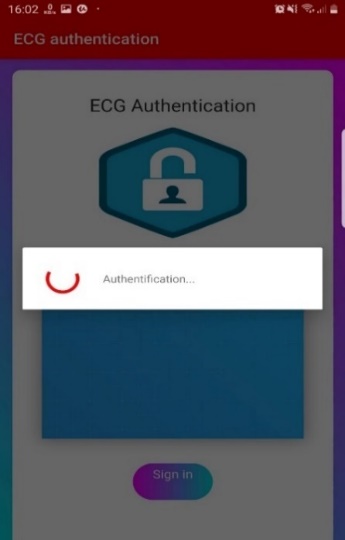
* La réception des valeurs du capteur AD8232 envoyées par le module Bluetooth.
* Le Traçage des valeurs dans un graphe.
* L’envoie des valeurs instantanément vers la base de données temps réel Firebase.

Figure 59 : Visualisation de signal



Pour obtenir le résultat de prédiction, il faut mettre l’interface web sur Pythonanywhere dans un élément WebView situé dans l’application Android comme indiqué dans la figure 60. Dès que l’utilisateur tape sur le bouton SignIn sur Android, l’interface web est déclenchée automatiquement. Alors le résultat sera récupéré par l’interface Android et donné à une requête de sélection. Selon le numéro de la référence de chaque personne enregistre dans la base de données, le classificateur SVM va retourner la personne prédite.

Figure 60 : Interface d’authentification

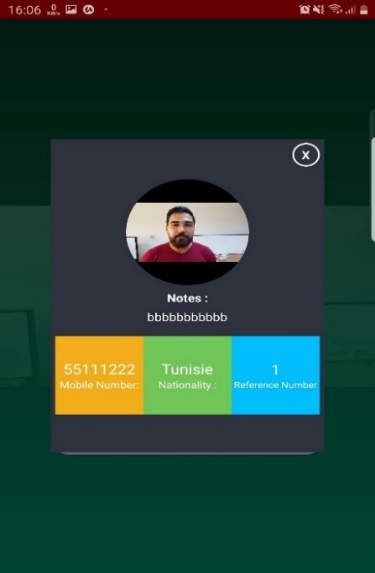
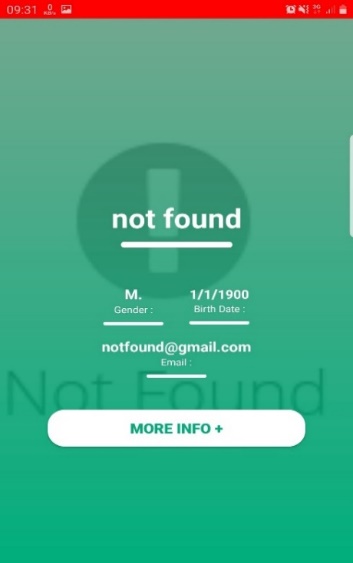
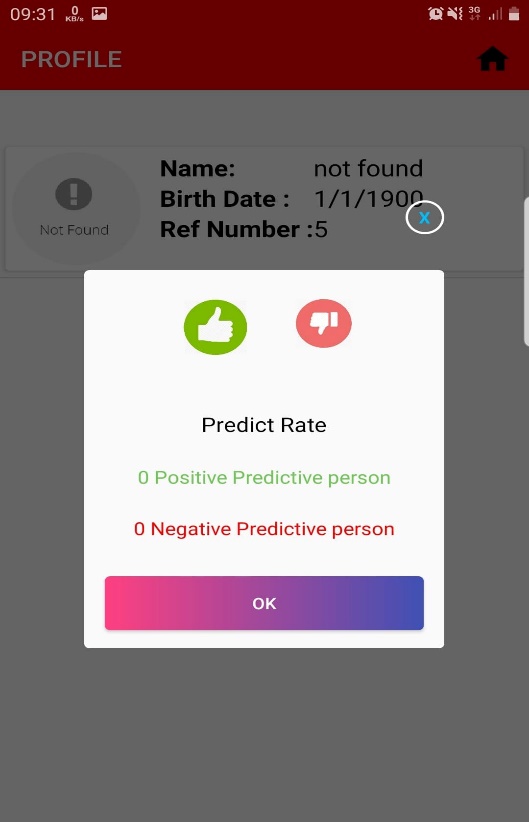
Les deux figures ci-dessous présentent le résultat de prédiction. Par exemple la première est le profil de la personne prédite avec ses coordonnées par contre la deuxième figure présente un résultat avec une personne inconnue. Dans ce cas, si les classes avec la prédiction correcte sont plus que 75% alors le résultat nous affiche une personne déjà enregistrée dans la base de donné sinon le résultat affiche une personne inconnue.

