جامعة شعيب الدكالي
ⵜⴰⵙⴷⴰⵡⵉⵜ ⵏ ⵛⵄⵉⴱ ⴷⴽⴰⵍⵉ
Université Chouaïb Doukkali

PROJET DE FIN D'ÉTUDES POUR L'OBTENTION DU DIPLÔME MASTER EN BUSINESS INTELLIGENCE ET BIGDATA ANALYTIC.

BIBDA
BUSINESS INTELLIGENCE AND BIG DATA ANALYTICS

CONCEPTION ET RÉALISATION D'UN MODÈLE DE RECONNAISSANCE VOCALE (SPEECH RECOGNITION) DU CONTENU AUDIO.

Réalisé par :

ZENNOU MOHAMMAD

Soutenu le 03/11/2020 devant le jury composé de :

Pr. MADANI Abdellah	Faculté des Sciences de El Jadida	(Président)
Pr. AZOUAOUI Ahmed	Faculté des Sciences de El Jadida	(Examinateur)
Pr. ZINE-DINE Khalid	Faculté des Sciences de Rabat	(Encadrant - Université)
M. ATIF Zakaria	Ing. BigData à HENCEFORTH	(Encadrant - Entreprise)

HENCEFORTH

2019-2020

RESUME

Ce rapport est un résumé de mon dernier stage chez HENCEFORTH, effectué dans le cadre de mon projet de fin d'études pour obtenir mon Master en Business Intelligence et Big Data Analytics.

Ce rapport donne un aperçu de la technologie de reconnaissance automatique du locuteur pour l'authentification biométrique. Une personne peut être identifiée par diverses caractéristiques telles que la signature, les empreintes digitales, la voix, les traits du visage, etc. Ce type de méthodes d'authentification est connu sous le nom d'authentification biométrique des personnes.

La reconnaissance du locuteur fait référence au processus de reconnaissance automatique de la personne qui parle sur la base d'informations individuelles incluses dans les ondes vocales. Pour une reconnaissance vocale fiable et de haute précision, des méthodes de représentation simples et efficaces sont nécessaires.

Dans cette étude, nous présentons plusieurs méthodes et techniques de Prétraitement audio comme première étape pour préparer les caractéristiques. Ainsi, des coefficients sont extraits du signal vocal entrant en utilisant le MFCC et le LPC et ils représentent les caractéristiques vectorielles de chaque locuteur. En outre, une approche d'apprentissage automatique indépendante du texte est utilisée dans ce projet, ce qui rend le modèle flexible à plusieurs langues, telles que la machine à vecteurs de support (SVM) et le réseau neuronal artificiel (ANN).

Mots clés : *Reconnaissance Vocale, Reconnaissance de locuteur, Détection d'Activité Vocale (VAD), Mel Frequency Cepstrum Coefficients (MFCC), Linear Predict Coefficient (LPC), Apprentissage Automatique. Apprentissage Profond.*

ABSTRACT

This report is a summary of my final internship at HENCEFORTH, completed as part of my graduation project to obtain my Master's Degrees in Business Intelligence and Big Data Analytics, This report gives an overview of automatic speaker recognition technology for biometric authentication. A person can be identified by various characteristics such as signature, fingerprints, voice, facial features, etc. This type of authentication methods is known as biometric person authentication.

Speaker recognition refers to the process of automatically recognizing who is speaking on the basis of individual information included in speech waves. For a reliable and high accuracy of speech recognition, simple and efficient representation methods are required.

In this study, we present several methods and techniques of audio preprocessing as the first step to prepare the features,Thus coefficients are extracted from incoming speech signal using MFCC and LPC and it represent features vector of the each speaker. Additionally, a Machine Learning text-independent approach is used in this project which makes the model flexible to multiple languages, such as Support Vector Machine (SVM), and Artificial Neural Network (A NN).

Keywords: *Speech recognition, Speaker recognition, Voice Activity Detection (VAD), Mel Frequency Cepstrum Coefficients (MFCC), Linear Predict Coefficient (LPC), Machine Learning, Deep Learning.*

نبذة مختصرة

هذا التقرير هو ملخص لتدريب عملي النهائي الخاص بي، والذي اكتمل كجزء من مشروع التخرج للحصول على درجات الماجستير في ذكاء الأعمال وتحليلات البيانات الضخمة، ويقدم هذا التقرير نظرة عامة على تقنية التعرف التلقائي على المتحدثين للمصادقة البيومترية. يمكن التعرف على الشخص من خلال خصائص مختلفة مثل التوقيع وبصمات الأصابع والصوت وميزات الوجه وما إلى ذلك. يُعرف هذا النوع من أساليب المصادقة باسم المصادقة البيومترية للشخص.

يشير التعرف على المتحدث إلى عملية التعرف التلقائي على المتحدث على أساس المعلومات الفردية المضمنة في موجات الكلام. للحصول على دقة عالية وموثوقة للتعرف على الكلام، يلزم وجود طرق تمثيل بسيطة وفعالة.

في هذه الدراسة، نقدم عدة طرق وتقنيات للمعالجة الصوتية المسبقة كخطوة أولى لإعداد الميزات، وبالتالي يتم استخراج المعاملات من إشارة الكلام الواردة باستخدام وهي تمثل ناقل السمات لكل المتحدثين.
بالإضافة إلى ذلك، يتم استخدام نهج التعلم الآلي المستقل عن النص في هذا المشروع مما يجعل النموذج مرنًا للغات متعددة، مثل الشبكة العصبية الاصطناعية.

REMERCIEMENTS

Je tiens à remercier mes superviseurs, le Dr ZINE-DINE Khalid, et M. ATIF Zakaria, pour leurs soutiens et leurs conseils constants durant la période de stage. Leur grande expérience m'a beaucoup aidé au cours de mes recherches et je suis vraiment reconnaissant d'avoir eu l'occasion de travailler avec eux.

Je voudrais également remercier les membres de mon comité, le staff de HENCEFORTH pour leur temps précieux, leur soutien, leurs conseils et leurs réactions.

J'exprime également ma gratitude au département d'informatique et à tous les membres de la faculté des sciences d'El-Jadida avec qui j'ai interagi au cours de mon programme universitaire. Merci d'avoir enrichi mes connaissances et pour en faire une expérience mémorable.

Enfin, je tiens à remercier mes parents, ma famille et mes amis pour leur soutien constant et leur confiance en moi.

Table des matières

Table des figures

Liste des Tables

LISTE DES ACRONYMES

AI	Artificial Intelligence		
ASR	Automatic Speech Recognition		
LPC	Linear Predict Coefficient		
MFCC	Mel Frequency Cepstrum Coefficients	**FP**	False Positive
ML	Machine Learning		
DL	Deep Learning		
ANN	Artificial Neural Network		
PLP	Perceptual Linear Prediction		
SNR	Signal Noise Ratio	**TN**	True Negative
SVM	Support Vector Machine		
CNN	Convolutional Neural Network		
GPU	Graphics Processing Unit		
TF	TensorFlow		
FC	Fully Connected		
FCN	Fully Convolutional Network		
MLP	Multi-Layer Perceptron		
ReLU	Rectified Linear Unit		
VAD	Voice Activity Detection	**TP**	True Positive

INTRODUCTION GÉNÉRALE

La reconnaissance du locuteur, également connue sous le nom de reconnaissance vocale ou de reconnaissance de la personne basée sur la parole, est la capacité à distinguer la voix humaine et à identifier ou vérifier l'identité d'une personne sur la base des empreintes vocales et des caractéristiques acoustiques.

Elle ne doit pas être confondue avec la reconnaissance vocale qui traite de la conversion de l'audio en texte. Les deux font partie du même domaine mais ont des objectifs très différents. La reconnaissance vocale offre une plus grande accessibilité aux utilisateurs en leur permettant de communiquer plus facilement avec le système, tandis que la reconnaissance du locuteur consiste à vérifier l'identité de la personne afin que le système connaisse la personne avec laquelle il est en contact.

Avant d'approfondir le concept de reconnaissance du locuteur, il est essentiel de bien comprendre les différences entre la reconnaissance de la parole et la reconnaissance du locuteur, leurs applications respectives, et comment l'apprentissage machine peut être utilisé pour atteindre l'objectif de la reconnaissance du locuteur. Comme la reconnaissance vocale concerne la conversion de l'audio en texte, elle dépend fortement de la langue et du corpus. Cependant, la reconnaissance du locuteur ne tient pas compte de la langue dans la plupart des cas et se concentre davantage sur les perceptions audios brutes et les données connexes pour identifier le caractère unique de la façon dont les gens parlent. Le modèle d'identification du locuteur est formé de manière à pouvoir comprendre les modèles et les caractéristiques uniques des empreintes vocales et à pouvoir les différencier du reste.

C'est là que l'intelligence artificielle (IA) et l'apprentissage automatique (ML) sont utiles, et les chercheurs ont commencé à utiliser ces techniques pour former leurs modèles de reconnaissance du locuteur afin d'obtenir de meilleurs résultats.

Une approche de haut niveau consiste à recueillir des échantillons de la parole d'une personne, à extraire des caractéristiques de l'audio qui conviennent au classificateur, à former le classificateur à la construction du modèle et à effectuer la classification pour la reconnaissance et/ou l'identification. Dans ce projet, nous étudierons comment cette approche est actuellement utilisée de différentes manières

pour atteindre l'objectif final de la reconnaissance du locuteur et comment elle peut être améliorée davantage dans des scénarios plus difficiles.

CHAPITRE 1 : INTRODUCTION ET CONTEXT GENERALE DE PROJET

Ce chapitre présente la portée générale et le contexte du projet. Il décrit d'abord l'entreprise où j'ai réalisé mon stage, puis présente les principaux objectifs du projet. Il présente également les contraintes et les missions ainsi que la méthode et le processus de développement utilisés pour mener à bien le projet. Enfin, il illustre la planification du projet et les outils de collaboration utilisés.

1 INTRODUCTION

La reconnaissance du locuteur est le processus d'identification de la personne sur la base d'un audio contenant la voix de la personne. Il s'agit de la capacité d'une machine à recevoir un son ou une voix en entrée, à effectuer des calculs dessus et à déterminer qui est le locuteur. Les chercheurs travaillent sur la reconnaissance du locuteur depuis de nombreuses années, presque quatre décennies. Cependant, avec les progrès rapides de la technologie et l'essor sans précédent de l'Internet des objets (IoT), les appareils intelligents, les assistants vocaux et les assistants à domicile sont devenus de plus en plus populaires.

La parole, comme nous l'avons vu précédemment, est le moyen de communication le plus élémentaire pour l'homme. On peut donc affirmer sans risque que l'intégration la plus transparente de la communication entre l'homme et la machine peut également être réalisée par la parole. En outre, la reconnaissance du locuteur est plus facile à utiliser dans un environnement où il y a plusieurs locuteurs. À l'ère de l'IoT actuelle, lorsque plusieurs personnes parlent à un appareil intelligent ou à un assistant vocal, il est important que l'assistant ne se contente pas de comprendre ce qu'on lui dit, mais qu'il sache aussi qui est le locuteur - afin que des informations pertinentes et personnalisées puissent être fournies à l'utilisateur.

Ainsi, la reconnaissance du locuteur joue un rôle essentiel dans le monde actuel et la technologie future. Des recherches actives sont menées par des scientifiques travaillant dans le domaine de l'interaction homme-machine (IHM) pour déduire le son

reçu par une machine [1]. Les deux principaux domaines d'intérêt des experts en IHM travaillant sur l'audio sont la reconnaissance de la parole et la reconnaissance du locuteur. Comme nous l'avons vu précédemment, la reconnaissance de la parole est l'art d'entraîner une machine à comprendre ce qu'une personne parle, tandis que la reconnaissance du locuteur est l'art d'identifier qui parle. Ensemble, ces deux éléments sont d'une importance capitale.

Les technologies de l'information et de la communication (TIC) jouent un rôle important dans la réalisation d'une communication vocale sans faille et jouent donc un rôle clé dans l'interaction homme-machine dans le monde actuel.

La reconnaissance du locuteur est également un domaine de recherche actif dans le secteur de la biométrie. La biométrie des mots de passe sont généralement considérés comme peu sûrs et posent plusieurs problèmes. Comme les utilisateurs conservent généralement des mots de passe plus faciles à retenir, on constate souvent que les mots de passe sont faciles à craquer à l'aide de machines performantes et de techniques de force brute. L'autorisation à deux facteurs a été introduite pour tenter de résoudre ce problème et a été mise en œuvre il y a plusieurs années, le second facteur étant généralement une carte physique ou des jetons matériels tels qu'un jeton RSA. Ils sont bons, mais pas pratiques, car les gens sont tenus d'emporter un matériel supplémentaire avec eux. La biométrie tente de réduire cet inconvénient en authentifiant et en identifiant les personnes à l'aide des caractéristiques qu'elles possèdent déjà, telles que le visage, l'iris, la voix, la reconnaissance des empreintes digitales, et bien d'autres encore. C'est la raison principale pour laquelle une étude sur l'amélioration de la biométrie est cruciale et exige des recherches approfondies pour obtenir une plus grande précision.

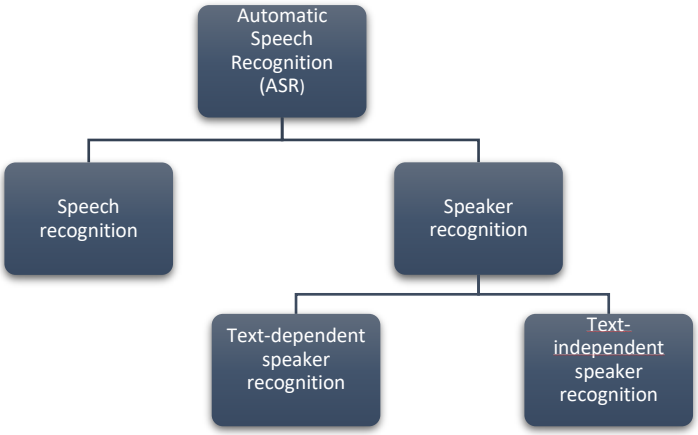


Figure 1 : Diagramme sur les types des ASR

2 PRESENTATION DE L'ORGANISME D'ACCUEIL

1.1.1 IDENTITE DE L'ENTREPRISE

Figure 2 : Logo De l'entreprise HENCEFORTH

HENCEFORTH (researcH and developmENt in advanCEd inFORmation TecHnology) est une jeune entreprise spécialisée dans les domaines du Big Data et de la Sécurité. Son siège est à Hay Riad à Rabat. Elle offre à ses clients des services de consultation et de développement de solutions liés à ces deux domaines. Elle reste constamment à proximité de ses clients pour un meilleur service basé sur les normes standards et aboutissant à des certifications.

1.2.2 PRINCIPAUX SERVICES

HENCEFORTH vise à relever de nouveaux défis technologiques en mettant l'accent sur la recherche et développement dans les nouveaux domaines liés à la technologie de l'information et de la communication de manière générale et au Big Data et la Sécurité des Systèmes d'Information en particulier. Elle vise à se tisser une place novatrice dans ces domaines aussi bien à l'échelle nationale qu'à l'échelle internationale.

Grâce à cette vision, HENCEFORTH propose à ses clients du conseil et de l'accompagnement dans la mise en place de solutions liées au domaine du Big Data avec toutes ses facettes et au domaine de la sécurité des systèmes d'information. Elle offre des solutions intégrées matériel, logiciel, formation, et conseils liés à ces domaines.

1.2.3 ORGANISATION ADMINISTRATIVE

HENCEFORTH est organisée en une direction générale dont dépend deux directions : la première concerne les affaires administratives et la deuxième concerne la direction recherche et développement. Elle offre ainsi à ses cadres et ingénieurs un cadre de travail agréable pour bien mener leurs activités de recherche et développement en collaboration avec des chercheurs de renommées aussi bien à l'échelle nationale qu'à l'échelle internationale.

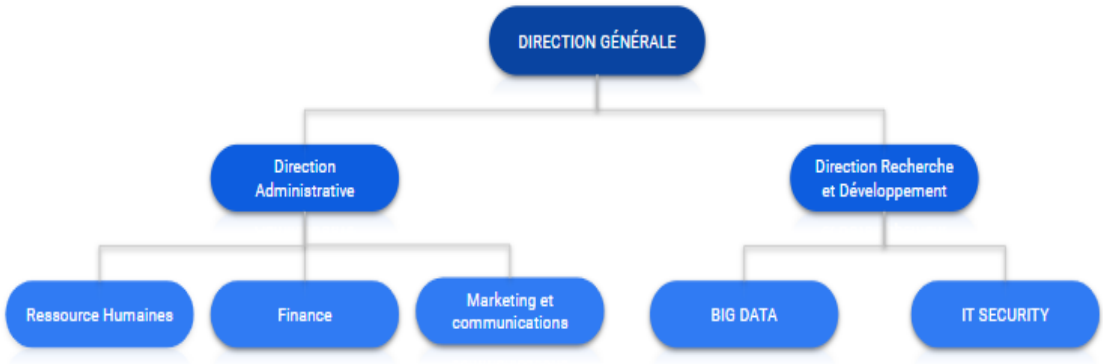


Figure 3 : L'architecture de l'organisation administrative de HENCEFORTH

3 CONTEXTE DE PROJET

3.1 MOTIVATION

L'interaction des humains avec les ordinateurs ne cesse de s'accroître et, dans le monde actuel, la plupart des appareils deviennent "intelligents" pour interagir efficacement avec les humains. Le concept de biométrie devient donc crucial pour l'étude et la recherche afin de vérifier si l'humain est bien l'identité prétendue.

Plusieurs techniques de biométrie sont utilisées depuis des temps immémoriaux. Les progrès les plus récents dans ce domaine ont toutefois été la reconnaissance à l'aide du visage, des empreintes digitales, de l'iris, de la géométrie de la main, de la voix et de quelques autres éléments. Bien que j'utilise moi-même des appareils intelligents, j'ai souvent l'impression que la voix est le moyen le plus non intrusif de communiquer avec la machine et de m'authentifier. Le scanner du visage et de l'iris est également pratique, mais pour cela, la personne doit être constamment devant la caméra, tandis que l'iris et la géométrie de la main exigent de l'utilisateur qu'il tienne physiquement l'appareil. La voix n'a pas de telles limitations, et nous savons qu'il existe une voix unique pour chaque individu.

Un exemple simple de cela pourrait être le déverrouillage du smartphone. Le visage, l'iris ou les empreintes digitales - tous exigent de l'utilisateur qu'il soit physiquement proche du matériel. Mais il n'en va pas de même pour le déverrouillage par la voix. En raison de la commodité qu'il apporte, les entreprises et les départements de recherche et développement étudient de près cette technique afin de rester pertinents dans le contexte actuel de la technologie de la biométrie.

Par conséquent, l'identification de la voix et la reconnaissance du locuteur sont des méthodes largement acceptées en biométrie pour l'authentification et/ou l'identification en raison de la commodité, de la flexibilité et de la praticabilité qu'elles offrent. En outre, plusieurs appareils utilisent actuellement la voix comme principal mode d'interaction, notamment les assistants vocaux tels que Google Home, Amazon Echo (Alexa), Siri d'Apple, Cortana de Microsoft, Bixby de Samsung, et bien d'autres encore. Ces appareils sont souvent utilisés dans les maisons où l'on s'attend à ce que plusieurs personnes interagissent avec eux. Ces appareils et technologies ayant gagné en popularité ces dernières années, il est important qu'ils puissent distinguer efficacement

les différents interlocuteurs qui pourraient leur parler. C'est précisément la raison pour laquelle une recherche sur la reconnaissance des locuteurs est importante et nous allons donc en discuter et l'étudier plus en détail.

3.2 OBJECTIF DE PROJET

Ma mission durant ce stage est la conception et la construction d'un modèle de Machine Learning qui permet de reconnaitre une personne à partir de sa propre voix, et pour cela j'ai réalisé comme objectifs les étapes suivantes :

- ✓ Collecter et préparer la DataSet.
- ✓ Faire un pré-traitement pour les fichiers audios.
- ✓ Extraire les caractéristiques pour chaque fichier audio.
- ✓ Construire le modèle de machine Learning pour la prédiction et la classification.
- ✓ Optimiser les performances de vitesse et de précision du modèle sur l'ensemble des données.
- ✓ Création d'une interface graphique.

3.3 STRUCTURE DE PROJET

La structure de notre projet est illustrée dans le diagramme suivant :

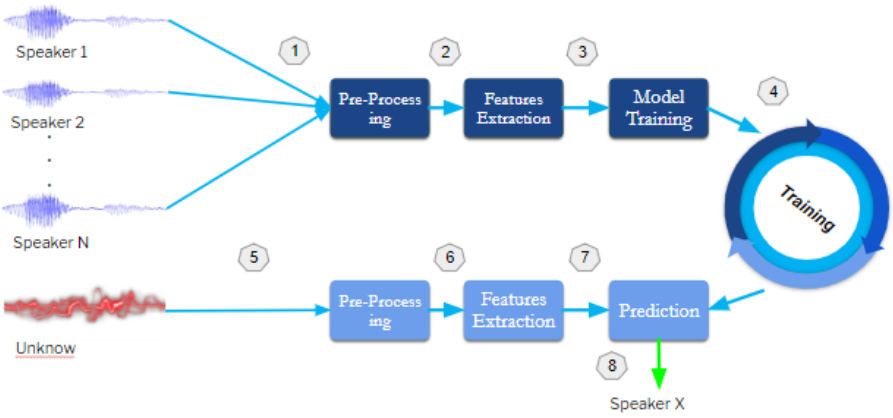


Figure 4 : Structure générale de projet

1. Les données audios d'entrés avec pour différents locuteurs.
2. Le pré-traitement des données d'entrées (réduction de bruite, Framing, Split, Trimming…)
3. L'Extraction des caractéristiques en utilisant l'une des techniques connues dans le secteur de traitement de signal (MFCC, LPC, PLP, etc.)
4. Entrainé notre modèle du machine Learning ou Deep Learning (KNN, RF, SVM, CNN...)
5. Le nouveau audio avec un locuteur inconnu à prédire
6. Pré-traitement pour le nouveau clip vocal
7. Extraire les caractéristiques du nouveau clip vocal
8. La prédiction de nouveau locuteur.

Après avoir présenté le contexte général de notre projet ainsi que les objectifs que nous voulons atteindre, nous aborderons les outils de gestion de projet et la planification de l'organisation de notre projet afin d'atteindre les objectifs définis.

4 GESTION DU PROJET

Chaque projet informatique nécessite un processus de développement bien défini pour garantir son succès. La pertinence du processus de développement du projet peut largement influencer sur le sort d'un projet informatique. Une procédure de développement mal choisie peut mener un projet à son échec.

4.1 METHODES AGILES

Les méthodes agiles sont basées sur toutes les valeurs qui ont été prises dans le Manifeste Agile, un texte écrit et signé par 17 experts pour leur contribution au développement d'applications informatiques :

- Priorité des personnes et des interactions sur les procédures et les outils ;
- Priorité des applications opérationnelles sur une documentation étendue ;
- Priorité du travail avec le client sur la négociation des contrats ;
- Priorité de l'acceptation des changements sur la planification.

Ces quatre valeurs sont à comparer avec les pratiques fréquemment rencontrées lors de la mise en œuvre des méthodes traditionnelles : priorité aux processus et aux outils, importance de la documentation, respect du contrat à la lettre, planification rigide.

Les méthodes agiles sont caractérisées par les éléments suivants :

- ✓ Livrer des versions opérationnelles rapidement et très fréquemment, afin de favoriser un retour d'information permanent de la part des clients ;
- ✓ Un changement bienvenu ;
- ✓ Assurer une coopération solide entre le client et les développeurs ;
- ✓ Maintenir un haut niveau de motivation ;
- ✓ Le fonctionnement de l'application est le premier indicateur du projet ;
- ✓ Garder un rythme durable ;
- ✓ Viser l'excellence technique et la simplicité ;
- ✓ Se remettre en question régulièrement.

4.2 OUTILS DE COLLABORATION

Pendant mon stage à HENCEFORTH, nous avons utilisé plusieurs outils qui nous ont permis d'atteindre les objectifs que nous nous étions fixés, ainsi que la communication avec nos superviseurs. Voici un bref aperçu de ces outils :

❖ Slack :

Afin de faciliter la communication et le partage des documents requis par le projet, un outil de collaboration professionnelle était nécessaire. Nous avons utilisé Slack pour notre projet car il s'agit d'une plateforme de communication collaborative ainsi que d'un logiciel de gestion de projet créé par Stewart Butterfield en août 2013 et officiellement lancé en février 2014.

La plateforme a l'avantage de s'intégrer facilement à d'autres services en ligne comme GitHub, Dropbox et Il permet également de faciliter la communication au sein de l'équipe et de centraliser le suivi et la gestion d'un projet.

Slack est la principale plateforme de communication au sein de l'usine numérique, nous l'avons surtout utilisée dans notre équipe, pour partager de nouvelles idées et de nouveaux dossiers ainsi que pour s'informer mutuellement des mises à jour et pour programmer des réunions.

❖ GitHub :

Pour nous collaborer et travaille en équipe a distant nous avons utilisé le GitHub. Le GitHub est une plateforme d'hébergement de code pour le contrôle de version et la collaboration. Elle vous permet, ainsi qu'à d'autres personnes, de travailler ensemble sur des projets, où que vous soyez, Le service GitHub a été développé par Chris Wanstrath, P. J. Hyett, Tom Preston-Werner et Scott Chacon en utilisant Ruby on Rails, et a débuté en février 2008. La société, GitHub, Inc. existe depuis 2007 et est située à San Francisco.

4.3 PLANIFICATIONS

La planification des projets est essentielle pour la gestion des projets, elle permet de :

- Définir les tâches à accomplir,
- Fixer des objectifs,
- Coordonner les actions,
- Gérer les ressources,
- Réduire les risques,
- Suivre les actions en cours,
- Rapport sur l'état d'avancement du projet.

Le diagramme de GANTT est un calendrier avec une liste de tâches en colonnes et en abscisse l'échelle de temps choisie. Il permet de visualiser facilement l'avancement du projet, ainsi que de planifier à l'avance les actions à envisager. Il permet également de gérer facilement les conflits de ressources et les retards éventuels en visualisant leur impact sur le projet.

Afin de mener à bien notre projet, nous avons établi le diagramme de GANTT, dans la figure ci-dessous.