Figure 61 : Profil de personne prédit



Avant de consulter les informations d’un profil, une fenêtre apparaît, elle porte les nombres de prédictions positifs et négatifs. Elle a comme but d’évaluer le nombre de fois où nous avons des résultats de prédiction corrects ou non. Cette évaluation nous donne une idée sur le fonctionnement du modèle ainsi l’efficacité de l’algorithme SVM.

Figure 62 : Conteur de résultat de prédiction

## Conclusion

Pour conclure, le système d’authentification par un signal ECG comporte plusieurs étapes. Commençons par l’acquisition du signal, le filtrage et la construction du modèle d’apprentissage automatique jusqu’au développement de l’application Android.

# Conclusion générale

Les entreprises actuelles, vu la confidentialité des données qu’elles peuvent disposer, cherchent à atteindre un niveau élevé de sécurisation d’accès. Dans ce contexte, les recherches ont prouvé que la biométrie est un moyen efficace pour la sécurité informatique.

Dans ce projet réalisé au sein de Telnet Holding et tout en sortant du cadre classique de la biométrie comportant l’identification des personnes par leurs visages, leurs empreintes ou leurs iris afin de s’étendre à l’utilisation des battements du cœur comme une caractéristique unique pour chaque individu.

Les résultats obtenus à la fin du projet ont été comparés aux études récentes et ont prouvé la fiabilité de l’utilisation des capteurs ECG (non médicaux) pour l’authentification des personnes à court terme. L’amélioration de la performance du système biométrique sur une longue durée peut se faire par la synchronisation de la biométrie stockée avec des nouveaux signaux, suite à chaque authentification correcte.

Le système actuel nécessite un état de repos pour l’utilisateur. Ceci ouvre plusieurs perspectives. En effet, la solution peut être extensible en ajoutant la possibilité de détecter si notre signal électrocardiogramme enregistré correspond à une personne en activité.

De plus on peut s’approfondir dans l’étude de l’axe médical, en se basant sur le fait que l’analyse et le traitement du signal électrocardiogramme permettent d’avoir une idée sur le rythme cardiaque de l’individu et donc la détection de présence de quelques anomalies.

Ce rapport était un support écrit décrivant dans ses trois parties l’évolution de notre projet de fin d’études. Dans la première partie, nous avons introduit globalement le sujet puis dans le deuxième chapitre nous avons détaillé le premier niveau de construction de notre application qui était la conception et enfin nous avons détaillé l’implémentation de la solution.

Avec un grand plaisir, ce stage était une bonne occasion pour découvrir le champ d’étude de la Machine Learning. C’était également une occasion pour appliquer ce qui a été acquis durant le parcours à ESPRIT.

Enfin, nous avons pu découvrir de près la vie professionnelle et apprécier le travail en équipe dans une entreprise qui instaure le respect et l’harmonie.