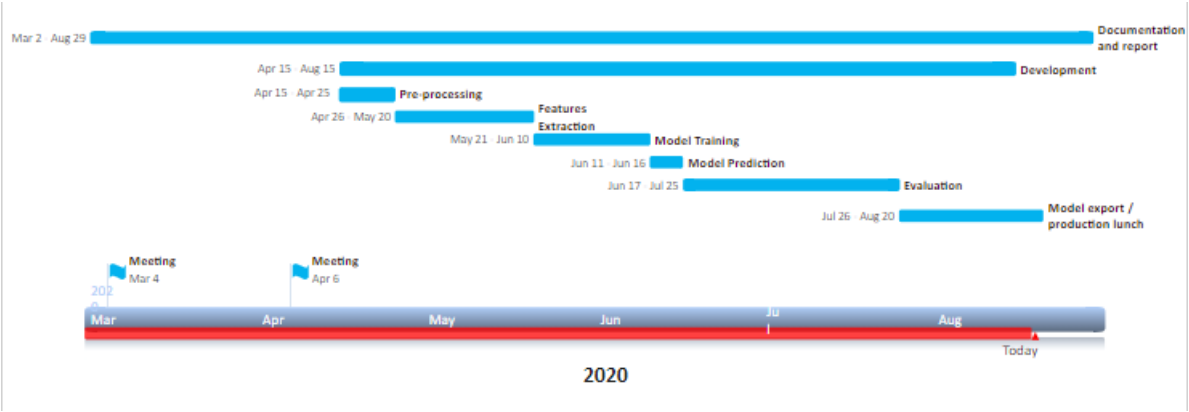


Figure 5 : Diagramme de Gante du projet

5 STRUCTURE DU DOCUMENT

Le reste de ce mémoire est organisé comme suit :

Dans le chapitre 2 on donne un bref aperçu sur la partie théorique de projet inclus les techniques de prétraitement, les techniques d'extraction des caractéristiques, et les techniques de machine Learning, en plus des travaux relatifs, ensuite le chapitre 3 décrit la Dataset qu'on a utilisé pour notre étude. Le chapitre 4 on présente les outils et bibliothèques qu'on a utilisés pour réaliser ce travail. Concernant le chapitre 5 on a discuté sur les mesures convenables pour notre étude. Dans Le chapitre 6 on a présenté les étapes pour implémenter notre projet dès le début jusqu'à la livraison de modèle. Une évaluation, une analyse et comparaison expérimentale se trouvent dans le chapitre 7, dans le dernier chapitre c'est-à-dire chapitre 8 vous aller trouver l'application web crée pour tester, et mettre à disposition le Système de reconnaissance vocale chez l'utilisateur.

CHAPITRE 2 : CONTEXTE THEORIQUE ET TRAVAUX RELATIFS

Dans ce chapitre, nous définirons chacun des techniques de prétraitement de signale, extraction des caractéristiques ainsi que les techniques de classification du machine Learning, et Deep Learning. Enfin, nous verrons certains travaux existant dans le secteur de la reconnaissance vocale, en particulier les modèles de classification qui sont mis en œuvre dans notre projet.

1. LES TECHNIQUES DE PRETRAITEMENT

Les fichiers audios utilisés pour l'ensemble de données ne sont pas enregistrés dans un environnement contraint et contiennent du bruit, généralement ambiant, et des pauses abruptes entre les mots et les phrases qui ne sont pas utiles pour l'ingénierie des caractéristiques audio, Par conséquent la phase liée au prétraitement audio est essentielle pour concevoir un bon modèle de classification des locuteurs.

1.1 LES TECHNIQUES DE TRAITMENT AUDIO

1.1.1 CONVERSION DES SONS EN BITS.

La première étape de la reconnaissance vocale est évidente, nous devons introduire des ondes sonores dans un ordinateur comme dans le traitement d'image.

La conversion A/D échantillonne les clips audios et numérise le contenu, c'est-à-dire qu'elle convertit le signal analogique en espace discret. Une fréquence d'échantillonnage de 8 ou 16 kHz est souvent utilisé.

Chaque échantillon est l'amplitude de l'onde à un intervalle de temps particulier, où la profondeur de bits détermine le degré de détail de l'échantillon, également connu sous le nom de gamme dynamique du signal (généralement 16 bits, ce qui signifie qu'un échantillon peut avoir une amplitude de 65 536 valeurs).

Les ondes sonores sont unidimensionnelles. À chaque instant, elles ont une valeur unique basée sur la hauteur de l'onde. Zoomons sur une petite partie de l'onde sonore et regardons.

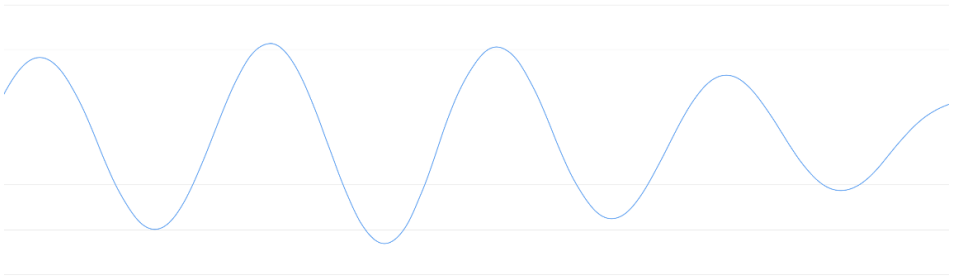


Figure 6 : Représentation d'un signal en ondes sonore

Pour transformer cette onde sonore en chiffres, il suffit d'enregistrer la hauteur de l'onde à des points équidistants :

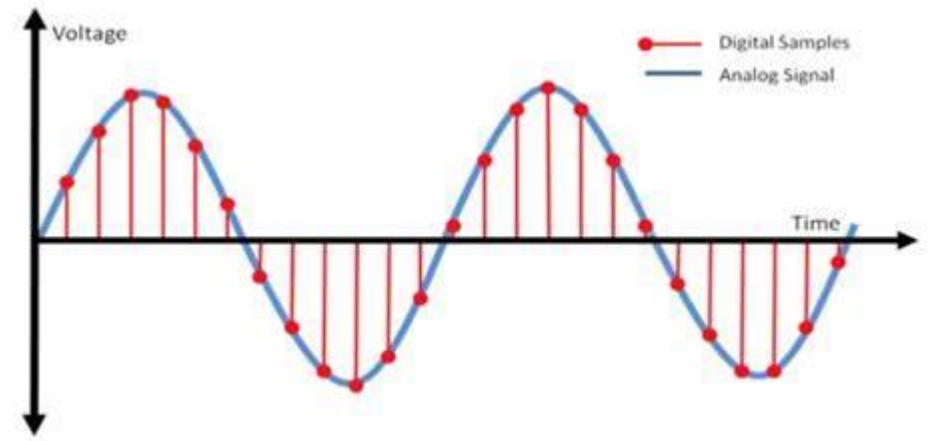


Figure 7 : Représentation d'un signal en ondes sonore avec un échantillonnage

C'est ce qu'on appelle l'échantillonnage. Nous effectuons une lecture des milliers de fois par seconde et enregistrons un nombre représentant la hauteur de l'onde sonore à ce moment-là.

L'audio de "qualité CD" est échantillonné à 44,1khz (44 100 lectures par seconde). Mais pour la reconnaissance vocale, un taux d'échantillonnage de 16khz (16 000 échantillons par seconde) est suffisant pour couvrir la gamme de fréquences de la parole humaine.

Échantillonnons un exemple d'un fichier audio 16 000 fois par seconde. Voici les 100 premiers échantillons :

[-1274, -1252, -1160, -986, -792, -692, -614, -429, -286, -134, -57, -41, -169, -456, -450, -541, -761, -1067, -1231, -1047, -952, -645, -489, -448, -397, -212, 193, 114, -17, -110, 128, 261, 198, 390, 461, 772, 948, 1451, 1974, 2624, 3793, 4968, 5939, 6057, 6581, 7302, 7640, 7223, 6119, 5461, 4820, 4353, 3611, 2740, 2004, 1349, 1178, 1085, 901, 301, -262, -499, -488, -707, -1406, -1997, -2377, -2494, -2605, -2675, -2627, -2500, -2148, -1648, -970, -364, 13, 260, 494, 788, 1011, 938, 717, 507, 323, 324, 325, 350, 103, -113, 64, 176, 93, -249, -461, -606, -909, -1159, -1307, -1544]

Figure 8 : Représentation numérique d'un signal

Quelle est la fréquence d'échantillonnage idéale pour l'enregistrement de morceaux ? Le théorème de Nyquist répondra.

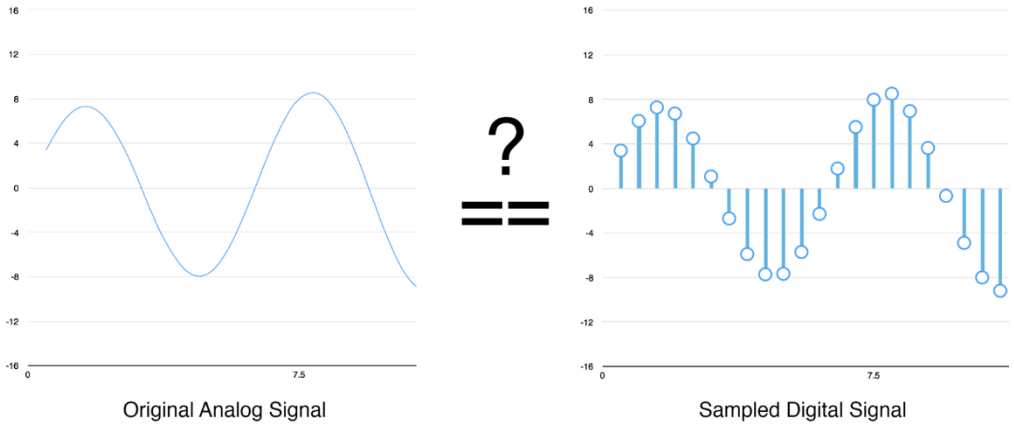


Figure 9 : Signal avec et sans échantillonnage

Théorème de l'échantillonnage Nyquist

Selon Nyquist [2] - Votre taux d'échantillonnage doit être au moins deux fois supérieur à la fréquence maximale que vous voulez capter du signal donné. En d'autres termes - Si on vous donne un signal ayant des fréquences allant de 1 à f Hz, et que vous ne voulez pas perdre d'informations (fréquence), votre taux d'échantillonnage (F) devrait être (F ≥2* f).

1.1.2 PREACCENTUATION

La préaccentuation augmente la quantité d'énergie dans les hautes fréquences. Pour les segments vocaux comme les voyelles, il y a plus d'énergie dans les basses fréquences que dans les hautes fréquences. C'est ce qu'on appelle l'inclinaison spectrale, qui est liée à la source glottale (la façon dont les plis vocaux produisent le son). L'augmentation de l'énergie dans les hautes fréquences rend l'information dans les formants supérieurs plus disponible pour le modèle acoustique. Cela améliore la précision de détection des téléphones. Chez l'homme, nous commençons à avoir des problèmes d'audition lorsque nous ne pouvons pas entendre ces sons de haute

fréquence. De plus, le bruit a une fréquence élevée. Dans le domaine de l'ingénierie, nous utilisons la préaccentuation pour rendre le système moins sensible aux bruits introduits plus tard dans le processus. Pour certaines applications, il suffit d'annuler la préaccentuation à la fin.

La préaccentuation utilise un filtre pour amplifier les fréquences plus élevées. Le filtre de préaccentuation peut être appliqué à un signal (x) en utilisant le filtre du premier ordre dans l'équation suivante :

$$Y[t] = X[t] - \alpha * X[t-1]$$

Où Y[t] est le signale préaccentuation, X[t] le signale originale a l'instant t, et les valeurs typiques du coefficient du filtre α sont entre 0,95 et 0,97. Vous trouverez ci-dessous le signal avant et après sur la façon dont le signal haute fréquence est amplifié.

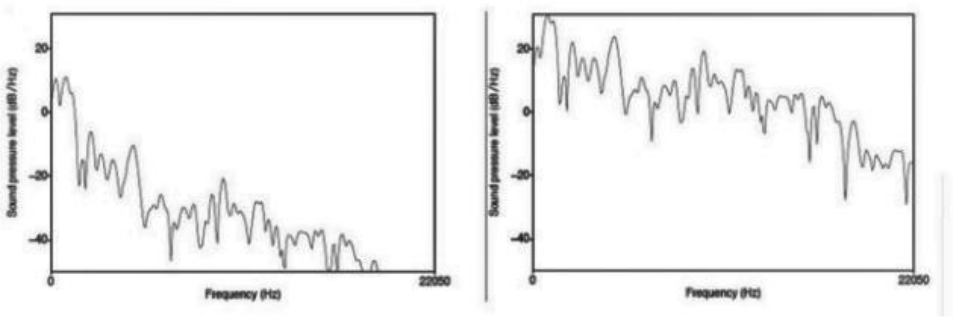


Figure 10 : La fréquence de signal avant et après La préaccentuation

1.1.3 FENETRAGE

Le fenêtrage est le concept qui consiste à diviser le signal en fragments de courte durée. Le raisonnement qui sous-tend de cette étape est que les fréquences d'un signal changent au cours du temps, de sorte que dans la plupart des cas, il n'est pas logique d'effectuer la transformation de Fourier sur l'ensemble du signal dans la mesure où nous perdrions les contours de fréquence du signal avec le temps. Pour éviter cela, nous pouvons supposer sans risque que les fréquences d'un signal sont stationnaires sur une très courte période de temps. Par conséquent, en effectuant une transformée de Fourier

sur cette courte période, nous pouvons obtenir une bonne approximation des contours de fréquence du signal en concaténant des fenêtres adjacentes.

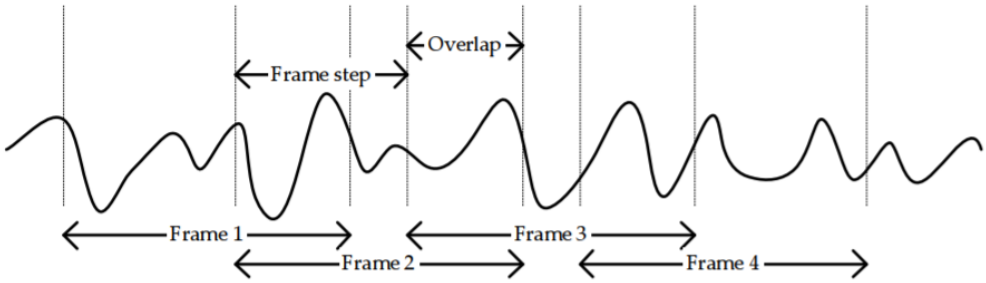


Figure 11 : La fragmentation d'un signal avec un chevauchement

Les tailles de trame typiques dans le traitement de la parole vont de 20 ms à 40 ms avec un chevauchement (Overlap) de 50% (+/-10%) entre des fenêtres consécutives. Les réglages les plus courants sont 25 ms pour la taille de la fenêtre, et une enjambée de 10 ms (chevauchement de 15 ms).

1.1.4 FONCTIONS DE FENETRAGE

Le fenêtrage consiste à découper la forme d'onde audio en fenêtres coulissantes, mais nous ne pouvons pas nous contenter de la couper au bord du fenêtre. La chute soudaine de l'amplitude créera beaucoup de bruit qui se manifestera dans les hautes fréquences. Pour découper le signal audio, l'amplitude doit diminuer progressivement près du bord du segment.

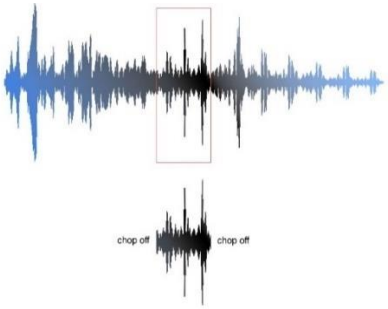


Figure 12 : Les bords d'un signal après le fenêtrage

Pour éviter ce problème nous utilisons les fonctions de fenêtrage, il existe plusieurs types des fonctions de fenêtrage, ici on cite trois types les plus connus dans le domaine de traitement de signal.

1.1.4.1 FENETRE RECTANGULAIRE

La fenêtre rectangulaire (parfois appelée fenêtre de Boxcar ou de Dirichlet) est la fenêtre la plus simple. Elle est équivalente au remplacement de toutes les valeurs d'une séquence de données, sauf N, par des zéros, ce qui donne l'impression que la forme d'onde s'allume et s'éteint soudainement :

$$W[n] = 1$$

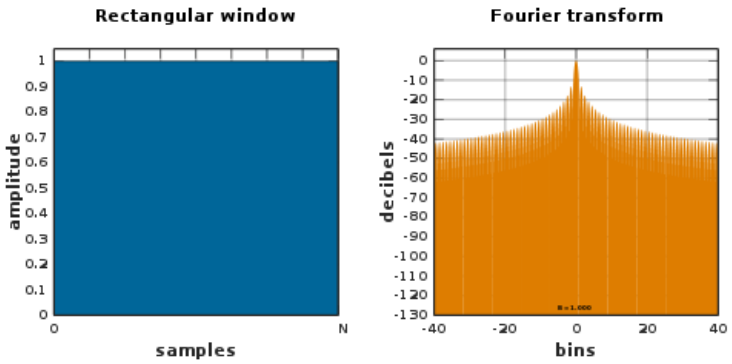


Figure 13 : Fonction de fenêtrage rectangulaire.

1.1.4.2 FENETRE DE HANNING

Nommé d'après Julius von Hann, et parfois appelé Hanning, probablement en raison de ses similitudes linguistiques et de formule avec la fenêtre de Hamming. Elle est également connue sous le nom de cosinus surélevé, parce que la version à phase zéro $w_0(n)$, est un lobe d'une fonction cosinus surélevée.

Cette fonction est un membre des familles cosinus-somme et puissance-sinus. Contrairement à la fenêtre de Hamming, les points d'extrémité de la fenêtre de Hann touchent juste zéro. Les lobes latéraux qui en résultent s'éteignent à environ 18 dB par octave.

$$W\,[n] = 0.5 * [1 - \cos(\frac{2\pi n}{N})]$$

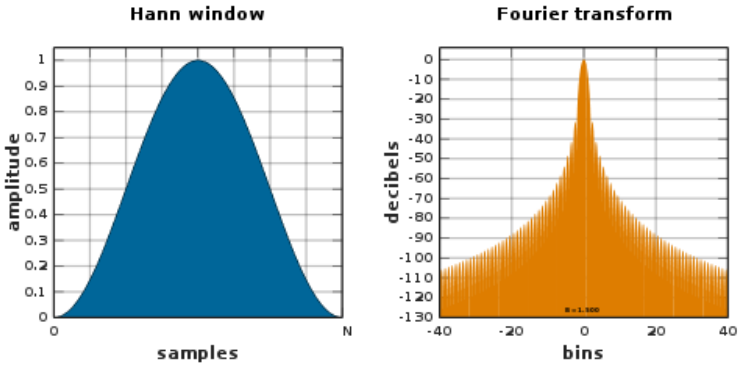


Figure 14 : Fonction de fenêtrage Hanning

.

1.1.4.3 FENETRE DE HAMMING

Le Hamming a été nommé d'après R. W. Hamming, un associé de J. W. Tukey et est décrit dans Blackman and Tukey. Il a été recommandé pour lisser la fonction d'autocovariance tronquée dans le domaine temporel. La plupart des références à la fenêtre de Hamming proviennent de la littérature sur le traitement du signal, où elle est utilisée comme l'une des nombreuses fonctions de fenêtrage pour lisser les valeurs. Elle est également connue sous le nom de fonction d'anodisation (qui signifie "suppression du pied", c'est-à-dire des discontinuités de lissage au début et à la fin du signal échantillonné) ou d'effilement.

L'approximation des coefficients à deux décimales abaisse sensiblement le niveau des lobes latéraux, jusqu'à une condition presque équirépartie. Dans le sens équiréparti, les valeurs optimales des coefficients sont a0 = 0,53836 et a1 = 0,46164.

$$W\,[n] = 0.54 - 046 * \cos(\frac{2\pi n}{N})$$

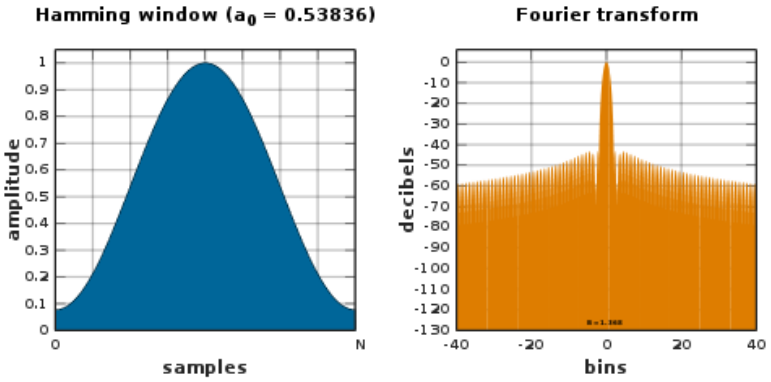


Figure 15 : Fonction de fenêtrage Hamming

1.1.5 DETECTION D'ACTIVITE VOCALE

La détection de l'activité vocale (VAD), également connue sous le nom de détection de l'activité de la parole (DAP), est une technique utilisée dans le traitement de la parole dans laquelle la présence ou l'absence de la parole humaine dans un signal sonore est détectée [3]. La détection de l'activité vocale joue un rôle essentiel dans les systèmes de traitement des signaux vocaux tels que le codage de la parole pour les téléphones cellulaires ou IP et le traitement frontal pour les applications de reconnaissance. Elle est également utilisée dans diverses techniques d'amélioration de la parole comme la réduction du bruit et l'annulation de certains échos.

1.1.5.1 LES TYPES DE VAD

Les signaux vocaux peuvent être analysés soit dans le domaine temporel, soit dans le domaine fréquentiel. Ainsi, les méthodes de traitement qui impliquent directement la forme d'onde du signal de parole sont appelées méthodes du domaine temporel. En revanche, les méthodes du domaine fréquentiel impliquent (explicitement ou implicitement) une certaine représentation spectrale. La figure présente un exemple de forme d'onde dans le domaine temporel et le spectre du même segment.

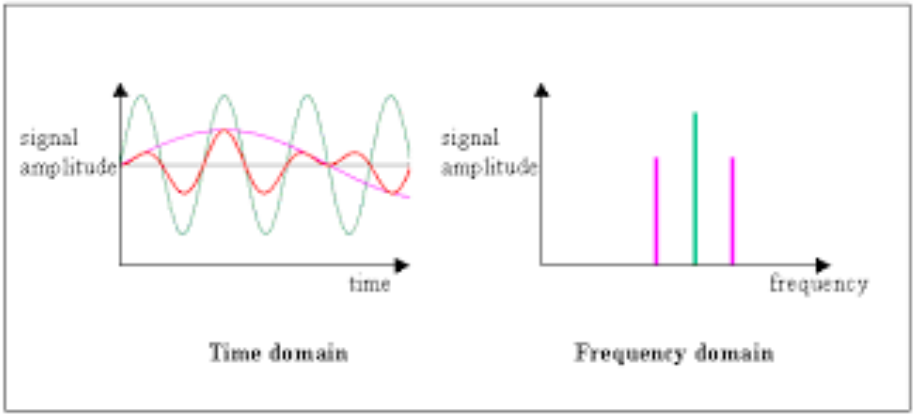


Figure 16 : Représentation du signal dans le domaine temporel et Fréquentiel

1.1.5.1.1 CARACTERISTIQUES DU DOMAINE TEMPOREL

Caractéristiques du domaine temporel fonctionnent raisonnablement bien dans des conditions de SNR élevé.

1.1.5.1.1.1 L'ENERGIE A COURT TERME

Nous avons observé que l'amplitude des signaux de parole varie sensiblement avec le temps. En particulier, l'amplitude des segments non vocaux est généralement beaucoup plus faible que celle des segments vocaux. L'énergie de courte durée du signal vocal fournit une représentation pratique qui reflète ces variations d'amplitude. En général, on peut définir l'énergie de courte durée comme :

$$E_n = \sum_{k=0}^{n} x^2\ k\ [n]\ h(n-k)$$

Où h[n] = w^2 [n] est la fenêtre d'analyse (fonction de fenêtrage) carrée appliquée sur un segment de parole. Nous pouvons supposer sans risque que la fenêtre d'analyse est prise en charge dans [-N, N]. Le choix de la réponse impulsionnelle, h[n], ou de manière équivalente la fenêtre d'analyse, détermine la nature de la représentation de l'énergie à court terme.

1.1.5.1.1.2 TAUX DE PASSAGE PAR ZERO A COURT DURÉE

Dans le contexte des signaux en temps discret, on dit qu'un passage par zéro se produit si un échantillon a un signe algébrique différent du précédent (ou du suivant). La vitesse à laquelle les passages à zéro se produisent est une mesure simple du contenu fréquent d'un signal. Cela est particulièrement vrai pour les signaux à bande étroite. Par exemple, un signal sinusoïdal de fréquence f0 Hz, échantillonné à une fréquence de Fs, a Fs/f0 échantillons par cycle de l'onde sinusoïdale. Chaque cycle comporte deux passages par zéro, de sorte que le taux moyen à long terme de passages par zéro est :

$$Z = \frac{2f_0}{F_s}$$

Ainsi, le taux moyen de passage par zéro donne un moyen raisonnable et simple d'estimer la fréquence d'une onde sinusoïdale.

Les signaux vocaux sont des signaux à large bande et l'interprétation du taux moyen de passage par zéro est donc beaucoup moins précise. Cependant, des estimations approximatives des propriétés spectrales peuvent être obtenues en utilisant une représentation basée sur le taux moyen de passage par zéro sur une courte période. Avant de discuter de l'interprétation du taux de passage par zéro pour la parole, définissons et discutons d'abord la théorie sous-jacente. Une représentation appropriée de La définition est la suivante :

$$Z_n = \sum_{m=-\infty}^{\infty} |sign\big(\mathrm{x(m)}\big) - \mathrm{sign(x(m-1))}|\mathrm{w[m-n]}$$

Où,

$$sign = \begin{cases} 1 & pour\ x > 0 \\ -1 & sinon \end{cases}$$

Et,

$$W[n] = \begin{cases} \frac{1}{2N} & pour\ 0 < n < N-1 \\ 0 & sinon \end{cases}$$

Cette représentation montre que le taux moyen de passage par zéro à court terme a les mêmes propriétés générales que l'énergie à court terme. Cependant, la définition

de l'équation de passage par zéro fait que le calcul de Zn semble plus complexe qu'elles ne le sont en réalité. Il suffit de vérifier les échantillons par paires pour déterminer où se produisent les passages à zéro, puis de calculer la moyenne sur N échantillons consécutifs (la division par N n'est évidemment pas nécessaire non plus).

1.1.5.1.1.3 ENVELOPPE DU SIGNAL

Pour une fenêtre donnée de N échantillons, MULSE est déduit comme suit :

$$M_n = |\max(S_n(i)).\min(S_n(i))| \; i \in (1 \dots N)$$

MULSE est une caractéristique temporelle calculée en multipliant les valeurs supérieures et inférieures de parties de l'enveloppe du signal.

1.1.5.1.2 CARACTERISTIQUES DU DOMAINE SPECTRAL

Lorsque le rapport signal/bruit (SNR) d'un signal est très faible (par exemple inférieur à 0 dB), le traitement dans le domaine temporel est difficile, car les valeurs des caractéristiques des trames vocales et non vocales ne varient pas autant qu'avec un SNR élevé.

1.1.5.1.2.1 SPECTRALE ENTROPIE

L'entropie est une mesure de probabilité des informations contenues dans un message. L'application du concept d'entropie au problème de la détection de la parole repose sur l'hypothèse que le spectre du signal est plus organisé pendant les blocs de parole que pendant les blocs non vocaux. Soit s(m) un signal vocal discret divisé en trames qui se chevauchent et soit Sn(f) - le spectre de magnitude de la nième trame pour la fréquence bin f.

La mesure de l'entropie est déduite dans le domaine de l'énergie spectrale comme suit :

$$H(|S_n(f)|^2) = \sum_{f=1}^{\infty} P(|S_n(f)|^2).\ln(P(|S_n(f)|^2))$$

Où,

$$P(|S_n(f)|^2) = \frac{|S_n(f)|^2}{\sum_{k=1}^{\infty} |S_n(k)|}$$

Est la probabilité du spectre de magnitude de la f ième bande dans la trame k. Elle est appelée fonction de masse de probabilité (PMF) et désigne la probabilité qu'une variable aléatoire discrète X prenne une valeur de xi, P (X = xi).

1.1.5.1.2.2 CENTROÏDE SPECTRAL

Le centroïde spectral est une mesure utilisée dans le traitement numérique des signaux pour caractériser un spectre. Il indique où se trouve le centre de masse du spectre. Du point de vue de la perception, il a un lien étroit avec l'impression de luminosité d'un son.

Il est calculé comme la moyenne pondérée des fréquences présentes dans le signal, déterminée à l'aide d'une transformée de Fourier, avec leurs magnitudes comme poids

$$C_n = \frac{\sum_{n=0}^{N-1} f(n)x(n)}{\sum_{n=0}^{N-1} x(n)}$$

Où x(n) représente la valeur de la fréquence pondérée, ou magnitude, du numéro de bin n, et f(n) représente la fréquence centrale de cette bin.

1.1.5.1.2.3 LE FLUX SPECTRAL

Mesure le changement spectral entre deux trames successives et est calculé comme la différence au carré entre les magnitudes normalisées des spectres des deux fenêtres à court terme successives :

$$F = \sum_{k=1}^{W} (En_i(k) - En_{i-1}(k))^2$$

Où ;

$$En_i(k) = \frac{X_i(k)}{\sum_{l=1}^{w} X_i(l)}$$

C-à-d, En est le ième coefficient de DFT normalisé à la ième trame.

1.1.5.2 ALGORITHME INDUSTRIEL DE VAD

IL existe plusieurs algorithmes de VAD dans l'industrie, dans la suite on cite quelques 'une qui sont très pratique.

1.1.5.2.1 G.729

L'Union internationale des télécommunications (UIT) a adopté un algorithme de codage de la parole de qualité téléphonique appelé "Conjugate Structure Algebraic code Excited Linear Prediction" (CS-ACELP). La recommandation correspondante est connue sous le nom de G.729. La recommandation de l'annexe B décrit un algorithme VAD qui est utilisé comme frontal dans la famille des codecs G.729 [4].

Le G.729 utilise les caractéristiques suivantes pour prendre des décisions concernant l'activité vocale :

- Fréquences spectrales de ligne (LFS) - un ensemble de coëxiens de prédiction linéaire est dérivé des 11 premiers termes de l'autocorrélation en utilisant les procédures G.729 (Annexe A), qui sont ensuite converties en un ensemble de LFS.
- Énergie à large bande
- Énergie en bande basse, mesurée à la bande 0-1 kHz
- Taux de passage à zéro (ZCR)

Le VAD G.729 fonctionne à une fréquence de trame de 10 ms. Les paramètres de différence sont calculés en soustrayant les valeurs des caractéristiques de la trame actuelle de la moyenne courante de chaque caractéristique. Ces variables forment les points générés par les trames de voix active sont regroupées dans une certaine région (hypervolume) de l'espace quadridimensionnel, tandis que les points générés par les trames de voix inactive sont regroupés dans une autre région (les régions peuvent se chevaucher). Une frontière de décision linéaire tridimensionnelle par morceaux identifie la région de voix inactive et celle de voix active qui la complète. On utilise quatorze hyperplans, chacun creusant une section de la limite de décision. Les

paramètres de chaque hyperplan ont été déterminés par une inspection visuelle de la distribution des points sur un grand corpus, à l'aide de diagrammes de dispersion. Bien que la méthode d'inspection visuelle soit la plus facile à réaliser, elle ne garantit pas du tout la meilleure performance.

Enfin, la décision VAD est adoucie pour tenir compte de la nature stationnaire du signal vocal et du bruit de fond. Ce lissage et cette correction utilisent quatre étapes de règles heuristiques qui résultent d'observations approfondies de la décision VAD initiale

1.1.5.2.2 MULTI-TAUX ADAPTATIF (AMR)

Le codec audio AMR (Adaptive multi-rate) est un système breveté de compression des données audio optimisé pour le codage de la parole [5]. La norme EN 301 708 de l'Institut européen des normes de télécommunications (ETSI) décrit deux algorithmes de détection de l'activité vocale adoptés pour l'AMR.

L'algorithme AMR VAD de type **I** utilise les caractéristiques suivantes pour la détection de l'activité vocale :

- Banque de filtres et 9 niveaux d'énergie de sous-bande.
- Tonalité. La fonction de détection du pitch a pour but de détecter les sons des voyelles et autres signaux périodiques.
- Tonalité. La détection de la tonalité est utilisée pour détecter les tonalités d'information (par exemple, les tonalités de progression d'appel, telles que la sonnerie ou la tonalité d'occupation), car la fonction de détection de la hauteur ne peut pas toujours détecter ces signaux.

L'AMR VAD comprend également l'analyse de signaux complexes corrélés, qui est utilisée pour détecter des signaux corrélés, comme la musique, car les fonctions de détection de la hauteur et de la tonalité ne peuvent pas toujours détecter ces signaux.

La décision de VAD intermédiaire est prise pour chaque trame de 20 ms et est calculée sur la base de la comparaison de l'estimation du bruit de fond et des niveaux de caractéristiques de la trame d'entrée. Enfin, la VAD ag est calculée en ajoutant Hangover à la décision VAD intermédiaire.

L'algorithme AMR VAD de type **II** utilise les niveaux d'énergie des sous-bandes et le SNR calculé dans le domaine spectral. Les décisions VAD intermédiaires sont prises toutes les 10 ms, et la décision NAL est calculée pour une trame de 20 ms.

1.1.6 CONCLUSION

Nous avons présenté différentes approches et techniques pour le prétraitement du signal, et chaque une de ces techniques a ses avantages et ses inconvénients, et dans notre travail nous utiliser certaines de ces techniques qui sont efficaces et simples pour obtenir plus de précision.

2 EXTRACTION DES CARACTERISTIQUES

L'extraction de caractéristiques dans les ASR est le calcul d'une séquence de vecteurs de caractéristiques qui fournit une représentation compacte du signal vocal donné. Elle est généralement effectuée en trois étapes principales.

La première étape est appelée analyse de la parole ou frontend acoustique, qui effectue une analyse spectrale et temporelle du signal de parole et génère des caractéristiques brutes décrivant l'enveloppe du spectre de puissance des courts intervalles de parole. La deuxième étape compile un vecteur de caractéristiques étendues composé de caractéristiques statiques et dynamiques. Enfin, la dernière étape transforme ces vecteurs de caractéristiques étendues en vecteurs plus compacts et plus robustes qui sont ensuite fourni comme une entrer au modèle de machine Learning pour la reconnaissance vocale.

Les caractéristiques de la parole largement utilisées pour la modélisation auditive sont les coefficients cepstraux obtenus par le codage prédictif linéaire (LPC). Une autre extraction de la parole bien connue est basée sur les coefficients cepstraux de fréquence Mel (MFCC). Les méthodes basées sur la prédiction perceptuelle, qui est bonne dans des conditions de bruit, sont le PLP et le RASTA-PLP (Filtrage des spectres relatifs des coefficients du domaine logarithmique). Il existe d'autres méthodes comme le RFCC, LSP, etc. pour extraire des caractéristiques de la parole. Mais MFCC, LPC et PLP sont les techniques les plus utilisés dans le domaine du traitement de la parole.

2.1 MFCC

Les coefficients cepstraux de la fréquence Mel sont un ensemble de caractéristiques audio spectrales qui sont efficaces dans les systèmes de reconnaissance de la parole/du locuteur. P. Mermelstein, J.S. Bridle et M.D. Brown sont les principaux auteurs de l'idée des MFCC [6]. Les MFCC sont une liste de coefficients qui dans leur totalité, représentent un cepstre de fréquence de Mel (MFC).

Les principaux objectifs des MFCC sont les suivants :

- Supprimer l'excitation des cordes vocales (F0) - l'information sur la hauteur.
- Rendre les caractéristiques extraites indépendantes.
- Ajuster la manière dont les humains perçoivent le volume et la fréquence du son.
- Saisir la dynamique des phones (le contexte).

Les étapes pour identifier les MFCC sont les suivantes :

- Un signal audio change constamment, donc pour simplifier les choses, nous supposons que sur des échelles de temps courtes, le signal audio ne change pas beaucoup (quand nous disons qu'il ne change pas, nous voulons dire statistiquement, c'est-à-dire statistiquement stationnaire, évidemment les échantillons changent constamment, même sur des échelles de temps courtes). C'est pourquoi nous cadrons le signal dans des trames de 20 à 40 ms (Fenêtrage). Si la trame est beaucoup plus courte, nous n'avons pas assez d'échantillons pour obtenir une estimation spectrale fiable, si elle est plus longue, le signal change trop au cours de la trame.
- L'étape suivante consiste à calculer le spectre de puissance de chaque fenêtre. Cette opération est motivée par la cochlée humaine (un organe de l'oreille) qui vibre à différents endroits en fonction de la fréquence des

sons entrants. Selon l'endroit de la cochlée qui vibre (qui fait onduler les petits poils), différents nerfs s'enflamment pour informer le cerveau que certaines fréquences sont présentes. Notre estimation du parodogramme (Transformation du Fourier) fait un travail similaire pour nous, en identifiant quelles fréquences sont présentes dans la fenêtre.

$$S_i(k) = \sum_{n=1}^{N} s_i(n)h(n)e^{-j2\pi kn/N}$$

Où h(n) est une fenêtre d'analyse pour le Nième échantillon.

En outre, la puissance du spectre est calculée comme suit

$$P(k)_i = \frac{1}{N}|S_i(k)|^2$$

- L'estimation spectrale du parodogramme contient encore beaucoup d'informations qui sont inutiles pour la reconnaissance automatique de la parole (ASR). En particulier, la cochlée ne peut pas distinguer la différence entre deux fréquences très proches l'une de l'autre. Cet effet devient plus marqué à mesure que les fréquences augmentent. C'est pourquoi nous prenons des groupes de bins de parodogrammes et nous les additionnons pour avoir une idée de la quantité d'énergie qui existe dans les différentes régions de fréquences. Cette opération est réalisée par notre Mel Filtres Bank : le premier filtre est très étroit et donne une indication de la quantité d'énergie existante près de 0 Hertz. Plus les fréquences sont élevées, plus nos filtres s'élargissent et moins nous nous préoccupons des variations. Nous ne sommes intéressés que par la quantité approximative d'énergie présente à chaque point. L'échelle de Mel nous indique exactement comment répartir nos Bank filtres et quelle est leur largeur.

  La formule pour convertir les hertz en mels est donnée comme suit :

$$M = 2595\ \log_{10}\left(1 + \frac{f}{700}\right) = 1127\ln\left(1\frac{f}{700}\right)$$

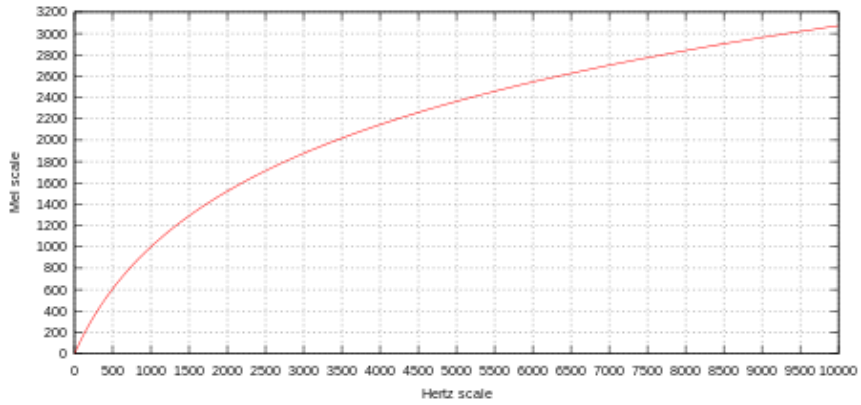


Figure 17 : La représentation du Mel scale Frequency

Chaque filtre de la Bank de filtres est triangulaire, avec une réponse de 1 à la fréquence centrale et diminue linéairement vers 0 jusqu'à atteindre les fréquences centrales des deux filtres adjacents où la réponse est de 0, comme le montre cette figure :

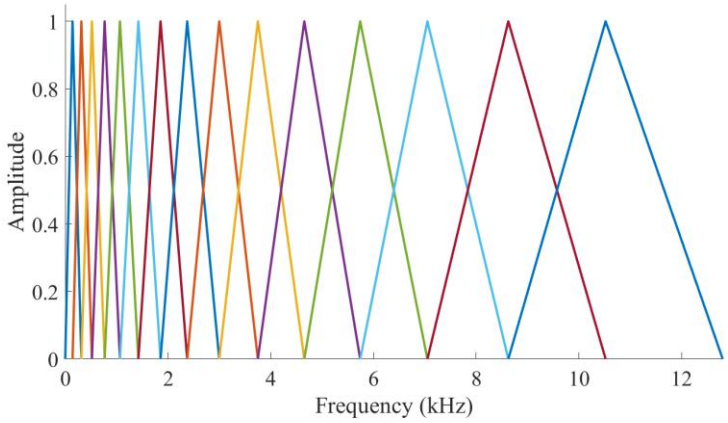


Figure 18 : Le filtre triangulaire de Filtre Bank

- Une fois que nous avons les énergies du filtre-Bank, nous prenons le logarithme de celles-ci. Ceci est également motivé par l'audition humaine : nous n'entendons pas le son sur une échelle linéaire. En général, pour doubler le volume perçu d'un son, nous devons y mettre 8 fois plus d'énergie. Cela signifie que de grandes variations d'énergie peuvent ne pas sembler si différentes si le son est fort au départ. Cette opération de compression fait que nos caractéristiques correspondent plus étroitement

à ce que les humains entendent réellement. Pourquoi le logarithme et non une racine cubique ? Le logarithme nous permet d'utiliser la soustraction de la moyenne cepstrale, qui est une technique de normalisation des canaux.

- L'étape finale consiste à calculer la Discret Cosinus Transformers (DCT) des énergies du filtre logarithmique. Il y a deux raisons principales pour lesquelles cette étape est effectuée. Comme nos Bank de filtres se chevauchent toutes, les énergies des Bank de filtres sont très corrélées entre elles. La DCT décorrèle les énergies, ce qui signifie que les matrices de covariance diagonale peuvent être utilisées pour modéliser les caractéristiques, Mais notez que seuls 12 -20 des 26 coefficients de la DCT sont conservés. Cela s'explique par le fait que les coefficients DCT plus élevés représentent des changements rapides dans les énergies des filtres et qu'il s'avère que ces changements rapides dégradent en fait les performances des ASR, nous obtenons donc une petite amélioration en les abandonnant.

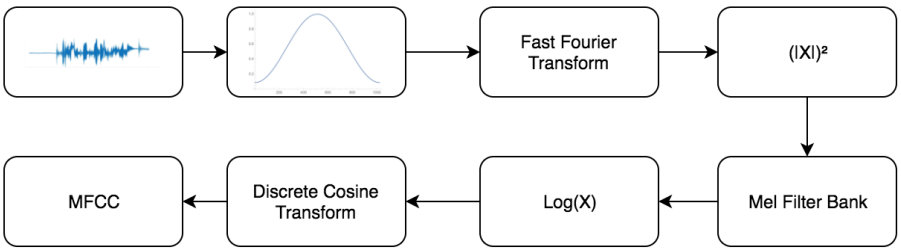


Figure 19 : Le processus d'extraire le MFCC

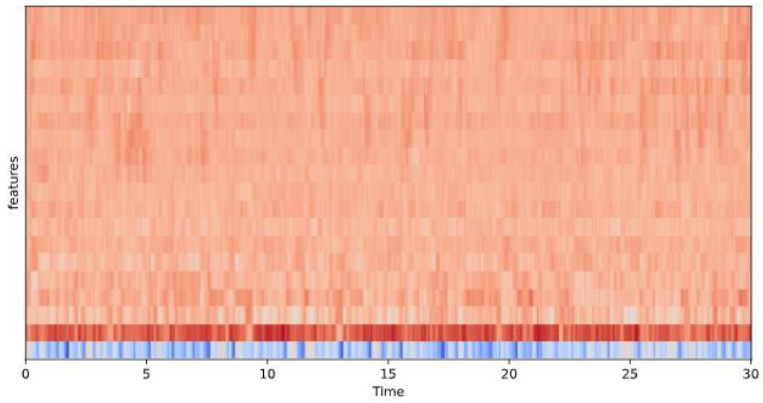


Figure 20 : Représentation des MFCCs en tant que spectrogramme

2.1.1 MFCC Delta : Différentiels

Au-delà de l'ensemble standard de coefficients connu sous le nom de MFCC, il existe également des variations de ce dernier. Les plus importantes sont celles qui sont obtenues en calculant les deltas des coefficients. Ils représentent le chemin que les MFCC rencontrent, qui est connu pour augmenter la précision des recherches sur la reconnaissance vocale en général. Les deltas des MFCC de premier ordre sont parfois aussi appelés les différentiels. La formule de calcul des coefficients des deltas du MFCC est la suivante :

$$d_t = \frac{\sum_{n=1}^{N} n(c_{t+n} - c_{t-n})}{2\sum_{n=1}^{N} n^2}$$

Où d_t est le coefficient de la valeur delta pour la fenêtre t, et c représente les coefficients MFCC.

2.1.2 MFCC DELTA-DELTA : ACCELERATIONS

De même, les coefficients delta-delta du MFCC sont calculés lorsque la formule est appliquée sur les coefficients delta du MFCC. On les appelle les coefficients MFCC delta-delta, parfois aussi appelés accélérations. La formule de calcul des coefficients delta-delta du MFCC est la même que celle des coefficients delta du MFCC, la seule modification étant que nous utilisons ici les coefficients delta au lieu des coefficients MFCC d'origine. Ainsi, la formule est :

$$dd_t = \frac{\sum_{n=1}^{N} n(d_{t+n} - d_{t-n})}{2\sum_{n=1}^{N} n^2}$$

Où ddt est le coefficient delta-delta pour la fenêtre t, et dt représente les coefficients delta MFCC.

Pour simplifier, le calcul du delta-delta du MFCC peut également être représenté comme suit : *MFCC delta-delta = Δ (MFCC delta) = Δ (Δ (MFCC)))* où Δ représente la fonction delta.

2.2 LPC

Les coefficients de prédiction linéaire (LPC) imitent les tractus vocaux humains et donnent une caractéristique de parole robuste. Il évalue le signal de parole en se rapprochant des formants, en éliminant ses effets du signal de parole et en estimant la concentration et la fréquence du reste du résidu. Le résultat indique chaque échantillon du signal comme une incorporation directe des échantillons précédents. Les coefficients de l'équation de différence caractérisent les formants, donc, le LPC à besoin d'approximer ces coefficients [7]. Le LPC est une méthode puissante d'analyse de la parole et a acquis une certaine notoriété comme étant une méthode d'estimation des formants [8] .

Les fréquences où se produisent les crêtes de résonance sont appelées les fréquences des formants. Ainsi, avec cette technique, les positions des formants dans un signal de parole sont prévisibles en calculant les coefficients de prédiction linéaire au-dessus d'une fenêtre coulissante et en trouvant les crêtes dans le spectre du filtre de prédiction linéaire. La LPC est utile pour le codage de la parole de haute qualité à un débit binaire faible.

Les autres caractéristiques qui peuvent être déduites de la LPC sont les coefficients cepstraux de prédiction linéaire (LPCC), le rapport de surface logarithmique (LAR), les coefficients de réflexion (RC), les fréquences spectrales de ligne (LSF) et les coefficients sinusoïdaux d'Arcus (ARCSIN). Le LPC est généralement utilisé pour la reconstruction de la parole. La méthode LPC est généralement appliquée dans les entreprises musicales et électriques pour la création de

robots mobiles, dans les entreprises de téléphonie, l'analyse tonale des violons et autres gadgets musicaux à cordes.

La méthode de prédiction linéaire est appliquée pour obtenir les coefficients de filtre équivalents au tractus vocal en réduisant l'erreur quadratique moyenne entre la parole d'entrée et la parole estimée [9]. L'analyse de prédiction linéaire du signal de parole prévoit tout échantillon de parole donné à une période spécifique comme une agrégation linéaire pondérée des échantillons précédents. Le modèle de prédiction linéaire de la création de la parole est donné par

$$s'(n) = \sum_{k=1}^{p} a_k s(n-k)$$

Où $\mathbf{s}'$est l'échantillon prédit, **s** est l'échantillon de parole, **p** est les coefficients de prédiction.

L'erreur de prédiction est donnée par :

$$e(n) = s(n) - s'(n)$$

Par la suite, chaque fenêtre du signal fenêtré est auto-corrélée, tandis que la valeur d'autocorrélation la plus élevée est de l'ordre de l'analyse de prédiction linéaire. Cette analyse est suivie de l'analyse LPC, où chaque fenêtre d'autocorrélation est convertie en un ensemble de paramètres LPC qui se compose des coefficients LPC. Un résumé de la procédure d'obtention du LPC est présenté à la figure 2. Le LPC peut être obtenu par

$$a_m = \log\left[\frac{1-k_m}{1+k_m}\right]$$

Où $\mathbf{a_m}$ est le coefficient de prédiction linéaire, $\mathbf{k_m}$ est le coefficient de réflexion.

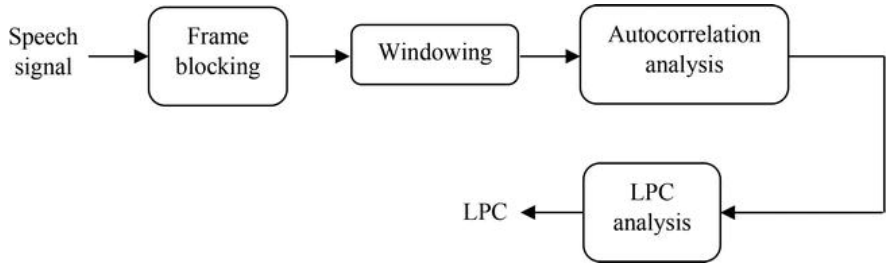


Figure 21 : Le processus d'extraire les LPC

L'analyse prédictive linéaire sélectionne efficacement les informations du tractus vocal d'un discours donné. Elle est connue pour sa rapidité de calcul et sa précision. LPC représente parfaitement les comportements de la source qui sont stables et cohérents. Dont le but principal est d'extraire les propriétés du tractus vocal. Il donne des estimations très précises des paramètres de la parole et est comparativement efficace pour le calcul.

2.3 PLP

La technique de prédiction linéaire perceptuelle (PLP) combine les bandes critiques, la compression de l'intensité sonore et la préaccentuation de l'intensité sonore égale dans l'extraction des informations pertinentes de la parole. Elle est basée sur l'échelle d'écorce non linéaire et était initialement destinée à être utilisée dans des tâches de reconnaissance vocale en éliminant les caractéristiques dépendantes du locuteur.

Le PLP donne une représentation conforme à un spectre à court terme lissé qui a été égalisé et comprimé de manière similaire à l'audition humaine, ce qui le rend similaire au MFCC. Dans l'approche PLP, plusieurs caractéristiques importantes de l'audition sont reproduites et le spectre auditif de la parole qui en résulte est approximé par un modèle autorégressif à tous les pôles [10]. Le PLP donne une résolution minimale aux hautes fréquences, ce qui signifie que l'approche est basée sur une Bank de filtres auditifs, mais donne des résultats orthogonaux similaires à l'analyse cepstrale. Il utilise des prédictions linéaires pour le lissage spectral, d'où son nom de prédiction

linéaire perceptuelle. Le PLP est une combinaison de l'analyse spectrale et de l'analyse de prédiction linéaire.

Afin de calculer les caractéristiques du PLP, la parole est fenêtrée (fenêtre de Hamming), la transformée de Fourier rapide (FFT) et le carré de la magnitude sont calculés. Cela donne les estimations spectrales de puissance. Un filtre trapézoïdal est ensuite appliqué à un intervalle d'un niveau pour intégrer les réponses des filtres à bande critique qui se chevauchent dans le spectre de puissance. Cela permet de comprimer efficacement les hautes fréquences dans une bande étroite. La convolution symétrique du domaine fréquentiel sur l'échelle des fréquences déformées par l'écorce permet alors aux basses fréquences de masquer les hautes fréquences, lissant simultanément le spectre. Le spectre est ensuite préaccentué pour se rapprocher de la sensibilité inégale de l'audition humaine à diverses fréquences. L'amplitude spectrale est comprimée, ce qui réduit la variation d'amplitude des résonances spectrales. Une transformation de Fourier discrète inverse (IDCT) est effectuée pour obtenir les coefficients d'autocorrélation. Un lissage spectral est effectué, ce qui permet de résoudre les équations autorégressives. Les coefficients autorégressifs sont convertis en variables cepstrales. L'équation pour le calcul de la fréquence de l'échelle d'écorce est :

$$Bark(f) = \frac{26.81\, f}{1960 + f} - 0.53$$

Où Bark(f) est la fréquence (Bark) et f est la fréquence (Hz).