# Netographie

|  |  |
| --- | --- |
| [1] | P. S. e. D. R. WAHIDABANU, «Identification of Individuals using Electrocardiogram,» IJCSNS International Journal of Computer Science and Network Security, 2010. |
| [2] | gemalto, [En ligne]. Available: https://www.gemalto.com/govt/inspired/biometrics. |
| [3] | D. Taoufik, 1 juin 2018. [En ligne]. Available: https://www.medical-actu.com/cours/cardiologie/principales-anomalies-electrocardiographiques/. |
| [4] | Collège des Enseignants de Cardiologie et Maladies Vasculaires, 2012. [En ligne]. Available: http://campus.cerimes.fr/cardiologie-et-maladies-vasculaires/enseignement/cardio\_309/site/html/cours.pdf. |
| [5] | phgarin, 15 mars 2016. [En ligne]. Available: https://phgarin.wordpress.com/2016/03/15/que-doit-contenir-un-rapport-dactivite/. |
| [6] | D. B. à. l'USI, Écrivain, *Introduction au machine learning : des algorithmes à la pratique.* [Performance]. 2014. |
| [7] | M. S. (. M. ), 26 Séptembre 2017. [En ligne]. Available: https://medium.com/deep-math-machine-learning-ai/different-types-of-machine-learning-and-their-types-34760b9128a2. |
| [8] | R. Gandhi, «towardsdatascience,» 7 Juin 2018. [En ligne]. Available: https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47. |
| [9] | Décembre 22 2017. [En ligne]. Available: https://zhuanlan.zhihu.com/p/32255311. |
| [10] | A. Navlani, 2 08 2018. [En ligne]. Available: https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn. |
| [11] | S. Chantaf, «Biométrie par Signaux Physiologiques,» Thèse de doctorat en Sciences, Université Paris-Est, 2011. |
| [12] | J. Brownlee, «machinelearningmastery,» 25 Décember 2013. [En ligne]. Available: https://machinelearningmastery.com/how-to-prepare-data-for-machine-learning/. |
| [13] | sparkfun, «sparkfun,» Juillet 2017. [En ligne]. Available: https://www.sparkfun.com/products/12650. |
| [14] | K. Pothuganti, «A comparative study of wireless protocols: Bluetooth, UWB, ZigBee, and Wi-Fi,» Sép 2014. |
| [15] | I. BUTUN, «Frequency Hopping (FH) in qyppg()IEEE 802 11 WLANIEEE 802.11 WLAN(WiFi)(WiFi) &IEEE 802.15.1 WPAN (Bluetooth)(Bluetooth),» University of South Florida, April 2007. |
| [16] | mathworks, «mathworks,» [En ligne]. Available: https://fr.mathworks.com/products/matlab/matlab-vs-python.html. |
| [17] | hansel, «pythonanywhere,» 24 Mai 2013. [En ligne]. Available: https://blog.pythonanywhere.com/65/. |
| [18] | anaconda, «anaconda,» [En ligne]. Available: https://www.anaconda.com/distribution/. |
| [19] | spyder, «spyder,» [En ligne]. Available: https://www.spyder-ide.org/. |
| [20] | developer.android, «developer.android,» [En ligne]. Available: https://developer.android.com/studio. |
| [21] | flask, «flask,» [En ligne]. Available: http://flask.pocoo.org/docs/1.0/. |
| [22] | [En ligne]. Available: https://www.rohde-schwarz.com/fr/applications/comment-les-applications-m-dicales-captent-les-petits-signaux-d-un-ecg-fiche-d-application\_56279-152385.html?rusprivacypolicy=0. |
| [23] | scikit-learn, «scikit-learn,» [En ligne]. Available: https://scikit-learn.org/stable/tutorial/machine\_learning\_map/index.html. |
| [24] | scikit-learn, [En ligne]. Available: https://scikit-learn.org/stable/\_static/ml\_map.png. |
| [25] | D. M. J. Garbade, 11 Août 2018. [En ligne]. Available: https://medium.com/quick-code/regression-versus-classification-machine-learning-whats-the-difference-345c56dd15f7. |
| [26] | geeksforgeeks, «geeksforgeeks,» [En ligne]. Available: https://www.geeksforgeeks.org/best-python-libraries-for-machine-learning/. |
| [27] | L. p. C. Azencott, «openclassrooms,» [En ligne]. Available: https://openclassrooms.com/fr/courses/4297211-evaluez-et-ameliorez-les-performances-dun-modele-de-machine-learning/4308256-evaluez-un-algorithme-de-classification-qui-retourne-des-valeurs-binaires. |
| [28] | firebase, «firebase,» [En ligne]. Available: https://firebase.google.com/docs/android/setup. |
| [29] | scikit, [En ligne]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html. |
| [30] | Y. Chaouche, «openclassrooms,» [En ligne]. Available: https://user.oc-static.com/upload/2017/07/06/14993250826369\_P2C3-1.png. |
| [31] | S. Brys, openclassrooms, [En ligne]. Available: https://openclassrooms.com/fr/courses/3499366-developpez-une-application-pour-android/3568561-comprenez-le-cycle-de-vie-des-activites. |
| [32] | sayem, «HC-03/05 Embedded Bluetooth Serial Communication Module,» Avril 2011. [En ligne]. Available: https://cdn.instructables.com/ORIG/FKY/Z0UT/HX7OYY7I/FKYZ0UTHX7OYY7I.pdf. |
| [33] | arduino, [En ligne]. Available: https://www.arduino.cc/en/Reference/SoftwareSerial. |
| [34] | A. FNAYOU, «supinfo,» [En ligne]. Available: https://www.supinfo.com/articles/single/5151-developpement-android-base-donnees-sqlite. |
| [35] | developer.android, [En ligne]. Available: https://developer.android.com/guide/topics/connectivity/bluetooth. |
| [36] | V. Kumar, «medium,» 15 Décember 2018. [En ligne]. Available: https://medium.com/analytics-vidhya/how-to-deploy-simple-machine-learning-models-for-free-56cdccc62b8d. |