L'identification obtenue par le PLP est meilleure que celle du LPC, car il s'agit d'une amélioration par rapport au LPC conventionnel car il supprime efficacement les informations dépendantes du locuteur. De plus, il a amélioré les performances de reconnaissance indépendante du locuteur et est robuste au bruit, aux variations du canal et aux microphones. Le PLP reconstruit avec précision la composante de bruit autorégressive. Le frontal basé sur le PLP est sensible à tout changement de la fréquence du formant.

La figure montre le processeur PLP, en indiquant toutes les étapes à suivre pour obtenir les coefficients PLP. Le PLP a une faible sensibilité à l'inclinaison spectrale, ce qui correspond aux conclusions selon lesquelles il est relativement insensible aux jugements phonétiques de l'inclinaison spectrale. De plus, l'analyse PLP dépend du

résultat de l'équilibre spectral global (amplitudes des formants). Les amplitudes des formants sont facilement affectées par des facteurs tels que l'équipement d'enregistrement, le canal de communication et le bruit additif. De plus, la résolution temps-fréquence et l'échantillonnage efficace de la représentatLJ0ion à court terme sont traités de manière ad hoc.

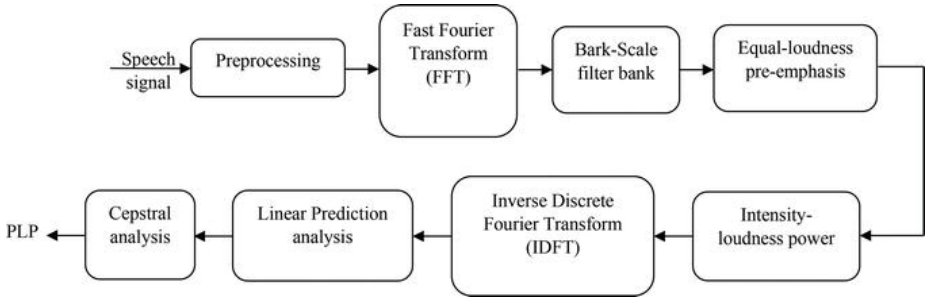


Figure 22 : Le processus d'extraire les PLP

2.4 CONCLUSION

Le tableau ci-dessous présente une comparaison entre les trois techniques d'extraction de caractéristiques qui ont été explicitement décrites ci-dessus. Même si la sélection d'un algorithme d'extraction de caractéristiques pour la recherche dépend de chaque individu, ce tableau a pu caractériser ces techniques en se basant sur les principales considérations dans la sélection de tout algorithme d'extraction de caractéristiques. Ces considérations comprennent la vitesse de calcul, la résistance au bruit et la sensibilité au bruit supplémentaire.

	Type de Filtre	**Dimension de Filtre**	**Ce qui est modélisé**	**Vitesse de calcule**	**Type des coefficients**	**Resistance de bruit**	**Sensibilité à la quantification/au bruit supplémentaire**	**Fiabilité**	**Fréquence capturé**
MFCC	Mel	Triangulaire	Systèmes auditive humain	Élevé	Cepstrale	Medium	Medium	Élevé	Basse
LPC	Prédiction Linéaire	Linéaire	Tractus vocale humain	Élevé	Autocorrélation Coefficients	Élevé	Élevé	Élevé	Basse

PLP	Bark	Trapézoïdale	Systèmes auditive humain	Medium	Cepstrale et Autocorrélation	Medium	Medium	Medium	Basse et Medium

Tableau 1 : Comparaisons entre les techniques d'extraction des caractéristiques

3 MACHINE LEARNING CLASSIFIEURS

Tout comme les techniques d'extraction de caractéristiques, les classificateurs d'apprentissage machine jouent également un rôle essentiel dans la détermination de l'efficacité globale du modèle de reconnaissance du locuteur. Comme nous avons l'intention de classer les audios et d'y déterminer le locuteur, il s'agit d'un problème de classification et c'est pourquoi nous parlerons de certains algorithmes efficaces d'apprentissage de machines de classification supervisées.

3.1 SUPPORT VECTOR MACHINE (SVM)

Support Vector Machine, parfois abrégé en SVM ou SVC (Support Vector Classifier) est une technique bien connue de classification par apprentissage machine supervisé. L'objectif d'un algorithme SVM est de construire un hyperplan n-dimensionnel qui peut être utilisé pour la classification ou la régression. Idéalement, un bon hyperplan est celui qui atteint la plus grande distance par rapport au point de données le plus proche d'une classe particulière [11]. Ceci peut être mieux expliqué à l'aide de la figure 23.

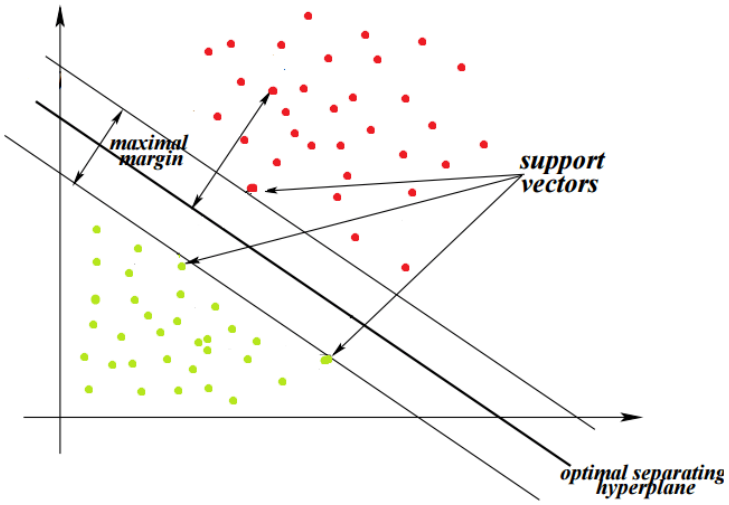


Figure 23 : L'hyperplan construit avec SVM

Ici, les points rouges et verts représentent deux classes différentes de l'ensemble de données. Les vecteurs de support sont les points qui aident à identifier l'hyperplan, et dans ce nuage de points, la classe verte fournit deux vecteurs de support et la classe rouge en fournit un. En utilisant ces vecteurs de support, nous traçons des lignes qui les traversent et qui permettent de distinguer les données de l'autre classe.

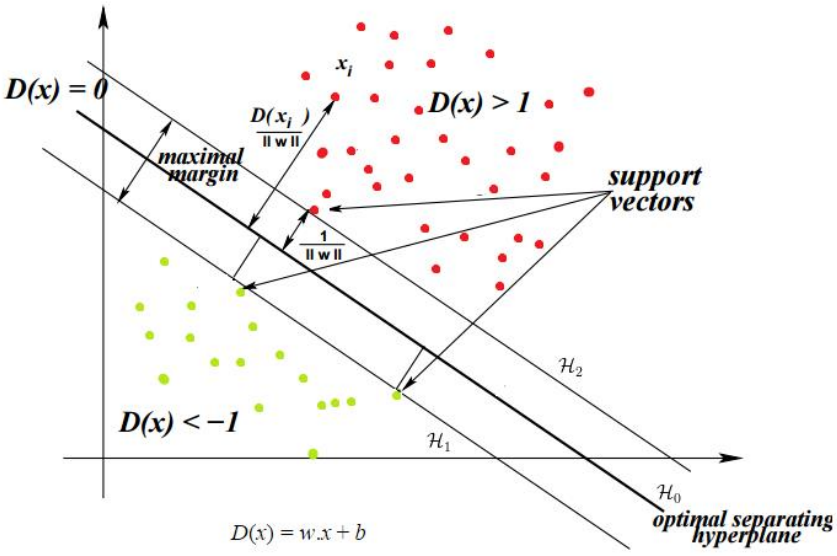


Figure 24 : Support Victor dans SVM

En faisant cela, on obtient les deux lignes w*x + b=1 et w*x + b=-1. En outre, ces deux classes sont séparables linéairement et le tracé de l'hyperplan est simple dans

ce cas. Un hyperplan est simplement une ligne droite qui divise les deux classes avec une distance maximale des points les plus proches de la ligne, également appelée vecteur de support. Cependant, la partie essentielle d'un SVM consiste à définir un hyperplan optimal qui maximiserait la largeur de la marge (w). Dans un scénario idéal, la largeur maximale est obtenue par la formule $\frac{2}{||w||}$.

Cependant, pour la plupart des scénarios, les données ne sont pas dispersées d'une manière qui serait linéairement séparable. De plus, une ligne pourrait ne pas être utilisée pour toujours définir l'hyperplan. Un plan peut être utilisé comme hyperplan dans les espaces de dimensions supérieures. Par ailleurs, les données ne sont pas toujours séparables de manière linéaire. Pour une distribution non linéaire des données, le SVM a un concept de "kernel trick" qui est utilisé pour définir l'hyperplan optimal. L'astuce du noyau convertit les données dans un espace de dimension supérieure pour obtenir une séparation linéaire entre les points. La figure 10 montre comment l'hyperplan passe d'une ligne (x,y) à un plan (x, y, xy) en fonction de l'espace de dimensionnalité.

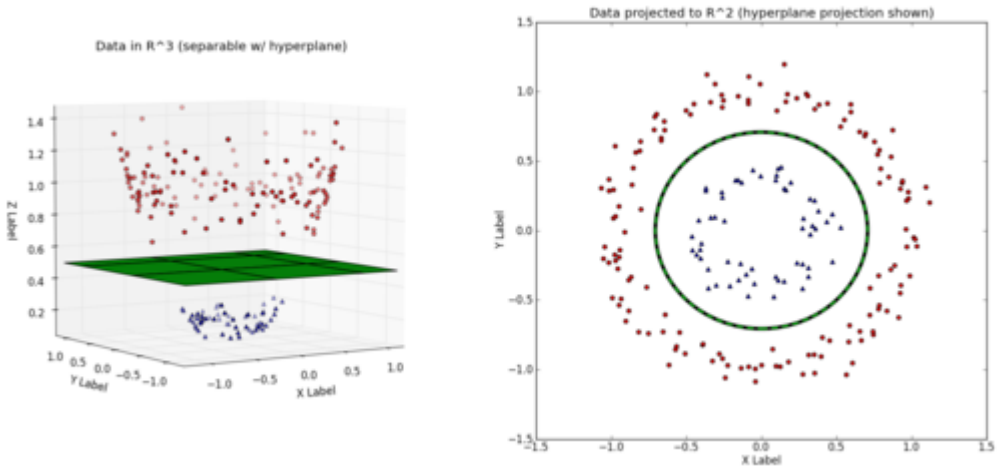


Figure 25 : Représentation de kernel trick.

La figure 10 représente une simple fonction de noyau qui utilise la formule x,y,x*y pour obtenir une séparation linéaire des données. Il s'agit d'une représentation simple de la façon dont les astuces du noyau fonctionnent. La fonction de base radiale gaussienne (RBF) est l'astuce du noyau la plus utilisée dans le SVM et elle est définie par l'équation suivante :

$$K(x, x') = e^{\left(\frac{\left||x-x'|\right|^2}{2\sigma^2}\right)}$$

Les SVM étaient principalement utilisés pour les problèmes de classification binaire, cependant, les récents développements dans les astuces du noyau les ont rendus efficaces pour les classifications multi classes également. Ils utilisent la technique "One vs All" (OvA) tout en effectuant la classification sur chaque classe lors d'une classification multi classe. Par exemple, supposons qu'il y ait trois classes - A, B et C - et qu'une entrée x soit à déterminer. Le SVM vérifiera d'abord x pour la classe A en tant que "A contre non-A" et calculera un score. De même, il fait de même pour "B contre non-B" et "C contre non-C" pour obtenir des scores multiples. Ces notes sont ensuite classées pour déterminer le classement de l'entrée x.

3.2 RANDOM FOREST (RF)

La classification aléatoire des forêts (RFC) est une autre technique de classification supervisée basée sur des arbres de décision utilisés dans l'apprentissage machine. Un algorithme d'arbre de décision construit un modèle arborescent de l'ensemble de données où chaque nœud est divisé en fonction de plusieurs critères personnalisables [12] . Random Forest est une collection d'arbres de décision non biaisés, non liés et non corrélés, et est donc appelée "forêt aléatoire". Avant de parler de forêts aléatoires, nous devons d'abord comprendre ce qu'est un arbre de décision et comment il fonctionne.

Un arbre de décision est une collection de nœuds connectés où chaque nœud est accessible depuis le nœud précédent en fonction de certains critères. Par exemple, si une personne souhaite aller pique-niquer mais ne peut pas déterminer si la journée lui convient, un arbre de décision peut ressembler à :

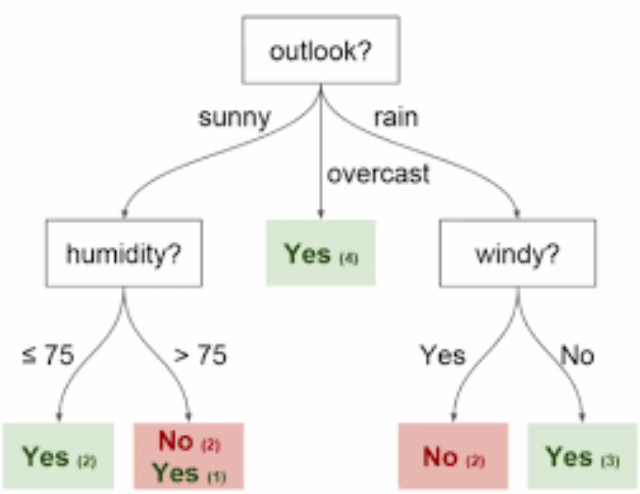


Figure 26 : Arbre de décisions.

Le premier nœud "Outlook" est divisé en trois conditions, à savoir "Sunny'", "Overcast" et "Rain", suivies par d'autres nœuds en dessous. Il s'agit d'une représentation très simple de l'arbre de décision. Une fois l'arbre de décision créé, il est assez simple pour l'algorithme de classer l'échantillon d'entrée fourni en parcourant simplement l'arbre. Toutefois, il n'existe pas d'arbre de décision parfait pour un ensemble de données. Il existe différentes méthodes permettant de moduler les fractionnements à chaque nœud et qui influent largement sur la formation de l'arbre de décision. Par exemple, l'arbre de décision serait très différent si le premier nœud était "Wind" au lieu de "Outlook". La performance de l'arbre de décision changerait également de manière significative en fonction de ce changement de conception.

Par conséquent, il est impératif de connaître les facteurs qui influent sur la détermination du nœud actuel et les critères de fractionnement.

Les deux concepts les plus importants qui sont largement utilisés pour cette détermination sont les valeurs de Gini et d'entropie. Gini est utilisé pour calculer l'impureté et Entropie est utilisé pour calculer le gain d'information du nœud. Ainsi, lors du calcul de l'impureté de Gini, le nœud qui fournit le moins d'impureté de Gini est sélectionné pour le fractionnement. La formule de calcul de l'impureté à partir de la valeur de Gini est définie comme 1 moins le carré de la probabilité de chaque classe. Symboliquement, elle peut être représentée par : $1 - \sum_{t=0}^{t=1} P_t^2$

Un fractionnement parfait aboutirait à un score de Gini de 0, ce qui serait toujours le cas aux nœuds des feuilles. De même, en utilisant l'entropie, l'objectif est de diviser au niveau du nœud qui fournit le gain d'information maximal.

Au nœud de la feuille de trois, le gain d'entropie est de 1, ce qui signifie un gain d'information complet et constitue donc une division parfaite.

L'entropie est calculée à l'aide de la formule : $E(s) = \sum_{i=1}^{c} -p \log_2 p_i$

Sur la base de cette valeur, nous calculons le gain d'information comme étant la différence entre les autres classes et l'entropie calculée. Le gain d'information pour chacune est ensuite comparé au fractionnement, à condition que le gain maximum soit sélectionné comme critère de fractionnement.

La principale différence entre Gini et Entropie est que Gini est généralement utilisé pour minimiser les erreurs de classification alors que l'Entropie est plutôt utilisée à des fins exploratoires. Par conséquent, le Gini est généralement considéré comme une bonne mesure pour les problèmes de classification. Une autre différence est que l'entropie est généralement plus lente à calculer en raison des opérations logarithmiques.

En outre, tous ces calculs sont effectués pour déterminer un seul arbre de décision. Comme nous l'avons vu précédemment, Random Forest est une collection d'arbres de décision sans lien entre eux, qui peuvent être créés en utilisant les valeurs de Gini ou d'Entropie. Sur la base de ces spécificités, l'algorithme construit une forêt de n arbres. En supposant qu'il y a 3 classes - A, B et C dans un ensemble de données et que la forêt contient 10 arbres. Supposons que trois arbres prédisent la classe A, cinq arbres prédisent la classe B, et les deux autres arbres prédisent la classe C. Dans ce cas, la prédiction globale de la forêt aléatoire serait la classe B, car c'est celle qui a le plus grand nombre de votes-cinq.

Cependant, ce problème peut aussi parfois conduire à un surdimensionnement car ces cinq arbres peuvent être biaisés en faveur de la classe B, ce qui entraîne la plupart des prédictions de la forêt en B.

Ainsi, l'augmentation du nombre d'arbres réduit considérablement le problème du surdimensionnement dans la classification de la Forêt Aléatoire. Ainsi, les forêts aléatoires (RF) sont considérées comme l'un des algorithmes de classification les mieux supervisés car ils utilisent des arbres de décision multiples qui non seulement fournissent une meilleure classification mais évitent également l'Overfitting.

3.3 K NEAREST NEIGHBORS (KNN)

L'algorithme de classification k Nearest Neighbors (kNN) [13] est l'un des algorithmes de classification par apprentissage machine les plus simples. Contrairement à la plupart des techniques, kNN est une technique paresseuse, ce qui signifie qu'il n'effectue les évaluations que lorsque cela est nécessaire. Cela rend cet algorithme rapide au coût d'une certaine précision. Lorsqu'une entrée est fournie, kNN calcule sa distance avec chaque vecteur de l'ensemble de données et détermine ensuite les k vecteurs ayant les plus petites distances. Les classes de ces k vecteurs sont ensuite comparées et la classe qui a été prédite le plus est retournée. Bien que cette approche semble simple, elle est efficace dans de nombreux algorithmes d'apprentissage machine.

Les deux décisions les plus importantes dans cet algorithme sont l'estimation de la valeur k et la métrique de la distance.

L'estimation de la valeur de k est généralement effectuée à l'aide de la méthode du coude qui tente de trouver cette valeur de k, après quoi la précision n'augmente pas beaucoup. L'estimation de la valeur de k par la méthode du coude donne donc une bonne valeur de k en fonction du temps et de la précision. En outre, les deux méthodes les plus utilisées.

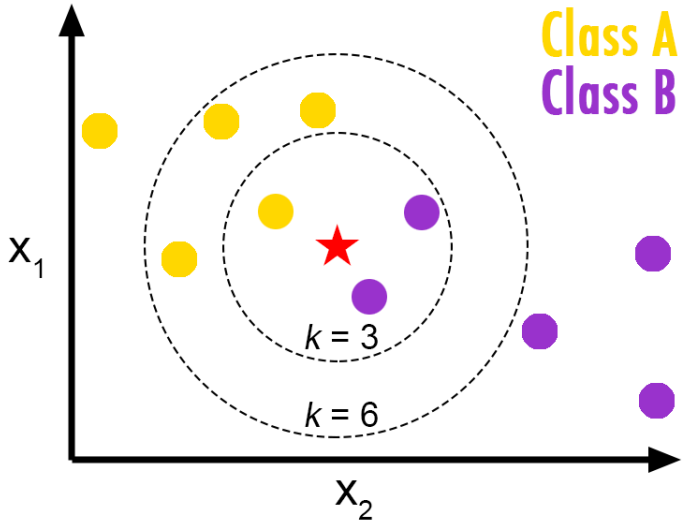


Figure 27 : Représentation de KNN

Les mesures de distance pour kNN sont la distance euclidienne et la similitude cosinus. Habituellement, avec des nombres continus tels que les coefficients MFCC, la distance euclidienne est une mesure plus efficace à choisir car elle permet de comparer la magnitude de la différence entre les vecteurs. La similarité cosinusoïdale permet également de calculer la différence de direction des vecteurs, ce qui n'est pas nécessaire pour la plupart des études.

Mathématiquement, la distance euclidienne est calculée comme suit :

$$E(x, y) = \sqrt{\sum_{i=0}^{n} (x_i - y_i)}$$

KNN un algorithme très simple mais au même temps il donne des résultats parfaits.

3.4 RESEAUX DE NEURONE ARTIFICIEL

Également connu sous le nom de réseau neuronal, il s'agit d'un modèle computationnel inspiré par la structure et les fonctions des neurones biologiques formant les éléments structurels du cerveau et considérés comme ses unités de traitement de l'information.

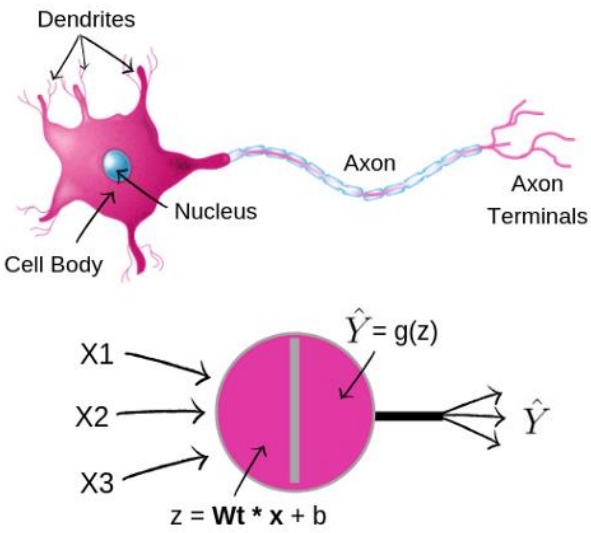


Figure 28 : Le neurone biologique et le neurone artificiel

Les réseaux neuronaux artificiels (Artifial Neural Network en anglais) [14], sont des modèles computationnels inspirés des systèmes nerveux. Ils ont la capacité d'acquérir et de maintenir des connaissances (basées sur l'information) et peuvent être définis comme un ensemble d'unités de traitement, représentées par des neurones artificiels, interconnectées par un grand nombre d'interconnexions (synapses artificielles), mises en œuvre par des vecteurs et des matrices de poids synaptiques. Les caractéristiques les plus pertinentes concernant les applications des neurones artificiels sont les suivantes :

- Adaptation à partir de l'expérience : Les paramètres internes du réseau, généralement ses poids synaptiques, sont ajustés par l'examen d'exemples successifs (modèles, échantillons ou mesures) liés au comportement du processus, permettant ainsi l'acquisition de connaissances par l'expérience.

- Capacité d'apprentissage : Grâce à l'utilisation d'une méthode d'apprentissage, le réseau peut extraire la relation existante entre les différentes variables de l'application.

- Capacité de généralisation : une fois le processus d'apprentissage terminé, le réseau peut généraliser les connaissances acquises, permettant ainsi d'estimer des solutions jusqu'alors inconnues.

- Organisation des données : sur la base des informations innées d'un processus particulier, le réseau peut organiser ces informations, permettant ainsi de regrouper des modèles présentant des caractéristiques communes.

- Tolérance aux pannes : Grâce au nombre élevé d'interconnexions entre les neurones artificiels, le réseau de neurones devient un système tolérant aux pannes si une partie de sa structure interne est corrompue à un certain degré.

- Stockage distribué : la connaissance du comportement d'un processus particulier apprise par un réseau de neurones est stockée dans chacune des synapses entre les neurones artificiels, ce qui améliore la robustesse de l'architecture en cas de perte de certains neurones.

- Prototypage facilité : en fonction des particularités de l'application, la plupart des architectures neurales peuvent être facilement prototypées sur du matériel ou des logiciels, puisque leurs résultats, après le processus de formation, sont généralement obtenus par quelques opérations mathématiques fondamentales.

Entrée : En moyenne, un neurone reçoit des entrées de 10^3 à 10^4 autres neurones. Par conséquent, l'entrée sera un vecteur de signaux x = [x 1, x 2, ..., x n], n désignant la longueur du vecteur.

Poids : les signaux seront multipliés par un certain poids pour représenter la force de la connexion synaptique.

Un neurone k reçoit en entrée les sorties d'un ensemble de n autres neurones, qui sont modifiés par un ensemble de poids w = [w 1k, w 2k, ..., w n k].

Activation : cette fonction peut être interprétée comme la probabilité que le neurone soit activé ou à l'état "on".

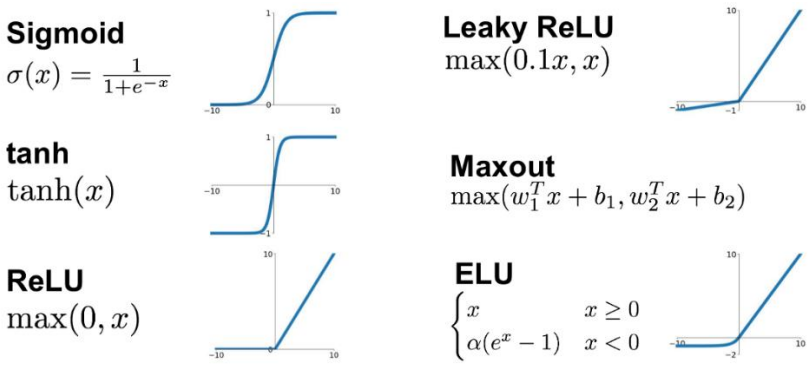


Figure 29 : Types des Fonctions d'activation.

En bref, un ANN est un outil d'analyse statistique non linéaire des données où les relations complexes entre les entrées et les sorties sont modélisées ou les modèles sont appris à partir des données d'observation, afin d'estimer les méthodes les plus rentables et les plus idéales pour arriver à des solutions.

3.4.1 PERCEPTRON MULTICOUCHE

Le perceptron multicouche *(multilayer perceptron* MLP) est un type de réseau neuronal_artificiel organisé en plusieurs couches au sein desquelles une information circule de la couche d'entrée vers la couche de sortie uniquement ; il s'agit donc d'un réseau à propagation directe (*feedforward*). Chaque couche est constituée d'un nombre variable de neurones, les neurones de la dernière couche (dite « de sortie ») étant les sorties du système global.

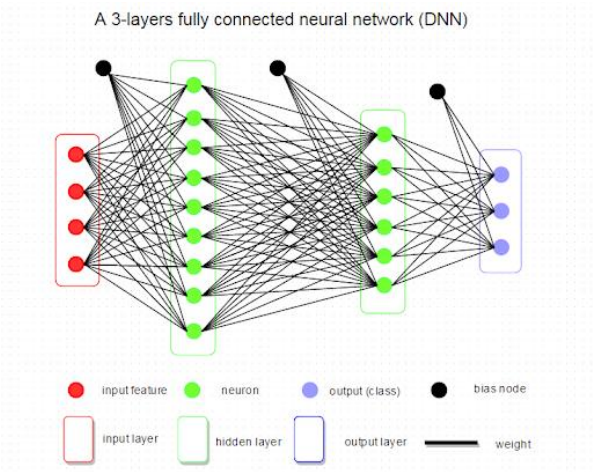


Figure 30 : Le perceptron multicouche

3.4.2 DEEP LEARNING

L'apprentissage profond (Deep Learning DL), une branche de l'apprentissage machine, est considéré comme une approche d'apprentissage de la représentation qui peut traiter directement et apprendre automatiquement des caractéristiques abstraites de niveau moyen et élevé acquises à partir de données brutes, en particulier des images. Il permet d'effectuer des tâches d'analyse automatique d'images, telles que la classification.

DL est une nouvelle branche de ML qui est basée sur un ensemble d'algorithmes pour modéliser des abstractions de haut niveau dans les données en extrayant des couches de traitement multiples, ce qui permet aux systèmes d'apprendre des fonctions de cartographie complexes directement à partir de données d'entrée $f : X \rightarrow Y$.

Sous-domaine de l'apprentissage de la représentation où de nombreuses couches d'étapes de traitement de l'information dans des architectures hiérarchisées et supervisées sont exploitées pour l'apprentissage non supervisé de caractéristiques et pour l'analyse/classification de modèles. Le principe de l'apprentissage approfondi est de calculer des caractéristiques ou des représentations hiérarchiques des données d'observation, où les caractéristiques ou facteurs de niveau supérieur sont définis à partir de ceux de niveau inférieur.

L'apprentissage approfondi consiste à apprendre plusieurs niveaux de représentation et d'abstraction qui permettent de donner un sens à des données telles que des images, des sons et du texte.

Comme le montrent les deux figures ci-dessous, l'apprentissage approfondi est un sous-domaine de l'apprentissage de la représentation, de l'apprentissage machine et de l'intelligence artificielle. C'est aussi l'intersection entre les domaines de recherche de la vision par ordinateur, du traitement de l'image et du traitement du signal, ainsi que de la modélisation graphique, de l'optimisation.

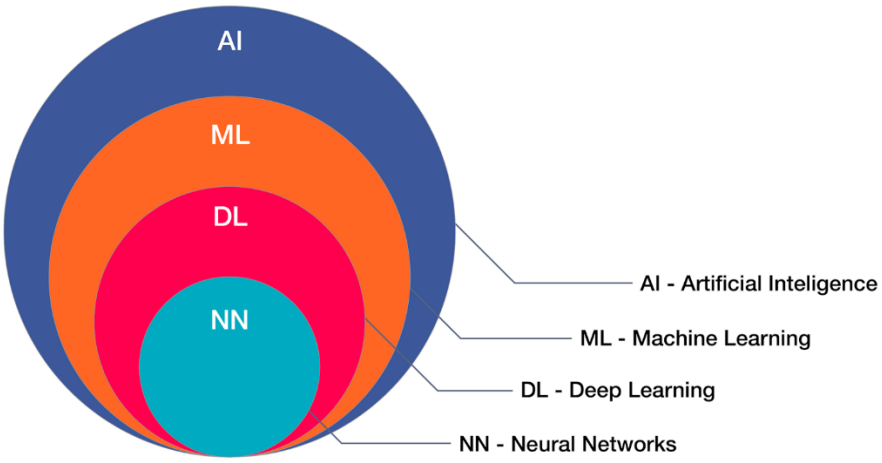


Figure 31 : les sous-domaines de IA

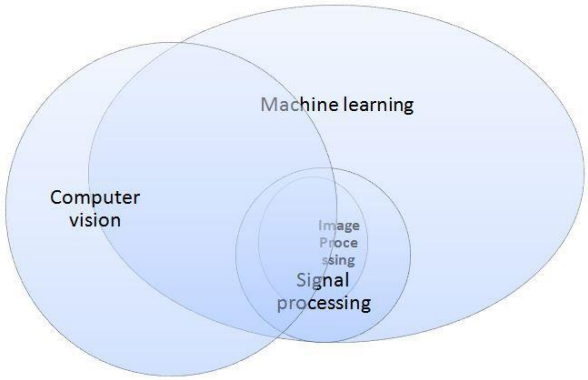


Figure 32 : Le traitement de signal dans le monde de l'informatique

Il existe plusieurs architectures d'apprentissage profond qui ont été largement étudiées ces dernières années, notamment Belief network ((DBN), Autoencoder, réseau neuronal profond convolutif (DCNN), réseau neuronal récurrent (RNN), Region Based Convolutional Neural Network (R-CNN), traitement du signal. Ils ont été appliqués avec succès dans divers domaines, tels que le traitement du langage naturel, la vision par ordinateur et le traitement du signal. Cependant, les tendances actuelles de la recherche ont démontré que les R-CNN sont très efficaces pour l'analyse automatique des images.

3.5 CONCLUSION

Pour ce travail, nous évaluerons certaines de ces techniques de classification par apprentissage machine telles que kNN, Support Vector Machines, Random Forest et Artificiel Neural Network.

4 ETAT D'ART

La reconnaissance automatique de la parole (ASR) peut être divisée en deux grandes parties : la reconnaissance de la parole et la reconnaissance du locuteur. Dans une perspective d'applications et de cas d'utilisation réels, il était logique pour les chercheurs de construire d'abord des systèmes qui comprennent la parole avant d'essayer d'identifier le locuteur. Les recherches sur la reconnaissance de la parole ont donc commencé près de dix ans avant la première étude connue sur la reconnaissance du locuteur. Davis et al. en 1952 aux Laboratoires Bell ont développé un système capable de reconnaître les chiffres prononcés par un locuteur [15]. Ils ont utilisé les fréquences des formants mesurées dans les voyelles des chiffres parlés pour la reconnaissance. Ce fut une découverte majeure dans le domaine de la RSA, car c'était la première tentative réussie de reconnaissance de la parole, mais elle se limitait aux chiffres. Les mots, les phrases ou même les nombres n'étaient pas détectables par cette approche. De plus, en raison du manque de ressources informatiques à cette époque, la détection était relativement lente et limitée.

Par la suite, un groupe de chercheurs a commencé à étudier la reconnaissance du locuteur, et en 1960, Pruzansky a lancé une recherche dans ce domaine aux Bell Labs [16]. Sa théorie consistait à corréler des spectrogrammes numériques pour une mesure de similarité afin de déterminer si le locuteur est bien la personne qu'il prétend être. Cette étude a donné le signal de départ de la recherche sur la reconnaissance du locuteur, et plusieurs méthodologies d'extraction de caractéristiques et de mesure de similarité ont été étudiées et proposées depuis lors.

Comme le désir d'atteindre des taux de précision plus élevés dans la reconnaissance du locuteur ne cessait de croître en même temps que les capacités matérielles et logicielles, les chercheurs ont commencé à utiliser plusieurs techniques informatiques dans le ASR. Outre la classification et/ou le regroupement de vecteurs de caractéristiques ou de signaux audio, une étape tout aussi cruciale consiste à générer ces vecteurs de caractéristiques à partir des fichiers audios. Ainsi, une recherche active

est menée dans l'extraction de caractéristiques pour la ASR, ce qui permet une reconnaissance et une identification efficaces.

.

4.1 SVM POUR LA RECONNAISSANCE DE LOCUTEUR

En 2012, Bao et al. ont poursuivi leurs recherches en utilisant le SVM pour la reconnaissance du locuteur, parallèlement au modèle de mélange gaussien (GMM) [17]. L'un de leurs principaux objectifs était d'aborder la question de la réduction des précisions avec plus de données vocales dans le SVM. Le GMM était un modèle couramment utilisé dans le domaine de la reconnaissance du locuteur, mais il était peu performant lorsque les données de formation ou de test n'étaient pas suffisantes. Le SVM, en revanche, peut résoudre le problème. Mais, comme nous l'avons vu plus haut, même le SVM a souvent des performances limitées avec une grande quantité de données. Bao et al. ont essayé d'utiliser les avantages du SVM et du GMM pour construire leur système de reconnaissance du locuteur. Ils ont formé et construit le GMM en utilisant l'algorithme de maximisation des espérances (EM) sur un large ensemble de données. Le modèle qui en résulte est censé avoir une représentation suffisamment précise de la distribution de la voix sous une forme compressée qui convient au SVM. Les expériences ont été réalisées sur un ensemble de données composé de 23 personnes, la durée des échantillons audio étant de 15 secondes et le GMM étant de l'ordre de 32. Pour des durées de formation comprises entre 5 et 18 secondes, ils ont obtenu des taux de précision compris entre 73,2 et 96,7%.

Les travaux les plus récents concernant la reconnaissance du locuteur à l'aide de machines à vecteurs de support (SVM) ont été réalisés par Chakroun et al. en 2015 [18]. La plupart des techniques de reconnaissance du locuteur utilisant les SVM ont été étudiées en utilisant le noyau linéaire. Bien qu'il s'agisse d'une approche relativement bonne, les données non structurées comme l'audio ont plus de chances d'être mieux comprises en utilisant une astuce du noyau SVM. Pour leurs recherches, ils se sont donc concentrés sur d'autres noyaux complexes au-delà du noyau linéaire. Cependant, avant de commencer leur phase de classification, ils ont d'abord exploré différentes techniques d'extraction de caractéristiques pour la reconnaissance du locuteur et ont conclu que les MFCC ainsi que leurs dérivés du second ordre (coefficients delta-

delta18) étaient idéaux pour extraire des caractéristiques d'enregistrements audio contraints. Par conséquent, en s'appuyant sur les coefficients MFCC + delta-delta, ils ont conçu un modèle SVM pour effectuer la tâche de classification. En outre, comme leur recherche visait à étudier l'efficacité d'un noyau SVM non linéaire dans la reconnaissance du locuteur, ils ont d'abord appliqué un SVM linéaire à leur ensemble de données pour obtenir la ligne de base. Plus tard, ils ont utilisé le noyau RBF (Radial Basis Function) du SVM pour les classifications. Bien qu'ils aient utilisé l'ensemble de données TIMIT, ils n'ont pas utilisé la totalité de l'ensemble de données pour leur recherche, mais un sous-ensemble de celui-ci. Leur ensemble de données n'est pas public, et nous ne connaissons donc pas la répartition des genres, des langues et des autres composantes connexes de leurs audios. Cependant, leur approche de l'utilisation du noyau RBF a montré de grandes améliorations car ils ont pu réduire le taux d'erreur égal de presque la moitié par rapport au noyau linéaire. Cette étude a donc constitué une avancée majeure en termes d'utilisation des SVM pour la reconnaissance du locuteur et la plupart des activités de reconnaissance du locuteur sont actuellement réalisées à l'aide d'un noyau SVM non linéaire tel que RBF ou Polynomial.

4.2 RESEAU DE NEURONE ARTIFICIEL POUR LA RECONNAISSANCE DE LOCUTEUR

Le Deep Learning a connu une forte croissance ces dernières années, et Tirumala et al. ont étudié en 2016 [19] les applications du Deep Learning dans la reconnaissance du locuteur. Ils ont estimé que l'apprentissage approfondi est largement utilisé dans plusieurs domaines tels que le traitement du langage naturel, la reconnaissance d'images et la vision par ordinateur, mais que ses applications sont limitées dans les domaines de la reconnaissance du locuteur en raison du manque de connaissances.

Avec cette recherche, ils ont essayé de réduire le manque de connaissances entre l'apprentissage approfondi et le groupe de chercheurs utilisant les approches traditionnelles pour la reconnaissance du locuteur.

Les réseaux neuronaux artificiels (ANN) sont efficaces pour ce type de recherche, cependant, comme le mécanisme de formation dépend de plusieurs couches, il peut souvent prendre beaucoup de temps. Afin de résoudre ce problème, Tirumala et al. ont utilisé l'apprentissage profond, une couche gourmande en entraînement et ont réduit le temps de formation.

Pour cette étude, Tirumala et al. ont envisagé une simple classification des techniques d'identification des locuteurs. Le premier niveau est divisé en deux parties - basé sur les empreintes vocales et basé sur un nouveau locuteur sans empreintes vocales dans la base de données, elles sont ensuite divisées en ensembles fermés/ouverts et en approches dépendantes/indépendantes du texte.

Outre la classification, l'un des principaux domaines de recherche en matière de reconnaissance du locuteur a toujours été la phase d'extraction des caractéristiques. C'est la phase où l'inscription d'un locuteur a lieu et où une copie de toutes les données relatives à l'empreinte vocale du locuteur est stockée. Comme cette phase peut souvent être limitée, la phase d'extraction de caractéristiques devient cruciale dans le modèle global de reconnaissance du locuteur. Tirumala et al. ont beaucoup insisté sur la manière dont l'apprentissage approfondi peut être utilisé pour surmonter les difficultés rencontrées dans cette phase. Ils ont conçu une topologie de réseau neuronal profond (DNN) où chaque niveau fonctionne au niveau acoustique. Le son d'apprentissage est

récupéré image par image et est transmis au DNN. Ainsi, la sortie de la couche précédente est utilisée comme représentation du locuteur particulier, appelée vecteur d.

Ces vecteurs d sont ensuite utilisés dans les classificateurs DNN pour l'identification du locuteur. Tirumala et al. [19] ont proposé une architecture consistant à utiliser deux DNN avec des vecteurs d et plusieurs couches cachées. Cette recherche ne présente pas de résultats concrets permettant de comparer leur approche avec les approches traditionnelles, mais elle sert de premier pas vers l'utilisation des réseaux neuronaux pour l'extraction et la classification des caractéristiques dans l'identification du locuteur.

En s'appuyant sur les techniques d'apprentissage approfondi discutées ci-dessus pour la reconnaissance du locuteur, Ge et al. ont poursuivi leurs recherches sur la manière dont un réseau neuronal à action directe pourrait être efficacement utilisé pour la reconnaissance du locuteur indépendante du texte [20]. Ils se sont concentrés sur l'utilisation d'une technique unique de détection active de la voix pour l'extraction de caractéristiques au lieu des techniques traditionnelles, car cela permet d'éliminer les divergences qui peuvent être présentes dans différents échantillons audios pour le même locuteur.

En outre, ils ont utilisé le MFCC sur la parole prétraitée pour la normalisation et la concaténation. 20 Ils ont normalisé les caractéristiques en utilisant la moyenne et la variance du locuteur avec lui-même au lieu d'utiliser les valeurs obtenues sur l'ensemble des données. C'est ce que l'on appelle le niveau du haut-parleur MVN. Pour la classification, ils ont utilisé les caractéristiques comprenant près de 400 vecteurs dimensionnels comme entrées du réseau neuronal et ont utilisé la célèbre approche d'Andrew Ng consistant à considérer une classification multi-classes comme n classifications binaires différentes, où n est le nombre de classes [21]. Ils ont mené leurs expériences sur l'ensemble de données TIMIT en utilisant les 200 premiers locuteurs masculins pour la phase de formation et ont utilisé des seuils et des tracés ROC pour la vérification. Avec cette approche, ils ont obtenu des taux de précision décents et un taux d'erreur égal de moins de 6%, ce qui est impressionnant. La recherche marque un progrès important dans l'utilisation des réseaux neuronaux pour la reconnaissance des locuteurs, mais elle n'est pas évaluée sur les sons de l'environnement naturel réel du travail. En outre, le processus de formation est beaucoup plus lent car il nécessite une

plus grande puissance de traitement et entraîne le modèle individuellement au lieu de considérer un groupe de locuteurs dans son ensemble.

5 CONCLUSION

Nous avons présenté dans ce chapitre différents approches et techniques soit pour le pré-traitement, extraction des caractéristiques, soit pour la classification par des algorithmes de machine Learning, on a aussi vu quelques travaux dans ce secteur et qui sont relie à notre projet afin de comparer et amélioré la précision de ces algorithmes. Ainsi on va implémenter les techniques les plus performants sur notre projet afin d'avoir la plus grande précision.

CHAPITRE 3 : ENVIRONNEMENT ET MISE EN PLACE DE LA SOLUTION

1 OUTILS ET BIBLIOTHEQUES

Nous allons passer en revue les différents choix techniques et outils que nous avons utilisés pour ce projet, ainsi que les IDE

1.1 OUTILS ET BIBLIOTHEQUES

PYTHON

Python a été développé pour la première fois par Guido van Rossum à la fin des années 1980. Aujourd'hui, il est devenu l'un des langages de programmation les plus populaires grâce à sa syntaxe claire et à sa lisibilité.

Python est un langage de programmation orienté objet, multiparadigme, polyvalent, interprété, de haut niveau, particulièrement adapté aux projets d'apprentissage machine et d'apprentissage profond.

LIBROSA

LibROSA est une bibliothèque Python conçue pour l'analyse sonore [29]. Elle est considérée comme l'une des bibliothèques les plus polyvalentes pour le traitement audio en Python. Elle fournit une multitude de fonctionnalités de traitement du son utiles comme la visualisation de formes d'onde, les extractions de caractéristiques, les opérations mathématiques, et quelques autres. Elle possède également des widgets sonores IPython intégrés qui facilitent le débogage de la phase de traitement du son.

NUMPY

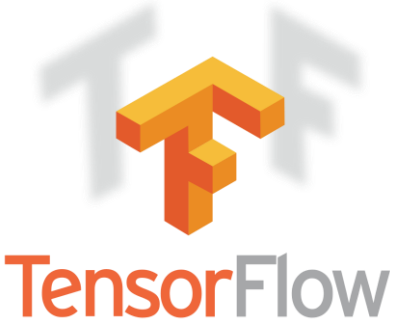
Numpy Fournit un support pour les représentations de grands tableaux multidimensionnels. En utilisant ses fonctions mathématiques intégrées de haut niveau, nous pouvons effectuer efficacement une analyse numérique sur une image et exprimer ces images sous forme de tableaux multidimensionnels.

SCIKIT-LEARN

Scikit-learn (également connu sous le nom de sklearn) est une bibliothèque efficace et bien connue construite pour l'apprentissage automatique en Python. Elle fournit des méthodes pour plusieurs tâches d'apprentissage machine et d'exploration de données telles que la régression, la classification, le regroupement, la modélisation et plusieurs autres activités d'apprentissage machine. Nous avons utilisé sklearn pour plusieurs tâches telles que la normalisation, la mise à l'échelle des données, la construction de modèles de classification, la génération de validations croisées, l'évaluation des performances du modèle et la visualisation des résultats.

TENSORFLOW

TensorFlow est une plate-forme de bout en bout pour l'apprentissage machine qui fonctionne à grande échelle et dans des environnements hétérogènes et prend en charge une variété d'applications, en mettant l'accent sur la formation et l'inférence sur les réseaux neuronaux profonds.



Le système de deuxième génération de Google Brain dispose d'un écosystème complet et flexible d'outils, de bibliothèques et de ressources communautaires permettant aux chercheurs de faire passer les nouvelles idées du concept au code et aux modèles de pointe à une publication plus rapide.

KERAS

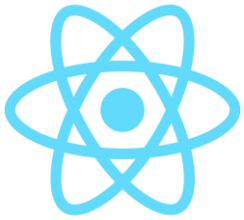
Keras est une API de réseaux neuronaux de haut niveau, écrite en Python et capable de fonctionner sur TensorFlow, CNTK ou Theano. Elle ne nécessite pas de fichiers de configuration des modèles au format déclaratif, puisque les modèles sont décrits dans un code python compact et plus facile à déboguer, ce qui permet une extensibilité aisée.

Keras a été développé dans le but de permettre une expérimentation rapide, en étant capable de passer de l'idée au résultat avec le moins de retard possible est la clé pour faire une bonne recherche en itérant rapidement à travers différentes architectures, et il peut fonctionner de manière transparente sur le CPU et le GPU.

STREAMLIT

Streamlit est une bibliothèque Python open-source qui permet de créer facilement de magnifiques applications web personnalisées pour l'apprentissage machine et la science des données.

REACT JS



React Une bibliothèque JavaScript pour la construction d'interfaces utilisateur. React rend la création d'interfaces utilisateur interactives facile. Concevez des vues simples pour chaque état de votre application, et React mettra à jour et rendra efficacement les bons composants lorsque vos données changent.

Autres bibliothèques Python

En dehors des bibliothèques mentionnées ci-dessus, quelques autres bibliothèques Python provenant de tiers et utilisées pour l'implémentation le sont : Scipy, Pandas, Matplotlib, Seaborn et PyAudio.

1.2 LES IDE

JUPYTER NOTEBOOK

JupyterLab est un environnement de développement interactif basé sur le web pour les cahiers, le code et les données Jupyter. JupyterLab est flexible : il permet de configurer et d'organiser l'interface utilisateur de manière à prendre en charge un large éventail de flux de travail dans le domaine des sciences des données, de l'informatique

scientifique et de l'apprentissage machine. JupyterLab est extensible et modulaire : écrivez des plugins qui ajoutent de nouveaux composants et s'intègrent aux composants existants.

Google Colab

Google Colab ou Colaboratory est un service cloud, offert par Google (gratuit), basé sur Jupyter Notebook et destiné à la formation et à la recherche dans l'apprentissage automatique. Cette plateforme permet d'entraîner des modèles de Machine Learning directement dans le cloud. Sans donc avoir besoin d'installer quoi que ce soit sur notre ordinateur à l'exception d'un navigateur.

VS Code

Visual Studio Code est un éditeur de code redéfini et optimisé pour la création et le débogage d'applications web et cloud modernes. Visual Studio Code est gratuit et disponible sur votre plateforme préférée.

2 DATASET

Le plus important factor dans un projet de machine Learning est la Dataset, car sans les données on ne peut pas construire un modèle qui peut identifier les patterns et les relations entre ces données, alors pour cela et avant de commencer n'importe qu'il projet de machine Learning il faut vérifier d'abord est ce qu'il existe ou est-ce que peut-on construire une Dataset pour ce problème.

Pour les problèmes de la reconnaissance vocale heureusement il existe pas mal de Dataset dans ce secteur-là, soit pour la reconnaissance vocale, ou soit pour la reconnaissance de locuteur, et pour ce dernier on peut citer VoxCeleb Dataset, Timit, et Speaker Recognition Dataset.

Concernant la Dataset utiliser dans ce projet, on a pu construire notre propre Dataset afin de tester notre voix dans la partie de test et voir l'impact de notre modèle vis-à-vis aux différentes langues.

2.1 POURQUOI CONSTRUISONS-NOUS UN TEL COMPOSANT

La darija marocaine est de plus en plus utilisée par les internautes marocains. Enregistrer des audio dans les réseaux sociaux en dialecte marocaine (Darija) est aujourd'hui une tendance et deviendra bientôt la règle. De nombreuses pages et groupes sur Facebook utilisent de plus en plus le darija comme langue de communiquer par des enregistrements audios dans leurs conversations et publications. Enfin, presque toutes les conversations WhatsApp sont des audio en darija en plus des larges quantités des vidéos uploader sur le YouTube chaque jour.

Par conséquent, il devient évident que la construction d'un algorithme de Machine Learning avec cette Dataset (Darija) efficace pour cibler la communité marocaine et pour aider les banques marocaines par exemple à reconnaitre (ou vérifier) leurs clientèles d'après leur voix comme une biométrie vocale.

Pour de nombreuses langues, des bases de données des audios ont déjà été créées et sont disponibles gratuitement sur Internet. Certains d'entre eux contiennent des

milliards de fichiers audio. De plus, une base de données existe déjà pour l'arabe classique et il n'en existe pas encore pour la darija marocaine.

2.2 UNE NOTE SUR LE DEFI DARIJA

L'intérêt croissant pour l'utilisation de Darija est un défi. Il n'existe en effet pas une seule Darija, mais on a plusieurs dialectes aussi en Darija (Chamalia, Hassanya, Felalia, Darija d'accent Amazigh etc..), et il existe plusieurs manières pour l'en parler.

Heureusement que notre approche est de Texte-Independent approche, ainsi on ne va pas s'intéressé au dialecte existant dans la Darija.

2.3 LA COLLECTION DE DARIJA DATASET

Durant trois semaines on a essayé de collecter une Dataset de dialecte Marocaine (Darija), soit depuis le YouTube, Instagrame ou par des audio WhatsApp envoyer par des amies.

Alors pour la collection dans la plupart de temps il était manuel car il faut vérifier par exemple pour chaque audio collecter depuis le YouTube est ce qu'il n'y a pas des vidéos qui sont monté par une musique en background, et aussi voir est ce que c'est juste une personne qui parle dans une telle vidéo ou plusieurs, des fois mêmes-ci il y'a des audio mixé on prend juste la voix d'une personne et on ignore les autres voix.

2.4 COMMENT ON A CONSTRUIT CETTE DATASET

Pour la construction de la Dataset on a essayé d'avoir des fichiers audios de la même longueur, ainsi que d'avoir le même nombre des fichiers audio pour chaque personne, et aussi équilibré entre les genres c'est-à-dire d'avoir les mêmes nombres des locuteur hommes et femmes.

Les fichiers WAV sont considérés comme plus utiles car ils couvrent toute la gamme des fréquences audibles par l'oreille humaine. Ainsi que l'extraction des caractéristiques de ces fichiers WAV est extrêmement cruciale. Cette phase constitue essentiellement la base des algorithmes d'apprentissage machine à utiliser pour la classification. Ainsi, les fichiers WAV sont largement préférés pour les études audios.

Le taux d'échantillonnage (Sample rate) est défini comme le nombre absolu d'échantillons audio présents en une seconde. La cohérence du taux d'échantillonnage est très importante dans une étude audio pour garantir que les coefficients extraits représentent les mêmes calculs sous-jacents.

Alors pour cela et à l'aide des scripts en python on a divisé chaque audio en morceau de durée de 30s avec un taux d'échantillonnage de 16000Hz, une Profondeur de bit (Bit Depth) de 16 bits et avec l'extension WAV.

2.5 DIVERSITE DE DATASET

Pour les besoins de cette étude, nous avons sélectionné 75 locuteurs sur lesquels évaluer nos modèles. Les principaux facteurs à l'origine de la sélection de ces 75 locuteurs sont les suivants :

- ✓ Avoir un nombre égal de locuteurs masculins et féminins : Cela permet d'éviter de trop adapter les techniques de formation et de test à un sexe particulier.
- ✓ Avoir des accents différents : Nous avons sélectionné des personnes de différentes régions de Maroc pour obtenir une variation des accents qu'elles parlent.
- ✓ Pour avoir plusieurs dialecte Marocaine : Comme la recherche est basée sur une approche indépendante du texte, nous pouvons non seulement avoir des personnes ayant des accents différents, mais aussi des personnes parlant des langues tout à fait différentes. Cela permet de simuler un scénario plus réel, car le modèle ne sera pas évalué sur une seule langue spécifique.