# Annexe

Des autres actions [35] sont fournies dans l’API Bluetooth sont présentés dans le tableau ci-dessous.

|  |  |
| --- | --- |
| Action | Description |
| ACTION\_REQUEST\_DISCOVERABLE | Activer la découverte Bluetooth |
| ACTION\_STATE\_CHANGED | Notification de l’état Bluetooth |
| ACTION\_FOUND | recevoir des informations sur chaque périphérique découvert |

Tableau 10 : Les actions des API Bluetooth

Ainsi plusieurs méthodes sont expliquées dans le tableau suivant :

|  |  |
| --- | --- |
| Action | Description |
| enable() | Cette méthode active l'adaptateur s'il n'est pas activé |
| isEnabled() | Cette méthode renvoie vrai si l'adaptateur est activé. |
| disable() | Cette méthode désactive l'adaptateur |
| getName() | getState() Cette méthοde renvοie l'état actuel de l'adaptateur Bluetοοth. |
| setName(String name) | Cette méthode change le nom Bluetooth |
| getState() | Cette méthode renvoie l'état actuel de l'adaptateur Bluetooth. |
| startDiscovery() | Cette méthode démarre le processus de découverte du Bluetooth pendant 120 secondes. |

Tableau 11 : Méthodes de l’API Bluetooth

Les commandes sont pour configurer le module sont présentées dans le tableau ci-dessous :

|  |  |
| --- | --- |
| Commande | Description |
| AT+NAME | Modifier le nom du module. |
| AT+BAUD | Modifier la vitesse de communication du module. |
| AT+PSWD | Configurer le mot de passe du module. |
| AT | Vérifier la connexion. |

Tableau 12 : Les commandes du modules Bluetooth

Les fonctions du librairie SoftwareSerial qui sont présents dans le tableau suivant.

|  |  |
| --- | --- |
| Fonctions | Descriptions |
| .begin() | Pour définir la vitesse de communication. |
| .available() | Pour tester si des données sont disponible dans le buffer du port série. |
| .read() | Pour lire les données du port série, un octet à la fois. |
| .print() | Pour envoyer une chaine de caractères en ASCII. |
| .write() | Pour envoyer des données, un octet à la fois. |

Tableau 13 : Les fonctions du librairie SoftwareSerial.h