En gardant ces facteurs à l'esprit, nous avons obtenu plus de 4215 fichiers audios pour entraîner le modèle. Chaque fichier audio dure 30 secondes, avec environ 50 audios par locuteur. Cependant, lorsqu'un système de reconnaissance des locuteurs est développé dans la vie réelle, le cas peut ne pas être aussi parfait. Il est courant d'avoir plus d'échantillons d'audios pour un locuteur et moins d'échantillons pour un autre. C'est pourquoi, pour simuler ce déséquilibre extrêmement courant dans la vie réelle, nous

avons veillé à disposer d'un nombre légèrement variable d'échantillons audio pour chaque personne afin de lui permettre de s'entraîner sur le modèle. Cela nous permet non seulement de tester la flexibilité des modèles, mais aussi de savoir si le taux positif réel pour un locuteur change avec le nombre d'échantillons et, si oui, dans quelle mesure.

Vous trouverez ci-dessous un bref aperçu de la diversité de l'ensemble des données :

- Genres : 49 % d'hommes, 50 % de femmes
- Dialectes Marocaines : Chamalya, Casawia, Marrakchia, Sahrawia…
- Durée de chaque fichier : 30s
- Nombre total des fichiers : 4215
- Nombre totale d'heures : > 35 h

La figure 33 indique le nombre de fichiers audio utilisés par locuteur pour l'étude :

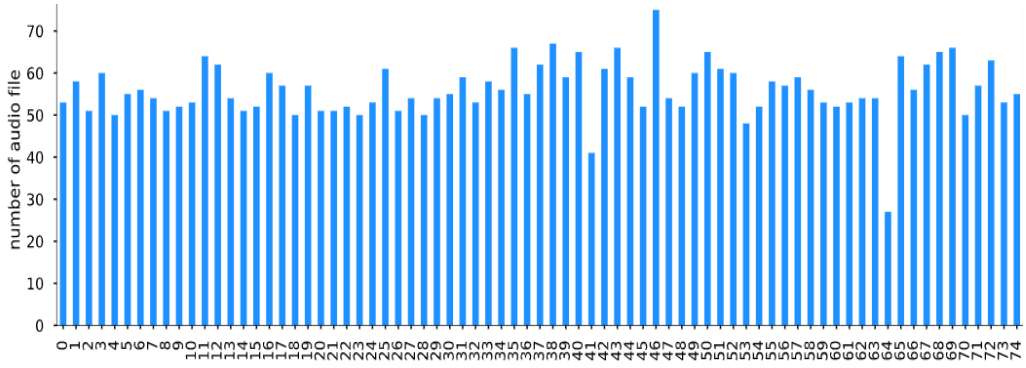


Figure 33 : Le nombre des fichiers audios pour chaque locuteur

La figure suivante montre la diversité de genre dans notre Dataset :

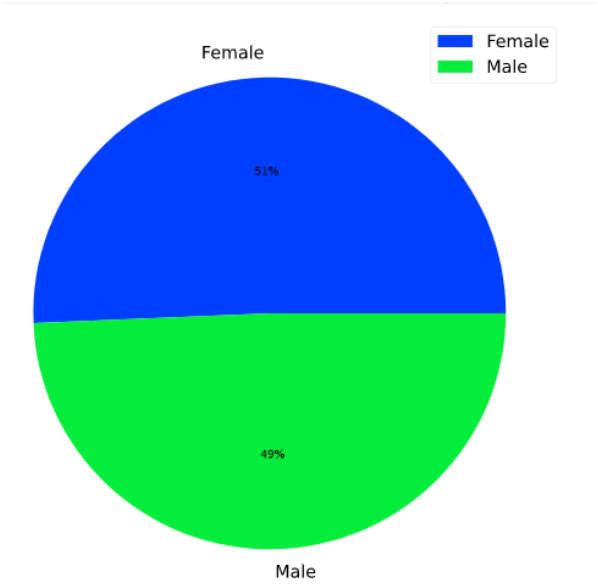


Figure 34 : La diversité de genre dans notre Dataset

On peut voir la structure de cette Dataset sur les fichiers dans notre machine sur la figure suivante :

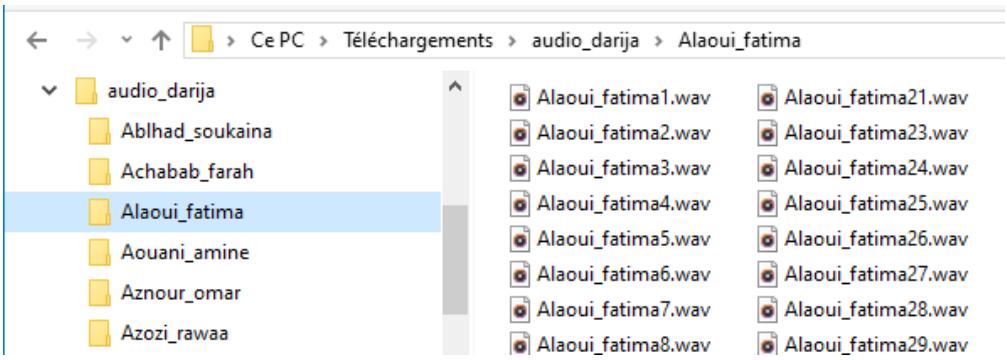


Figure 35 : La structure de Dataset dans nos fichiers

Le tableau ci-dessous donne toutes les informations sur chaque locuteur mais ici on a affiché juste les dix premiers locuteurs :

	Depuis YouTube	Nom de Locuteur	Genre	Nombre d'Audio	Longueur en (min)
1	OUI	Ablhad Soukaina	Female	53	26.5
2	NON	Achabab Farah	Female	58	29

3	OUI	Alaoui Fatima	Female	51	25.5
4	NON	Aouani Amine	Male	60	30
5	NON	Aznour Omar	Male	50	25
6	OUI	Azozi Rawaa	Male	55	27.5
7	OUI	Badir Aziz	Male	56	28
8	OUI	Bamossa Wafae	Female	54	27
9	OUI	Bamou kawtar	Female	51	25.5
10	OUI	Beldi Houyame	Female	52	26

Tableau 2: Des détails sur la Dataset

2.6 CONCLUSION

L'ensemble de données contient également plusieurs types de bruit de fond dans les 4215 fichiers audio collectés. Le bruit ambiant est le plus courant de tous, mais les fichiers audios contiennent également des bruits de véhicules à moteur, des discours d'autres personnes, de la musique, des applaudissements, des coups de klaxon. Ceci constitue une étude intéressante puisque l'algorithme d'élimination du bruit ambiant que nous avons mis en œuvre sera évalué sur différents types de bruits de fond.

3 IMPLEMENTATION

Dans ce chapitre et après l'aperçu qu'on a cité et détailler que ce soit sur les techniques de pré-traitement, d'extraction des caractéristiques ou celles de la classification, on va voir dans la suite les techniques qu'on a implémenté et comment on les a implémentés.

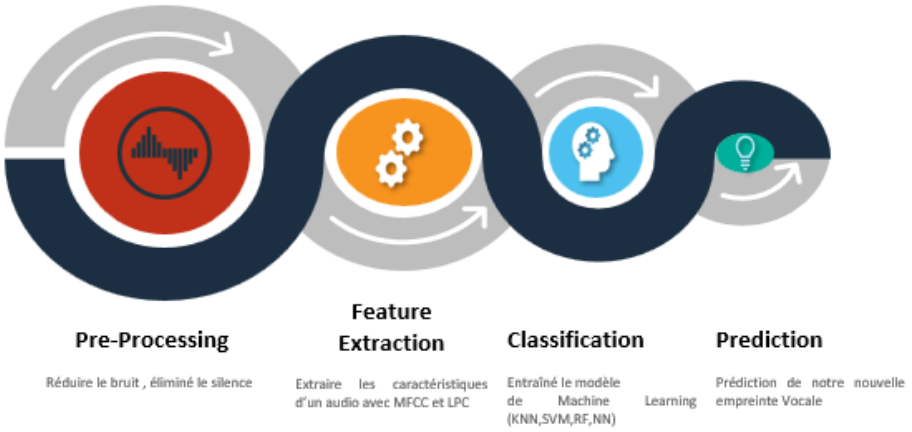


Figure 36 : Flux de travail de projet

3.1 PRE-TRAITEMENT

Pour l'implémentation, nous avons utilisé l'un des fichiers de notre dataset avec l'extension wav, PCM 16 bits, appelé "zennou_mohammad.wav", qui a une fréquence d'échantillonnage de 16000 Hz. Le fichier wav est un signal de parole propre comprenant une seule voix qui prononce quelques phrases avec une durée de 30s.

3.1.1 LA PREACCENTUATION

La préaccentuation augmente la quantité d'énergie dans les hautes fréquences. Pour les segments vocaux comme les voyelles, il y a plus d'énergie dans les basses fréquences que dans les hautes fréquences, ici on a utilisé un coefficient égal à 0.97

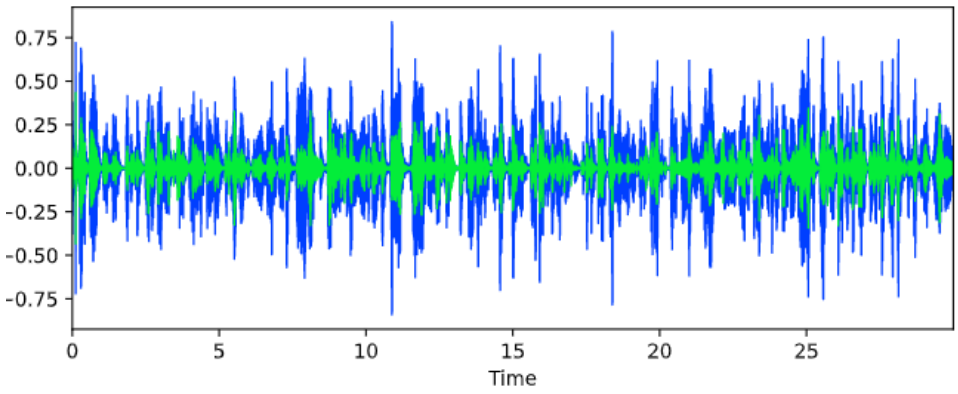


Figure 37 : Le signal d'origine et signal après La préaccentuation

3.1.2 DETECTION D'ACTIVITE VOCALE

Afin d'éliminer le silence et réduire le bruit ont implémenté quelque technique de VAD dans le domaine temporel, et vous pouvez voir sur le schéma suivant la méthode qu'on poursuite :

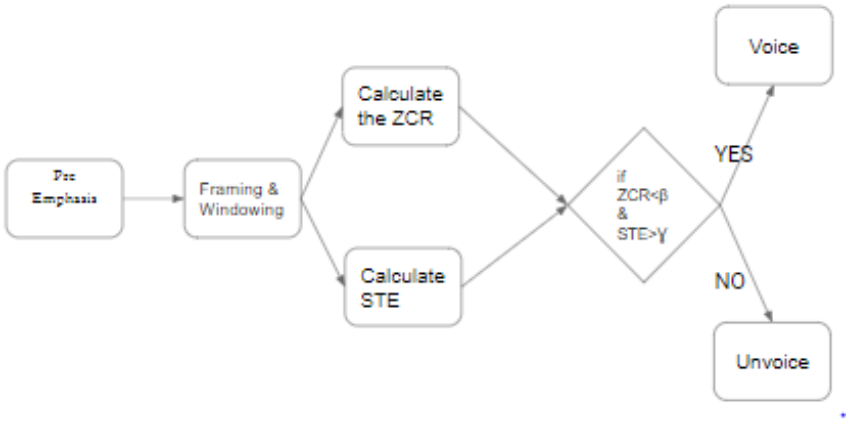


Figure 38 : Le processus de Voice Activity Détection

Pour une première étape nous avons segmenté les audios en petit segment de 25ms avec un chevauchement de 10ms (Fenêtrage), en suite nous avons appliqué le Hamming Windows afin que l'amplitude diminue auprès des bords (La fenêtre de Hamming présente une légère baisse soudaine au bord alors que la fenêtre de Hanning

n'en présente pas). Car La chute soudaine de l'amplitude va créer beaucoup de bruit qui se manifeste dans les hautes fréquences. Pour découper le signal audio, l'amplitude doit diminuer progressivement près du bord d'une fenêtre.

Le nombre de passages à zéro (ZCR) est un indicateur de la fréquence à laquelle l'énergie est concentrée dans le spectre du signal. La voix est produite par l'excitation des voies vocales par le flux d'air périodique au niveau de la glotte et présente généralement un faible taux de passage par zéro, tandis que la voix non prononcée est produite par la constriction des voies vocales suffisamment étroite pour provoquer un flux d'air turbulent qui produit du bruit et présente un taux de passage par zéro élevé.

L'énergie d'une parole est un autre paramètre de classification des parties vocales/non vocales. La partie vocale du discours a une énergie élevée en raison de sa périodicité et la partie non vocale du discours a une énergie faible.

Afin de détecter les parties vocales dans notre signal, on a utilisé les techniques de VAD expliquer au-dessus, et à l'aide des seuils définies on observons les signaux, de la plupart des régions de notre Dataset, on a conclu que avec un seuil $\beta = 0.10$ pour ZCR on peut classifier les régions qui comprenne de la voix de la non voix, c'est-à-dire c'est notre ZCR $< \beta$ on a de la voix, sinon il y a que de bruit ou silence, la même chose pour STE avec un seuil $\gamma = 0.005$ on peut classifier nos régions qui contient de la voix, des autres qu'il ont pas, si on a un STE $> \gamma$ signifier que on a de la voix, sinon on la pas.

3.2 EXTRACTION DES CARACTERISTIQUES

MFCC

La plupart des études de classification audio réalisées à ce jour portent sur des fichiers audios enregistrés dans des environnements de contraintes et suggèrent que les MFCC avec deltas sont les caractéristiques les plus appropriées. Comme nous l'avons vu en détail précédemment, les MFCC simulent étroitement le système auditif humain et sont donc bien adaptés à une telle étude. Cependant, l'utilisation des coefficients delta MFCC est en augmentation et il est important de rechercher si le calcul de la valeur

delta affectera pratiquement un ensemble de données audio enregistré dans une configuration beaucoup plus réaliste.

Comme nous l'avons vu précédemment, le calcul du MFCC est basé sur la longueur de la fenêtre, la longueur du saut et le nombre de coefficients. Pour nos expériences, nous avons fixé la longueur de la fenêtre (frame size) à 25 ms, la longueur du saut (Overlap) à 10 ms et le nombre de coefficients à 20. Ceci est dû au fait que notre audio est échantillonné à 16KHz, et la règle empirique pour le nombre de coefficients dit que les coefficients jusqu'à $\frac{sr+\frac{sr}{2}}{2}$ sont utiles, où sr est la fréquence d'échantillonnage (sample rate) de l'audio en KHz. Nous calculons donc le nombre de coefficients n_mfcc comme :

$$n\_mfcc = \frac{sr + \frac{sr}{2}}{2} = \frac{3sr}{4}$$

Ainsi, pour une fréquence d'échantillonnage de 16KHz, n = (3 * 16) / 4 = 12 coefficients au moins sont utiles.

Comme la première valeur d'après l'application de la DCT pour le calcul du MFCC donne la somme des log - énergies, nous devons prendre les 15 premiers coefficients du MFCC pour obtenir les 14 coefficients. Ainsi, bien que nous obtenions 12 selon la règle empirique, nous prenons 1 coefficient de plus au début pour obtenir tous les coefficients valeurs requises. Pour notre recherche, nous avons utilisé la bibliothèque LibROSA de Python pour atteindre ces coefficients. La méthode lib.feature.mfcc() accepte des arguments tels que les données de séries temporelles audio, le taux d'échantillonnage, le nombre de coefficients, la longueur de la fenêtre et la longueur du saut. En utilisant les paramètres abordés dans cette section ci-dessus, nous mettons en œuvre la méthode comme :

mfcc_features = librosa.feature.mfcc(signal, sample_rate, n_mfcc=15, hop_length=int(0.010 *sample_rate), n_fft = int(0,025 * sample_rate))

Les fonctions ci-dessus nous permettent d'obtenir 15 valeurs MFCC (1 énergie + 14 coefficients) pour chaque audio dans l'ensemble de données. Comme la taille de la fenêtre et la longueur du saut sont respectivement de 25 ms et 10 ms, et que la longueur des fichiers audio est sensiblement supérieure à ces valeurs, nous prenons la moyenne

des coefficients pour chaque audio et le stocker dans un fichier JSON pour former notre ensemble de données.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	-3.652701	0.938378	-0.055832	0.433431	0.020086	0.118445	0.101112	0.326780	0.296562	0.285342	0.205146	0.247135	0.273065	0.247256	0.215797
1	-3.658557	0.957359	-0.034329	0.365960	0.134468	0.084634	0.136040	0.314334	0.219138	0.301665	0.190921	0.268044	0.222324	0.251218	0.246782
2	-3.677225	0.816670	-0.047907	0.406790	0.034803	0.149239	0.146861	0.304800	0.274004	0.323590	0.215839	0.285010	0.248308	0.284956	0.234261
3	-3.655680	0.943684	-0.118401	0.313354	0.072897	0.168432	0.161171	0.332029	0.255196	0.318065	0.199106	0.252691	0.227411	0.291281	0.238764
4	-3.617184	1.133024	-0.119696	0.271080	0.066272	0.166422	0.124647	0.295460	0.211888	0.302655	0.206893	0.225544	0.245500	0.254522	0.232973
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
4210	-3.697447	0.146822	0.060332	0.619398	-0.049668	0.478666	0.153453	0.278556	0.273735	0.243274	0.336396	0.304937	0.325221	0.225988	0.300336
4211	-3.692879	0.231139	0.003643	0.590551	-0.104707	0.424166	0.099386	0.308923	0.259177	0.301235	0.309240	0.254851	0.364605	0.301345	0.349324
4212	-3.666521	0.333865	-0.170916	0.621836	-0.130528	0.418122	0.040614	0.306175	0.267645	0.281882	0.336562	0.270043	0.423674	0.299037	0.368510
4213	-3.693778	0.256340	0.019444	0.580591	-0.110204	0.486470	0.137771	0.287480	0.260192	0.268127	0.343288	0.310660	0.330176	0.226754	0.296689
4214	-3.698888	0.222330	-0.042375	0.497466	-0.084774	0.371684	0.128848	0.283279	0.366548	0.285240	0.355309	0.315285	0.364487	0.293699	0.341862

4215 rows × 15 columns

Figure 39 : Les caractéristiques de MFCC dans un DataFrame

LPC

Toutes comme les MFCCs, nous allons extraire les coefficients de LPC mais cette fois-ci nous avons utilisé une bibliothèque sous nom de Audiolazy qui nous permet d'avoir les caractéristiques d'un fichier audio, avec 15 caractéristiques lpc (signal, ordre=15).

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	-1.613728	1.271305	-0.849101	0.792407	-0.903176	0.952619	-0.903289	0.832086	-0.707718	0.646451	-0.451268	0.352547	-0.208379	0.209760	-0.156936
1	-1.290846	0.568136	-0.403060	0.974739	-1.217318	0.924176	-0.652244	0.656264	-0.640542	0.570066	-0.381539	0.377772	-0.250447	0.133815	-0.071486
2	-1.475860	0.965703	-0.525345	0.605267	-0.877426	1.015318	-0.931059	0.855657	-0.725164	0.654266	-0.589902	0.528987	-0.301154	0.206112	-0.116809
3	-1.642322	1.360097	-1.061341	1.101503	-1.258679	1.399490	-1.215761	0.979778	-0.887217	0.883783	-0.676238	0.512432	-0.331074	0.215913	-0.087111
4	-1.279579	0.855101	-0.973753	1.148826	-0.873675	0.825336	-0.873696	0.800076	-0.683201	0.700516	-0.540487	0.386594	-0.123298	-0.020578	0.047297
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
4210	-0.752000	0.822492	-0.462226	0.284957	-0.097652	0.381730	-0.339401	0.512237	-0.303398	0.282214	-0.224860	0.314088	-0.186669	0.150642	-0.092092
4211	-0.910015	1.045513	-0.780014	0.536305	-0.220684	0.358673	-0.291759	0.535075	-0.498997	0.476536	-0.405356	0.294107	-0.118117	0.135577	0.028670
4212	-1.119184	1.396706	-1.190778	0.958977	-0.630218	0.637335	-0.591999	0.745809	-0.656984	0.622767	-0.453932	0.307878	-0.076612	0.100763	-0.012334
4213	-0.806142	0.879056	-0.548723	0.370164	-0.177446	0.475761	-0.360685	0.552313	-0.328311	0.289344	-0.237490	0.318170	-0.179503	0.164853	-0.068932
4214	-0.995461	1.213325	-0.884743	0.547068	-0.203749	0.245257	-0.254994	0.536654	-0.494587	0.501695	-0.305001	0.346683	-0.140932	0.154406	-0.075707

4215 rows × 15 columns

Figure 40 : Les caractéristiques de LPC dans un DataFrame

3.3 CLASSIFICATION

Comme indiqué précédemment les classificateurs d'apprentissage machine utilisés pour cette étude sont SVM, Random Forest, k Nearest Neighbors et ANN. Tous ont été implémentés en utilisant la bibliothèque sklearn sauf ANN où nous avons utilisé Tensorflow et Keras pour construire l'architecture des réseaux du neurones et a entrainé le modèle.

Pour SVM, nous utilisons la classe SVC du paquet sklearn.svm, nous avons utilisé le noyau RBF pour la classification et C = 10 avec gamma = 0,1.

Pour Random Forest, nous avons utilisé la classe RandomForestClassifier du paquet sklearn.ensemble. Nous avons utilisé les valeurs de Gini pour calculer les divisions de l'arbre car il s'agit d'un problème de classification. En outre, nos estimateurs (nombre d'arbres) seront de 200 car un estimateur plus élevé nous donnera de meilleurs résultats et évitera également le problème du overfitting.

Concernant k Nearest Neighbours, nous utilisons la classe KNeighborsClassifier du paquet sklearn.neighbors. Comme nous l'avons vu précédemment, la distance euclidienne est la forme de mesure de distance la plus largement acceptée pour kNN, c'est pourquoi nous l'utilisons en fixant la mesure comme Minkowski et les valeurs p comme 2. Cette combinaison indique à sklearn que la métrique de distance à utiliser est euclidienne.

Finalement, Pour ANN nous avons implémenter ce dernier avec Keras et TF comme des Framework de Deep Learning afin de simplifier la création des architecture des réseaux de neurone et les entraîné facilement, et à l'aide de GPU intégré dans le Colab nous avons pu entrainer nos modèle de ANN dans un temps réduit, et pour bien avoir les hyperparamètres les plus efficients pour notre model nous avons utilisé le Keras Tuner, ci-dessous vous aller trouver les paramètres utilisé pour L'ANN, ainsi que l'architectures des réseaux de neurones sur laquelle nous avons entrainé nous model de ANN.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
flatten (Flatten)            (None, 15)                0
_________________________________________________________________
dense (Dense)                (None, 416)               6656
_________________________________________________________________
dropout (Dropout)            (None, 416)               0
_________________________________________________________________
dense_1 (Dense)              (None, 384)               160128
_________________________________________________________________
dropout_1 (Dropout)          (None, 384)               0
_________________________________________________________________
dense_2 (Dense)              (None, 384)               147840
_________________________________________________________________
dropout_2 (Dropout)          (None, 384)               0
_________________________________________________________________
dense_3 (Dense)              (None, 75)                28875
=================================================================
Total params: 343,499
Trainable params: 343,499
Non-trainable params: 0
_________________________________________________________________
```

Figure 41 : L'architecture de notre réseau de neurone

En outre, pour tous les classificateurs ci-dessus, nous effectuons une validation stratifiée décuplée afin de garantir une distribution cohérente des classes lors de la phase de formation et d'essai et de s'assurer que notre modèle ne se surajoute pas pour une répartition particulière.

Modèles de classifications	**Hyperparamètres**
SVM	Kernel = RBF C = 10 Gamma = 0.1
RF	N_Estimator = 200
KNN	Distance = Euclidien P = 2 K = 5
ANN	Nombre de couches caché = 3 Activation = ReLu Dropout rate = 0.3 Epochs = 100

Batch size = 32
Learning rate = 0.001

Tableau 3 : La configuration et paramètres pour chaque modèle

3.4 Conclusion

Après avoir implémenté toutes les techniques soit pour le prétraitement, extraction des caractéristiques, ou soit pour la classification, nous allons comparer dans le chapitre suivant les deux techniques d'extraction des caractéristiques sur notre ensemble de données audio, et nous utilisons les classificateurs d'apprentissage automatique pour déterminer quelle combinaison d'extraction des caractéristiques et d'apprentissage automatique fonctionne le mieux pour une telle étude.

4 MESURES D'EVALUATION

L'évaluation du modèle d'apprentissage machine est une partie importante de toute étude de recherche sur l'apprentissage machine. Un modèle peut donner une bonne note en utilisant une mesure mais peut avoir du mal à être performant en utilisant une autre. Il est donc essentiel d'identifier d'abord les paramètres d'évaluation adaptés à notre recherche, puis d'évaluer les modèles sur la base de ces paramètres.

Comme notre étude est fondamentalement une classification multi-classes, nous devons d'abord comprendre les mesures d'évaluation utilisées dans de tels cas. Les valeurs des vrais positifs, des vrais négatifs, des faux positifs et des faux négatifs sont largement utilisées pour calculer ces paramètres. On peut mieux les comprendre à l'aide d'une matrice de confusion.

4.1 MATRICE DE CONFUSION

Comme son nom l'indique, une matrice de confusion est une matrice 2-D qui décrit la performance globale du modèle [31]. Les lignes et les colonnes de la matrice sont marquées par des classes, et les diagonales indiquent les classifications correctes. Ces diagonales sont donc les vrais positifs dans une matrice de confusion multi-classes. La figure 36 montre un exemple de matrice de confusion et comment les valeurs Vrai Positif (TP), Vrai Négatif (TN), Faux Positif (FP) et Faux Négatif (FN) en sont dérivées.

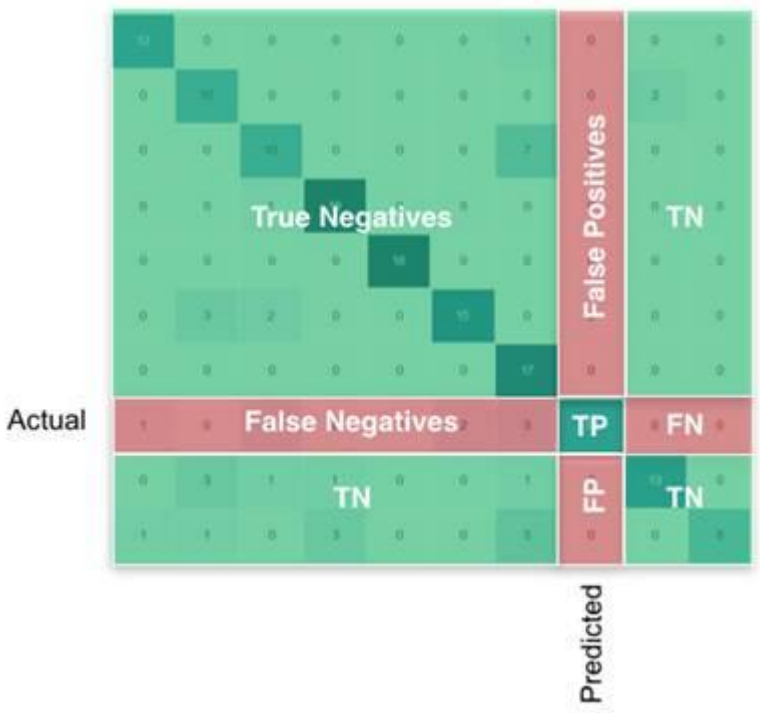


Figure 42 : Matrice de confusion

4.2 PRECISION DE LA CLASSIFICATION (ACCURACY)

La formule de précision est définie comme suit :

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP}$$

C'est une très bonne mesure d'évaluation pour la classification binaire. Cependant, pour une classification multi-classes, cette mesure n'est pas considérée comme correcte car les vrais négatifs ne sont "vrais" que du point de vue d'une classe. Dans l'ensemble, un vrai négatif est un cas mal classé et ne peut être considéré comme une correction correcte.

C'est pourquoi nous utilisons le concept d'exactitude de la classification. L'exactitude de la classification est définie comme suit :

$$Accuracy = \frac{TP}{TP + TN + FN + FP}$$

Ici, seules les vraies valeurs positives sont prises en compte pour le calcul de la précision et constituent donc une mesure d'évaluation plus efficace que la précision traditionnelle.

La précision de la classification est une très bonne mesure si nous avons un nombre égal d'audios pour chaque classe. Cependant, c'est rarement le cas et même dans cette étude, l'ensemble des données n'est pas équilibré de manière égale.

Ainsi, bien qu'elle soit une bonne mesure d'évaluation, elle n'est pas parfaite pour une telle étude.

4.3 PRECISION

Dans les problèmes de classification, la précision est une mesure largement utilisée pour évaluer la qualité d'un modèle. Elle est particulièrement utile lorsque l'ensemble de données contient de multiples faux positifs qui faussent la précision globale [32]. La formule de précision est :

$$Precision = \frac{TP}{TP+FP}$$

Et se concentre donc moins sur les faux négatifs. Ainsi, la mesure est bonne lorsqu'elle est utilisée dans des problèmes où les FN ne sont pas une grande préoccupation. Cependant, dans notre étude, un faux négatif peut être critique du point de vue d'un système biométrique car il peut potentiellement compromettre la sécurité d'un client. Par conséquent, bien qu'il s'agisse d'une bonne mesure pour un tel ensemble de données, elle pourrait ne pas être tout à fait appropriée pour ce scénario.

4.4 RECALL

Contrairement à Precision, Recall se concentre davantage sur les faux négatifs. Il est calculé comme :

$$Recall = \frac{TP}{TP + FN}$$

Et constitue donc une très bonne mesure pour les études où un faux négatif est absolument inacceptable.

Par exemple, lors de la classification d'une maladie cardiaque, un faux négatif signifierait que l'on diagnostique une personne "non affectée" alors qu'elle est réellement affectée. Cependant, en raison de ce biais vers les faux négatifs, un rappel ne permet pas d'évaluer correctement les faux positifs. Bien que les faux positifs aient n impact moindre sur cette étude, ils n'en sont pas moins importants et le fait d'utiliser le rappel uniquement pour mesurer la métrique d'évaluation n'est peut-être pas la meilleure solution.

4.5 F1 SCORE

Le F1-Score est utilisés pour réduire les effets de compromis créés par la précision et le rappel. Il combine les deux mesures pour trouver un équilibre entre leurs limites et est particulièrement efficace pour les ensembles de données déséquilibrés. F1 Score est obtenue en calculant la moyenne pondérée de la précision et du rappel, ce qui permet d'accorder la même importance aux faux positifs et aux faux négatifs. La formule du F1 score est définie comme suit :

$$F1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Ainsi, un F1 Score plus élevé indique des FN et FP plus faibles sans tenir compte du biais créé par les vrais négatifs dans le calcul. C'est précisément ce que tente d'obtenir un modèle de classification multi classe déséquilibré et par conséquent, le F1 Score est généralement considéré comme la mesure la plus efficace pour évaluer un tel modèle.

Chapitre 4 : EXPERIENCES ET CONCEPTION DU PROJET

Dans ce chapitre on va voir toutes les expériences qu'on a implémenté, en plus de leurs résultats, afin de comparer entre les techniques d'extraction des caractéristiques et les modèles de classifications, pour avoir la combinaison la plus performante entre ces techniques et modèles de classification, dans le but d'avoir la meilleure précision pour notre ASR.

1 EXPERIENCES

1.1 EXPERIENCE 1 : SVM + MFCCS COEFFICIENTS

Dans cette expérience, nous testons notre ensemble de données avec les coefficients MFCC (15 attributs) en utilisant le classificateur SVM. Comme nous l'avons vu dans la section sur la mise en œuvre nous utilisons le Noyau RBF : Sur la base de notre implémentation, et on va invoquer le K-Fold : avec k=10 Afin d'effectue une validation stratifiée décuplée et évalue le modèle.

Le nombre total d'instances correctement classées après avoir entrainé et évaluer le modèle est de 1042 sur 1054 et le nombre total d'instances incorrectement classées est de 11 sur 1054.

	Precision	Recall	F1-score	Support
Accuracy			98.95	1054
Macro avg	98.92	98.90	98.98	1054
Weighted avg	99.02	98.95	98.95	1054

Tableau 4 : Résultats de classification pour la combinaison de SVM et MFCC

Vous trouverez ci-dessous la carte thermique de la matrice de confusion utilisant le noyau RBF sur les coefficients MFCC :

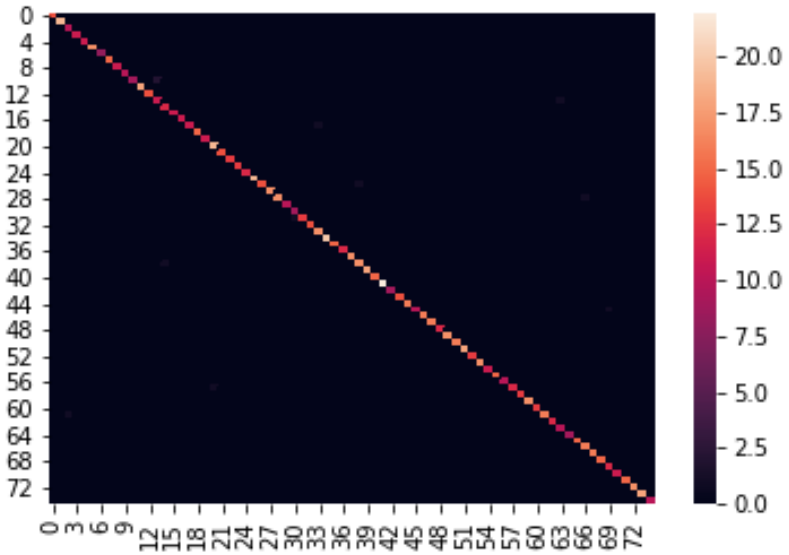


Figure 43 : Matrice de confusion pour les résultats de classification de SVM avec MFCC

1.2 EXPERIENCE 2 : SVM + LPC COEFFICIENTS

Dans cette expérience, nous testons notre ensemble de données avec les coefficients LPC (15 attributs) en utilisant le classificateur SVM. Comme indiqué dans la section sur la mise en œuvre, nous effectuons l'expérience en utilisant le noyau RBF.

Le nombre total d'instances correctement classées est de 928 sur 1054 et le nombre total d'instances incorrectement classées est de 126 sur 1054.

	Precision	Recall	F1-score	Support
Accuracy			88.04	1054
Macro avg	88.86	87.77	87.81	1054
Weighted avg	88.81	88.04	87.96	1054

Tableau 5: Résultats de classification pour la combinaison de SVM et LPC

Vous trouverez ci-dessous la carte thermique de la matrice de confusion utilisant le noyau RBF sur les coefficients LPC :

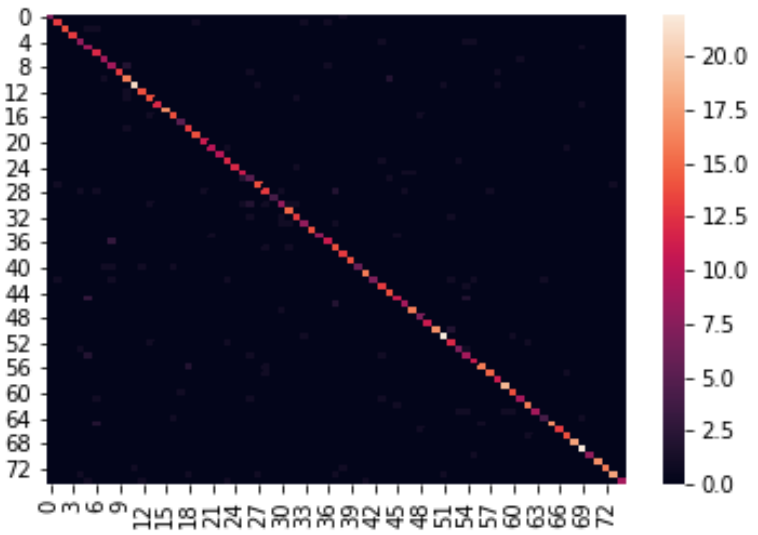


Figure 44 : Matrice de confusion pour les résultats de classification de SVM avec LPC

Comparaison :

Donc pour le SVM on peut conclure que les MFCCs donne plus de précision que les LPCs, et le tableau ci-dessous donne le F1-score après l'évaluation avec les 10 k-Fold.

Modèle	*Extraction des caractéristiques*	*F1-score*
SVM	MFCC	97.02 %
	LPC	87.28 %

Tableau 6 : : Comparaison entre les MFCC et LPC pour le SVM

1.3 EXPERIENCE 3 : RANDOM FOREST + MFCCS COEFFICIENTS

Cette fois-ci on va tester notre Dataset sur le modèle de Random Forest et les caractéristiques de MFCCs, avec la configuration qu'on a donnée ci-dessus, Le nombre total d'instances correctement classées est de 1028 sur 1054 et le nombre total d'instances incorrectement classées est de 25 sur 1054, ainsi on a obtenu les résultats suivants :

	Precision	Recall	F1-score	Support
Accuracy			97.62	1054
Macro avg	97.54	97.47	97.39	1054
Weighted avg	97.83	97.62	97.63	1054

Tableau 7 : Résultats de classification pour la combinaison de Random Forest et MFCC

On peut constater bien la classification avec la matrice de confusion ci-dessous :

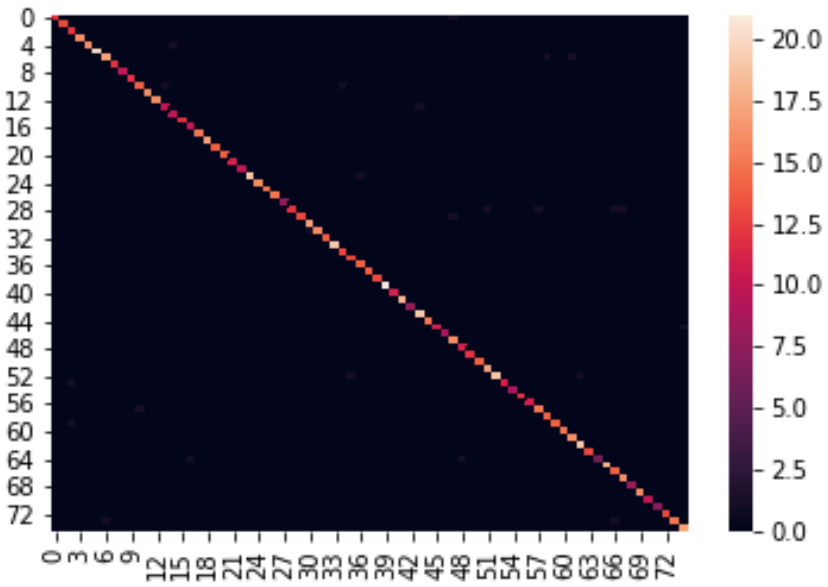


Figure 45 : Matrice de confusion pour les résultats de classification de Random Forest avec MFCC

1.4 EXPERIENCE 4 : RANDOM FOREST + LPC COEFFICIENTS

Dans cette expérience, nous allons tester avec les coefficients LPC en utilisant le classificateur Random Forest.

Le nombre total d'instances correctement classées est de 840 sur 1054 et le nombre total d'instances incorrectement classées est de 214 sur 1054.

	Precision	Recall	F1-score	Support
Accuracy			79.69	1054
Macro avg	80.98	79.60	78.89	1054
Weighted avg	81.72	79.69	79.34	1054

Tableau 8 : Résultats de classification pour la combinaison de Random Forest et LPC

Vous trouverez ci-dessous la carte thermique de la matrice de confusion utilisant Random Forest sur les coefficients LPC :

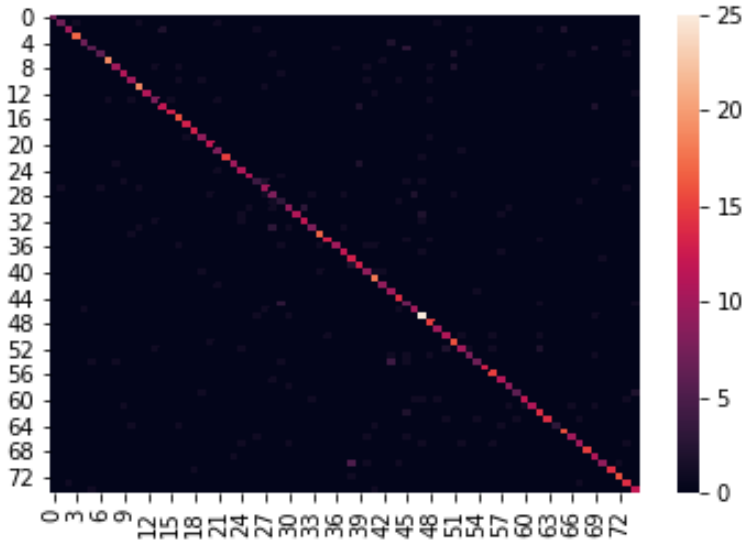


Figure 46 : Matrice de confusion pour les résultats de classification de Random Forest avec LPC

Comparaison :

Alors comme pour L'SVM encore on trouve que les MFCCs avec le Random Forest est plus performantes que les LPC.

Le tableau ci-dessous donne le F1-score après l'évaluation avec les 10 k-Fold de notre modèle de RF.

Modèle	*Extraction des caractéristiques*	*F1-score*
RandomForest	MFCC	98.02 %
	LPC	80 %

Tableau 9 : Comparaison entre les MFCC et LPC pour le RF

1.5 EXPERIENCE 5 : KNN + MFCCS COEFFICIENTS

Dans cette 5ème expérience on a pris les MFCCs comme des caractéristiques pour l'entrée de modèle de machine Learning KNN avec k = 3 comme on a indiqué au-dessus.

Le nombre total d'instances correctement classées est de 1032 sur 1054 et le nombre total d'instances incorrectement classées est de 21 sur 1054. Et on a obtenu les résultats suivants :

	Precision	Recall	F1-score	Support
Accuracy			98.07	1054
Macro avg	98.16	98.02	97.99	1054
Weighted avg	98.19	98.00	97.99	1054

Tableau 10 : Résultats de classification pour la combinaison de KNN et MFCC

Vous trouverez ci-dessous la carte thermique de la matrice de confusion utilisant KNN sur les coefficients MFCC :

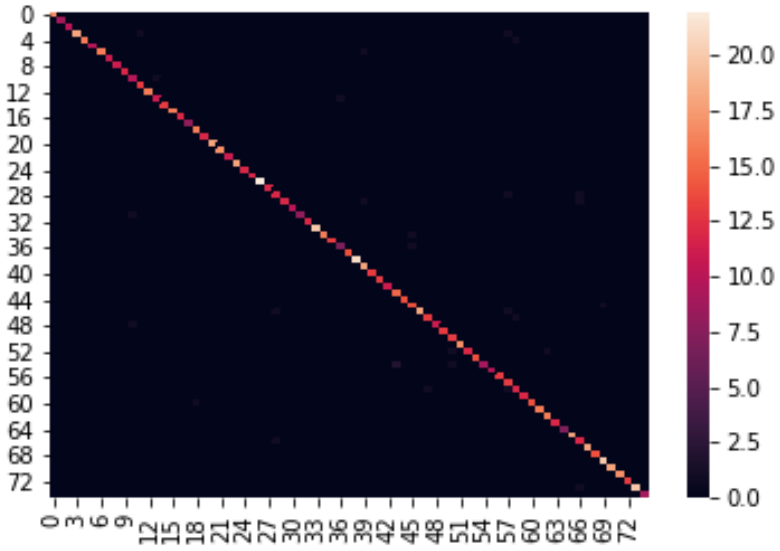


Figure 47 : Matrice de confusion pour les résultats de classification de KNN avec MFCC

1.6 EXPERIENCE 6 : KNN + LPC COEFFICIENTS

Maintenant on va voir la combinaison entre le KNN et les LPCs, d'après cette expérience on a obtenu les résultats suivants :

Le nombre total d'instances correctement classées est de 840 sur 1054 et le nombre total d'instances incorrectement classées est de 214 sur 1054.

	Precision	Recall	F1-score	Support
Accuracy			79.69	1054
Macro avg	80.02	78.72	78.15	1054
Weighted avg	80.95	79.69	79.35	1054

Tableau 11: Résultats de classification pour la combinaison de KNN et LPC

Vous trouverez ci-dessous la carte thermique de la matrice de confusion utilisant le KNN sur les coefficients LPC

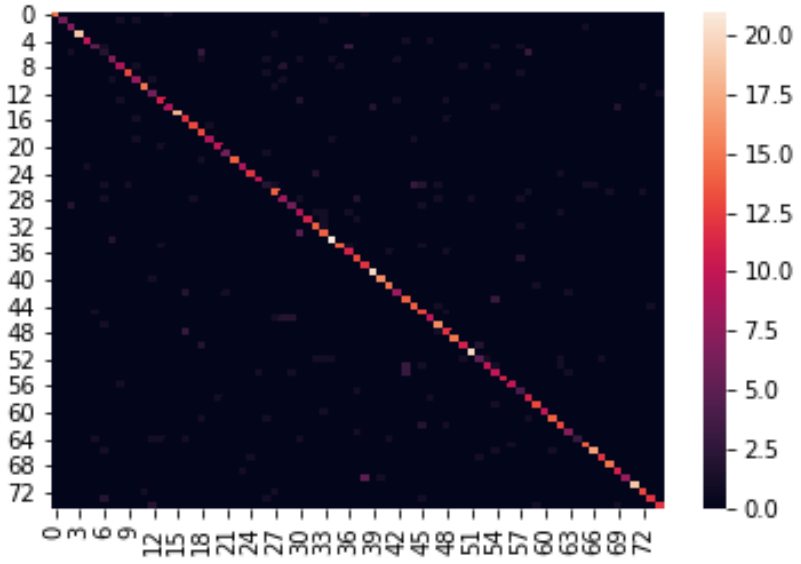


Figure 48: Matrice de confusion pour les résultats de classification de KNN avec LPC

Comparaison :

Donc les MFCCs restent les plus performante, comme ont pu le voir dans le tableau suivant on a eu 97 % précision pour le KNN avec MFCC, cependant on a eu 81% avec LPC comme une technique d'extraire les caractéristiques.

Le tableau ci-dessous donne le F1-score après l'évaluation avec les 10 k-Fold pour le modèle de KNN.

Modèle	*Extraction des caractéristiques*	*F1-score*
KNN	MFCC	97 %
	LPC	81 %

Tableau 12: Comparaison entre les MFCC et LPC pour le KNN

1.7 EXPERIENCE 7 : ANN + MFCC COEFFICIENTS

Dans cette expérience on va passer vers les réseaux de neurones afin de bien tester nos données avec différents modèles de classification, donc on va utiliser les réseaux de neurone artificiel avec l'architectures et les hyperparamètres mentionnés ci-dessus, et on commence par les MFCCs comme une technique d'extraction des caractéristiques.

Le nombre total d'instances correctement classées est de 1022 sur 1054 et le nombre total d'instances incorrectement classées est de 32 sur 1054.

	Precision	Recall	F1-score	Support
Accuracy			96.96	1054
Macro avg	96.94	96.62	96.49	1054
Weighted avg	97.32	96.96	96.95	1054

Tableau 13 : Résultats de classification pour la combinaison de ANN et MFCC

Vous trouverez ci-dessous la carte thermique de la matrice de confusion utilisant ANN sur les coefficients MFCC :

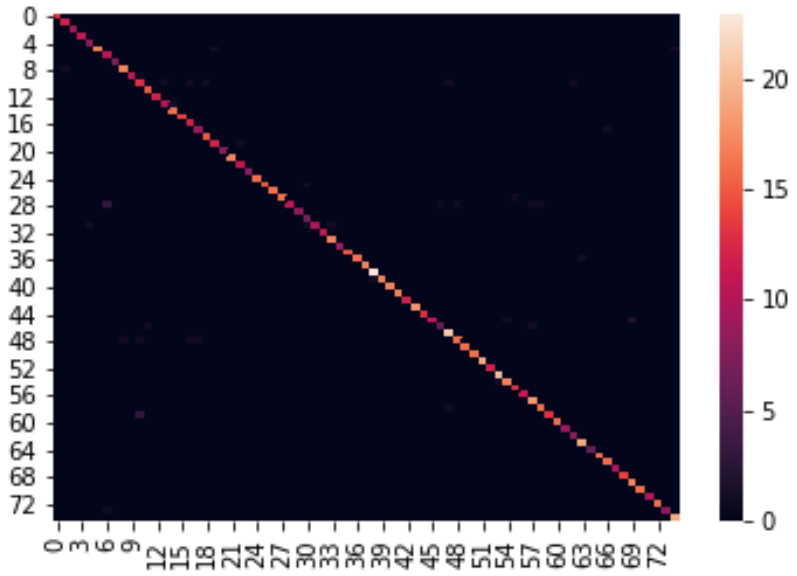


Figure 49 : Matrice de confusion pour les résultats de classification de ANN avec MFCC

1.8 EXPERIENCE 8 : ANN + LPC COEFFICIENTS

Finalement, dans cette expérience on a combiné entre LPC et ANN, avec la même architecture et par conséquence on a eu ces résultats :

Le nombre total des locuteurs correctement classées est de 953 sur 1054 et le nombre total des locuteurs incorrectement classées est de 101 sur 1054.

	Precision	Recall	F1-score	Support
Accuracy			90.41	1054
Macro avg	90.33	90.45	90.49	1054
Weighted avg	91.01	90.41	90.91	1054

Tableau 14 : Résultats de classification pour la combinaison de ANN et LPC

Vous trouverez ci-dessous la carte thermique de la matrice de confusion utilisant ANN sur les coefficients LPC :

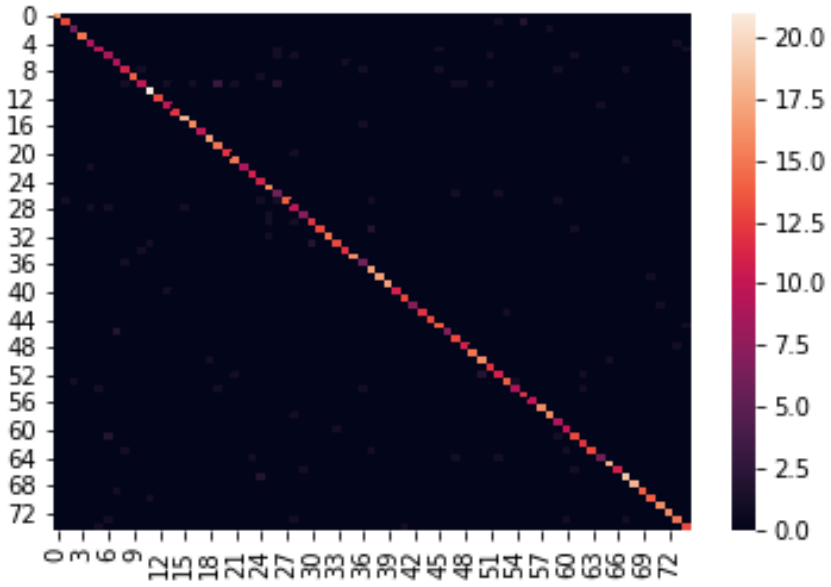


Figure 50: Matrice de confusion pour les résultats de classification de ANN avec LPC

Comparaison :

Ainsi, le ANN performe plus avec les MFCCs qu'avec les LPCs.

Modèle	*Extraction des caractéristiques*	*F1-score*
ANN	MFCC	96.96 %
	LPC	90.91 %

Tableau 15 : Comparaison entre les MFCC et LPC pour le ANN

2 RESULTATS

Ainsi, un total de 8 expériences a été réalisées en utilisant deux ensembles de données ayant un type différent de coefficients MFCC et LPC. Ces deux ensembles de données ont ensuite été chargés dans 4 différents classificateurs d'apprentissage

machine supervisés. Comme il s'agit effectivement d'une recherche de classification multi-classes, les paramètres de rappel, le score F1 et la précision de la classification sont les plus importants pour l'étude. Vous trouverez ci-dessous un résumé des performances de chaque combinaison d'extraction de caractéristiques et de technique d'apprentissage machine par rapport à toutes les autres techniques.

Modèle de classification	*Extraction des caractéristiques*	*F1-score*
SVM	MFCC	97.02 %
	LPC	87.28 %
Random Forest	MFCC	98.02 %
	LPC	80 %
KNN	MFCC	97 %
	LPC	81 %
ANN	MFCC	96.96 %
	LPC	90.91 %

Tableau 16 : Comparaison générale entre toutes les combinaisons

Ainsi on peut conclure que la combinaison de RF et MFCC plus performante pour notre étude, et si on veut le meilleur modèle pour les LPCs alors le ANN est plus efficace pour ce type des caractéristiques.

Vous trouvez ci-dessous une visualisation Bar Chart pour les F1-score de chaque modèle en tenant compte aux techniques d'extraction des caractéristiques.

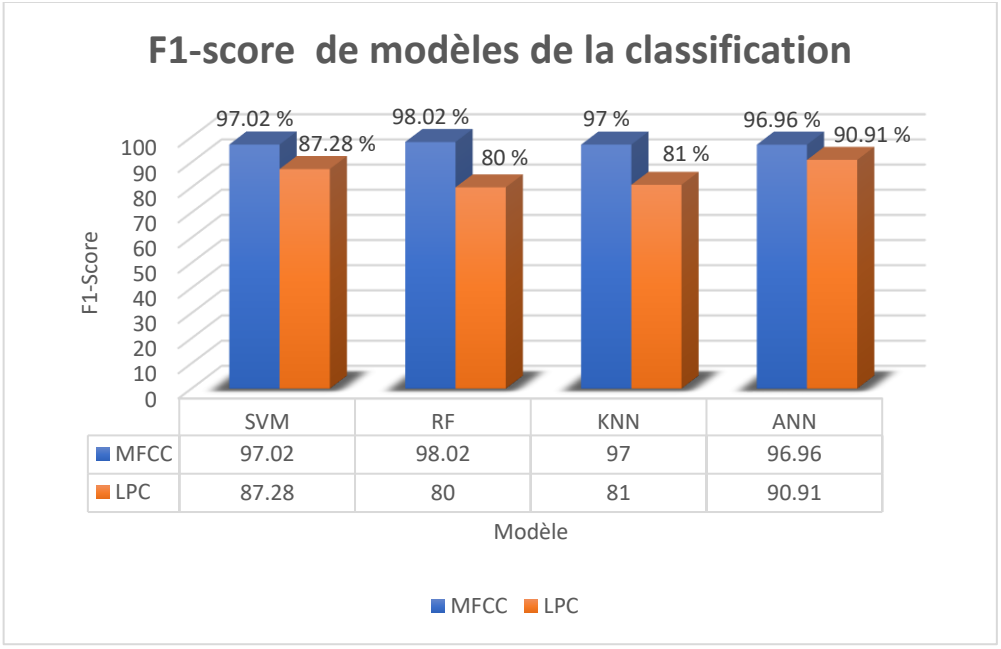


Figure 51 : Bar chart pour la visualisation des F1-score des modèles de classifications

3 APPLICATION DE TEST

Afin d'évaluer Notre modèle, et pour avoir un bon aperçu sur notre travaille et dans le but d'applique le concept de l'interaction homme machine, on a créé une application web dans laquelle on trouve plusieurs options, alors pour cella on a choisi de travailler avec la nouvelle Framework pour les applications web de data science, Streamlit cette dernier est une bibliothèque Python open-source qui permet de créer facilement de magnifiques applications web personnalisées pour l'apprentissage machine et la science des données.

Streamlit crée une API qui adopte le style fonctionnel de Streamlit tout en capturant ces cas d'utilisation aussi simplement que possible, le résultat est le nouveau package streamlit.components.v1 qui comprend trois fonctions. Pour les composants statiques :

• html(...), qui permet de construire des composants à partir de HTML, Javascript et CSS

• iframe(...), qui vous permet d'intégrer des sites web externes

Pour les composants bidirectionnels :

• declare_component(...), qui permet de construire des widgets interactifs qui communiquent de manière bidirectionnelle entre Streamlit et le navigateur.

Le composant Streamlit bidirectionnel se compose de deux parties :

- Un frontend, qui est construit à partir de HTML et de toute autre technologie web que vous souhaitez (JavaScript, React, Vue, etc.), et qui est rendu dans les applications Streamlit via une balise iframe.

- Une API Python, que les applications Streamlit utilisent pour instancier et dialoguer avec ce frontend

Dans notre application on a utilisé les composants bidirectionnels, afin de faire la communication entre python et le React comme une technologie web pour notre application.

Ainsi, pour la page d'accueil on trouve deux composantes principales : Premièrement en gauche on trouve le menu dans le quelle ont choisi la façon de prédictions (Single ou Conversation), et l'option de la manière avec laquelle on va préparer ou bien charger le fichier audio à prédire, on va détailler sur les options dans la suite. Et pour l'autre composante on trouve l'emplacement pour uploader le fichier ou enregistrer le nouveau fichier audio, en plus des buttons soit pour la prédiction, ou l'enregistrement des audios et aussi un bar ou on peut choisir la durée d'audio qu'on veut enregistrer.

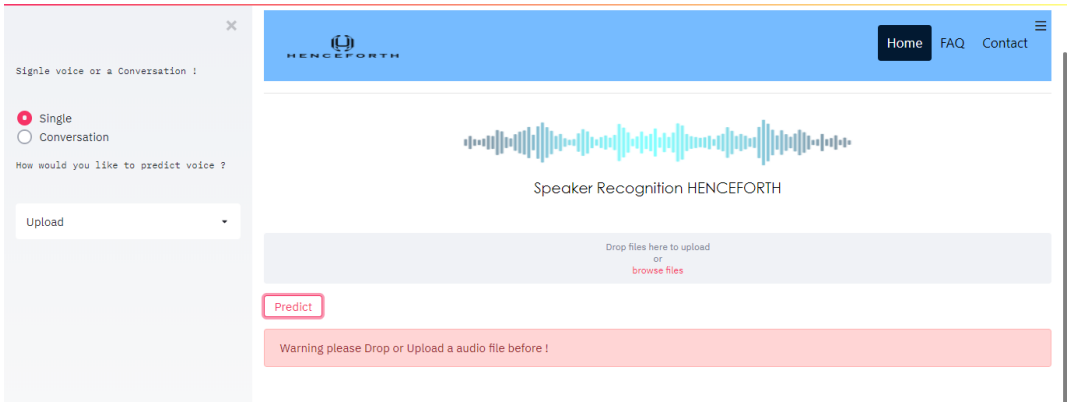


Figure 52 : Capture d'écran de l'application web

- Dans la figure ci-dessous on a les options sur le Menu des options de notre application :

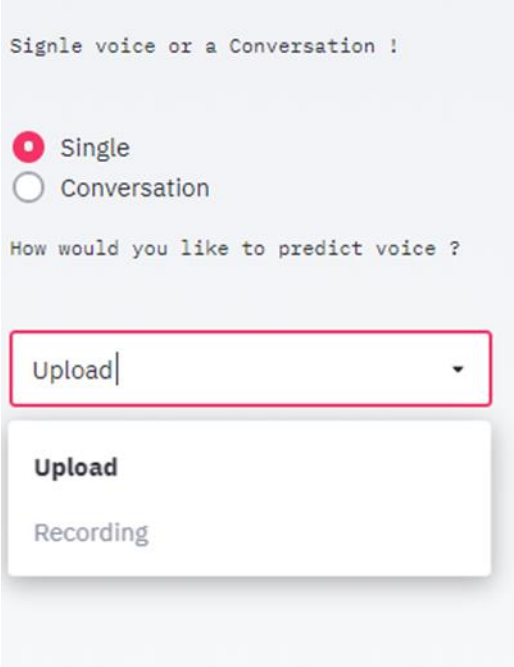


Figure 53 : Capture d'écran du menu des options de l'application

- L'option Single réfère sur si on a par exemple un fichier dans le quelle on veut prédire un seul locuteur, c'est à dire on a juste une voix d'une personne dans le fichier audio.
- L'option Conversation nous permet de prédire plusieurs locuteurs dans une conversation.
- L'option Uploader si on veut prédire avec un fichier audio qui déjà existe sur notre machine.
- Recording si on veut enregistrer directement notre voix ou conversation avec une autre personne et prédire juste après.

- Si on choisit l'option Single et Upload on va avoir l'emplacement de choisir un fichier et le Botton de prédiction.

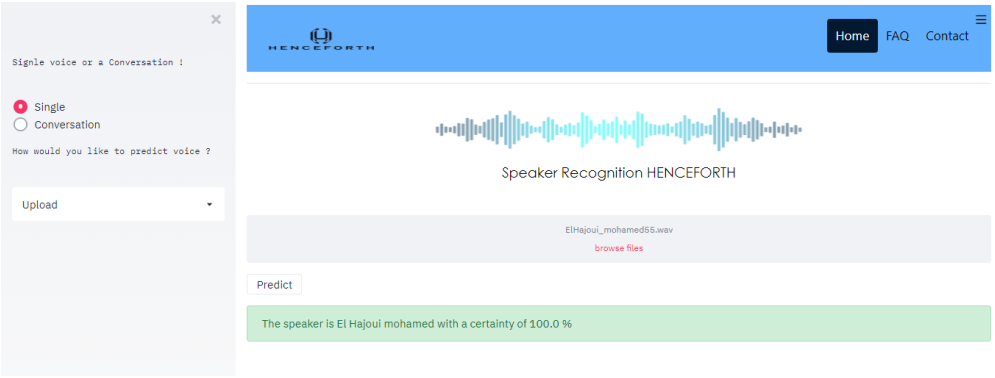


Figure 54 : Capture d'écran de l'application quand on choisit les options Single Upload

- Si on choisit l'autre options de conversation et Upload, dans la prédiction on va avoir la prédiction de deux personnes dans une conversation.

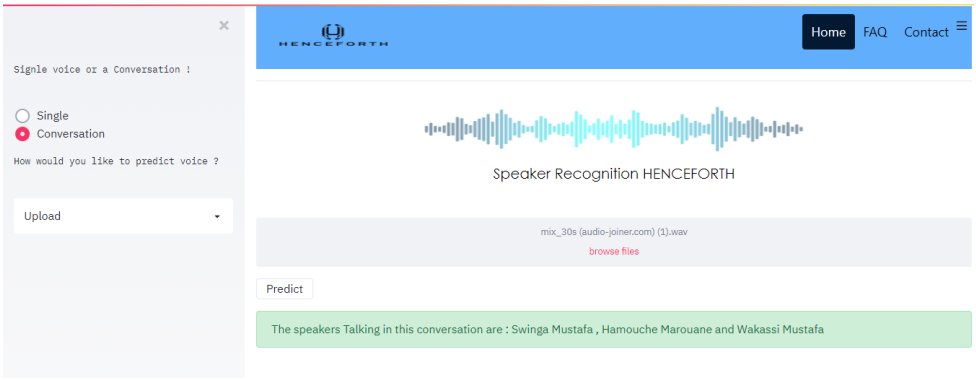


Figure 55 : Capture d'écran de l'application quand on choisit les options conversation et Upload

- Dans le cas où on veut prédire une seule personne avec l'enregistrement de sa propre voix ont choisi les options suivantes :

Dans un premier temps avant toutes on saisit la durée d'enregistrement, après on clique sur le Button 'recording'

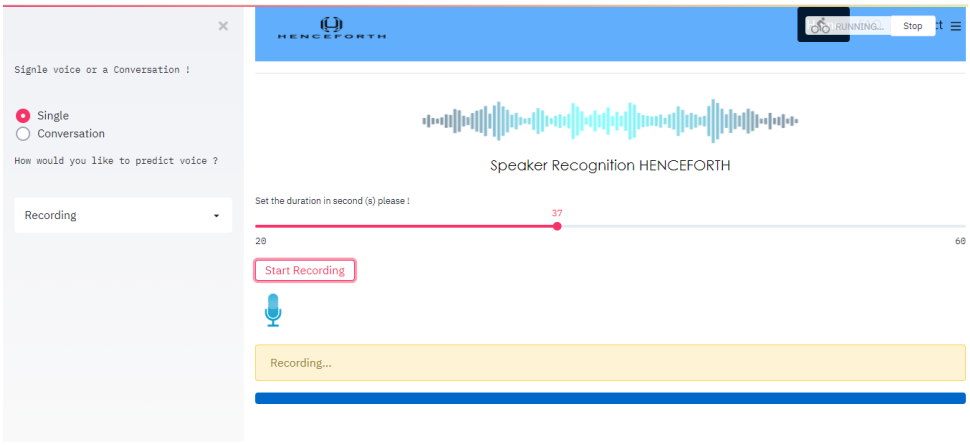


Figure 56 : Capture d'écran de l'application quand on choisit les options Single et Recording.

Ensuite on aura l'audio enregistrer pour le vérifier, en plus de Button 'Predict' pour prédire le locuteur dans l'audio.

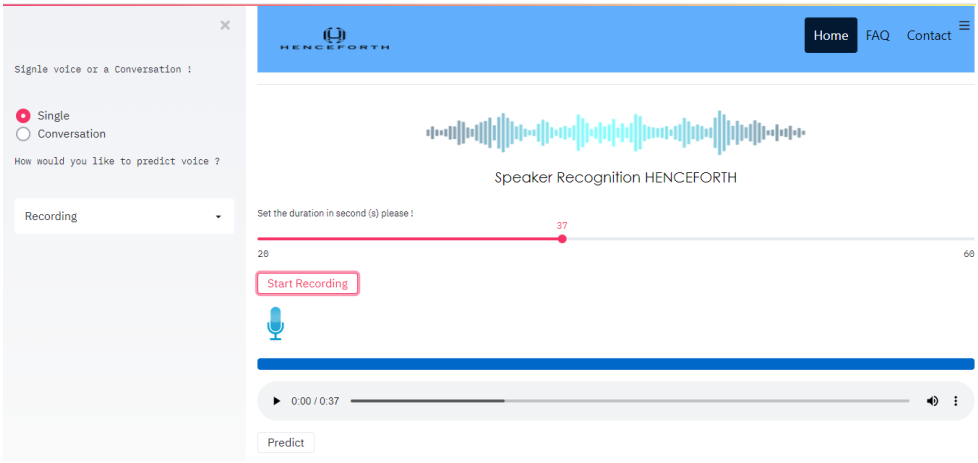


Figure 57 : Capture d'écran lors d'enregistrement d'un nouveau fichier audio

Finalement, on peut voir les résultats de prédiction.

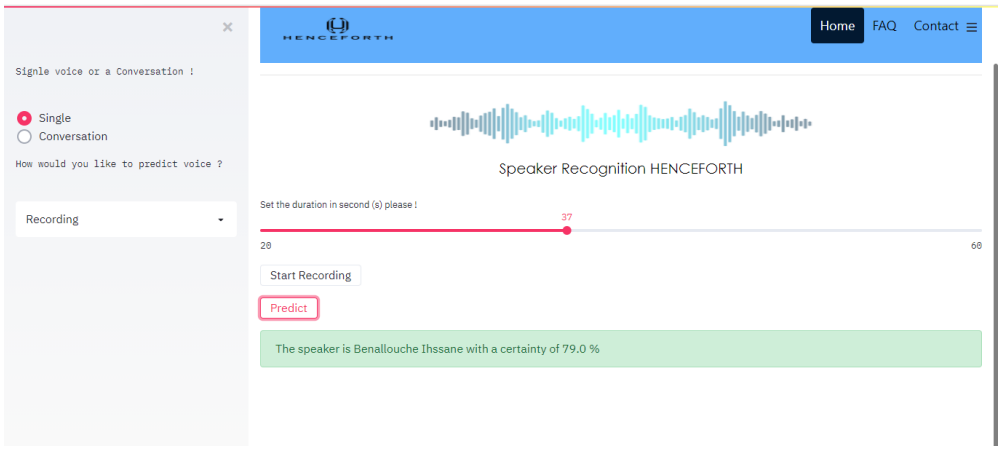


Figure 58 : Capture d'écran lors de prédiction pour le nouveau fichier audio enregistrer

- La même chose pour la prédiction de plusieurs personnes dans une conversation.

Ainsi, notre application web est prête pour tester, et mètre notre modèle à disposition aux utilisateurs.

CONCLUSION GÉNÉRALE

Cette étude est divisée en trois grandes parties : le prétraitement audio, l'extraction des caractéristiques et la classification par apprentissage machine. Du fait que les audios utilisés dans notre étude n'ont pas été enregistrés dans des environnements contraints, le prétraitement audio a constitué une partie extrêmement cruciale du projet. Les trois éléments les plus importants sur lesquels nous nous sommes concentrés pour le prétraitement étaient la suppression du silence, réduction du bruit ambiant et la mise en valeur des voix humaines. Nous y sommes parvenus en utilisant des techniques de VAD et des MFCC pour l'élimination de silence, la réduction du bruit et l'amélioration des voix, respectivement. Comme l'ensemble des données a été échantillonné à 16 kHz, les algorithmes ont fonctionné parfaitement sur tous les ensembles audios, ce qui a permis d'obtenir un signal audio plus clair à une fréquence d'échantillonnage constante. L'extraction des caractéristiques était primordiale, car c'est le cœur de la classification. La conversion des données audio brutes avec des vecteurs significatifs aurait un impact direct sur la façon dont les algorithmes de classification fonctionnent sur cet ensemble de données. C'est pourquoi nous avons décidé d'utiliser les coefficients cepstraux de fréquence Mel (MFCC), et Linear Predict Coefficient (LPC) pour cette phase. De plus, la plupart des études de reconnaissance des locuteurs calculent LPC pour améliorer la précision du modèle. Bien qu'il s'agisse d'un choix efficace, il ne fonctionne mieux que sur les audios sans intervention extérieure du bruit. Pour notre ensemble de données, nous n'avons pas voulu extraire les caractéristiques de haut et de bas de gamme de l'audio afin que l'accent soit mis sur les médiums (voix humaine). Par conséquent, nous avons trouvé que l'utilisation des coefficients MFCC est efficace dans notre étude. Nos résultats se sont avérés corrects puisque nos scores F1 ont augmenté de 10%, 18%, 16%

et 6% en utilisant respectivement, SVM, Random Forest, KNN et ANN sur des modèles formés uniquement avec les coefficients MFCC au lieu des coefficients LPC.

Enfin, le choix d'une technique de classification appropriée par apprentissage machine est également important car ces techniques déterminent la manière dont le modèle interprète les données. Ainsi, en tirant parti des recherches existantes et en appliquant certaines de nos connaissances, nous avons décidé d'opter pour SVM, Random Forest, kNN et ANN. Le SVM est largement utilisé dans la reconnaissance des locuteurs et nous avons donc utilisé cela avec des coefficients MFCC et LPC pour créer une ligne de base, puis nous avons mis en œuvre les RFC, kNN et ANN pour vérifier s'ils vont à l'encontre de l'approche SVM. Alors que kNN et ANN n'a pas réussi à améliorer les performances, avec les MFCC, contrairement au Random Forest Classifiers qui a considérablement amélioré la précision du modèle et nous a donné les meilleurs résultats sur notre ensemble de données, par contre pour les LPCs le ANN a bien réussi d'améliorer la performance avec ce type des caractéristiques.

Ainsi, avec cette étude, nous pouvons conclure que les techniques d'apprentissage par machine peuvent certainement être utilisées pour construire des modèles de reconnaissance de locuteurs pour les audios enregistrés dans des environnements sans contraintes.

En outre, l'étude suggère également qu'en dehors du modèle, la détermination des caractéristiques à extraire de l'audio joue également un rôle essentiel dans la détermination des performances du modèle.

TRAVAIL DE FUTURE

Outre les techniques de traitement audio existantes, il sera intéressant de voir comment d'autres approches peuvent avoir un impact sur l'extraction des caractéristiques et la précision de la classification des modèles d'apprentissage machine. Les audios enregistrés dans un environnement non contraint contiennent beaucoup de bruits indésirables, et il est crucial de les éliminer pour améliorer la précision des modèles. Les approches actuelles généralisent les hautes et les basses fréquences pour limiter leur gain afin de supprimer le bruit ambiant et d'améliorer le discours humain. Bien que cette approche donne de bons résultats, le bruit continue d'exister dans les audios, ce qui limite la précision. Des recherches supplémentaires dans ce domaine pour parvenir à une élimination complète ou presque complète du bruit peuvent améliorer considérablement les tâches ultérieures d'extraction et de classification des caractéristiques pour la reconnaissance des locuteurs, ainsi que d'autre algorithme de Deep Learning comme les CNN et RNN.

REFERENCES

[1] P. R. a. S. L. O. Cohen, « The role of voice input for human-machine communication,» *proceedings of the National Academy of Sciences 92,* p. 22, 1995.

[2] Vaidyanathan, «Generalizations of the sampling theorem: Seven decades after Nyquist.,» *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, 48(9),* pp. 1094-1109, 2001.

[3] J. G. J. a. S. J. Ramirez, «Voice activity detection. fundamentals and speech recognition system robustness. Robust speech recognition and understanding,,» pp. 1-22., 2007.

[4] A. S. E. S. H. M. D. L. C. a. P. Benyassine, « ITU-T Recommendation G. 729 Annex B: a silence compression scheme for use with G. 729 optimized for V. 70 digital simultaneous voice and data applications,» *IEEE Communications Magazine, 35(9,* pp. 64-73, 1997.

[5] R. Meston, «Sorting through GSM codecs: A tutorial. EE Times: Connecting the Global Electronics Community,» *Retrieved from https://www. eetimes. com/document. asp.,* 2003.

[6] P. Mermelstein, « "Distance measures for speech recognition, psychological and instrumental," in Pattern Recognition and Artificial Intelligence,» pp. 374-388, 1976.

[7] S. S. A. a. K. C. Agrawal, « Prosodic feature based text dependent speaker recognition using machine learning algorithms,» *International Journal of Engineering Science and Technology, 2(10),* pp. 5150-5157, 2010.

[8] A. Gill, «A review on feature extraction techniques for speech processing,» *International Journal Of Engineering and Computer Science,,* 2016.

[9] P. a. C. M. Kumar, « Speaker identification using Gaussian mixture models,» *MIT International Journal of Electronics and Communication Engineering, 1(1),* pp. 27-30, 2011.

[10] H. Hermansky, «Perceptual linear predictive (PLP) analysis of speech,» *the Journal of the Acoustical Society of America, 87(4),* pp. 1738-1752, 1990.

[11] Z. Xuegong, «Introduction to statistical learning theory and support vector machines,» *Acta Automatica Sinica, 26(1),* pp. 32-42, 2000.

[12] «Decision Tree,» [En ligne]. Available: https://en.wikipedia.org/wiki/Decision_tree.

[13] O. Harisson, «knn,» 2018. [En ligne]. Available: https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761.

[14] A. n. network. [En ligne]. Available: https://en.wikipedia.org/wiki/Artificial_neural_network.

[15] K. B. R. a. B. S. Davis, «Automatic recognition of spoken digits,» *The Journal of the Acoustical Society of America, 24(6),* pp. 637-642, 1952.

[16] S. Pruzansky, «Pattern-Matching Procedure for Automatic Talker Recognition,» *The Journal of the Acoustical Society of America, 35(3),* pp. 354-358, 1960.

[17] H. a. J. Bao, «The research of speaker recognition based on GMM and SVM,» *IEEE international conference on system science and engineering (ICSSE),* pp. 373-375, 2012.

[18] R. Z. L. F. M. a. H. A. Chakroun, «A novel approach based on Support Vector Machines for automatic speaker identification.,» *IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA),* pp. 1-5, 2015.

[19] S. a. S. S. Tirumala, «A review on deep learning approaches in speaker identification,» *Proceedings of the 8th international conference on signal processing systems,* pp. 142-147, 2016.

[20] Z. I. A. C. S. S. R. a. G. A. Ge, «Neural network based speaker classification and verification systems with enhanced features,» *IEEE Intelligent Systems Conference (IntelliSys),* pp. 1089-1094, 2017.

[21] J. Garofolo, «TIMIT acoustic phonetic continuous speech corpus,» *Linguistic Data Consortium,,* 1993.

[23] «wikipedia,» [En ligne]. Available: https://en.wikipedia.org/wiki/Artificial_neural_network.

ANNEXE

Vous trouverez ci-dessous les visualisations graphiques de lors de l'entrainement de nos modèles de ANN soit avec les MFCC ou bien pour LPC.

Le premier graphique présente le suivi de l'accuracy et de Loss lors de l'entrainement de ANN avec les MFCCs :

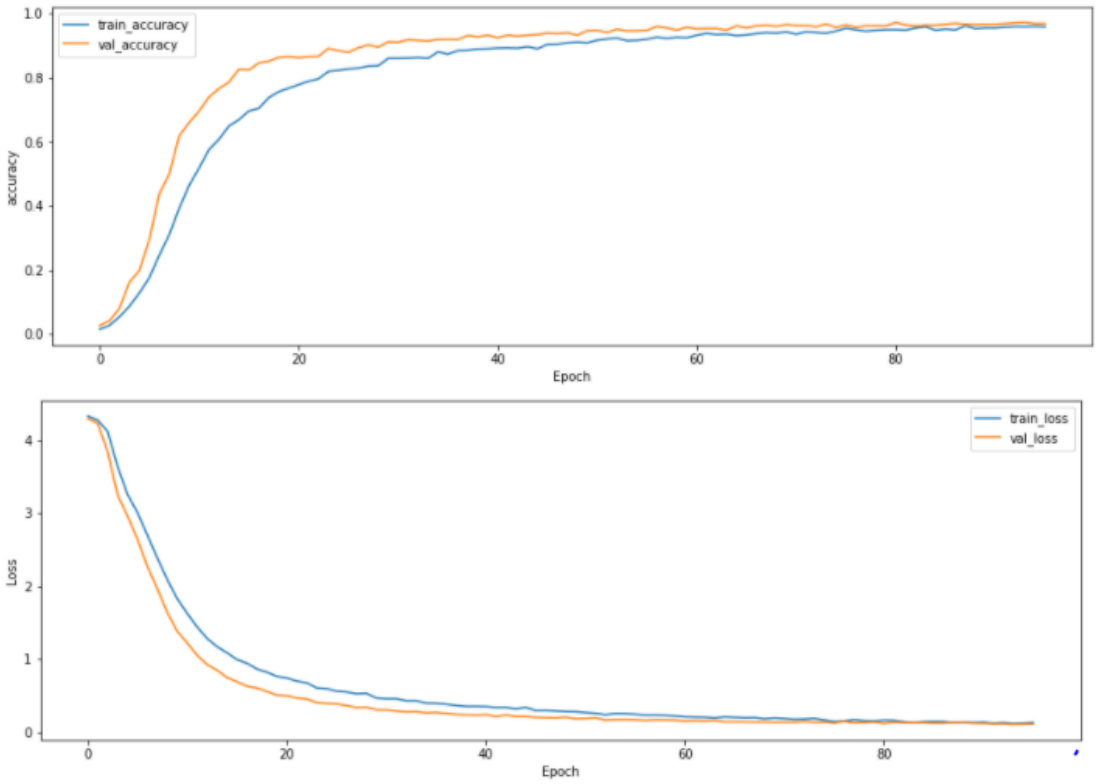


Figure 59: le suivi de l'accuracy et de loss lors de l'entrainement de ANN avec les MFCCs

Le premier graphique présente le suivi de l'accuracy et de Loss lors de l'entrainement de ANN avec les LPCs :

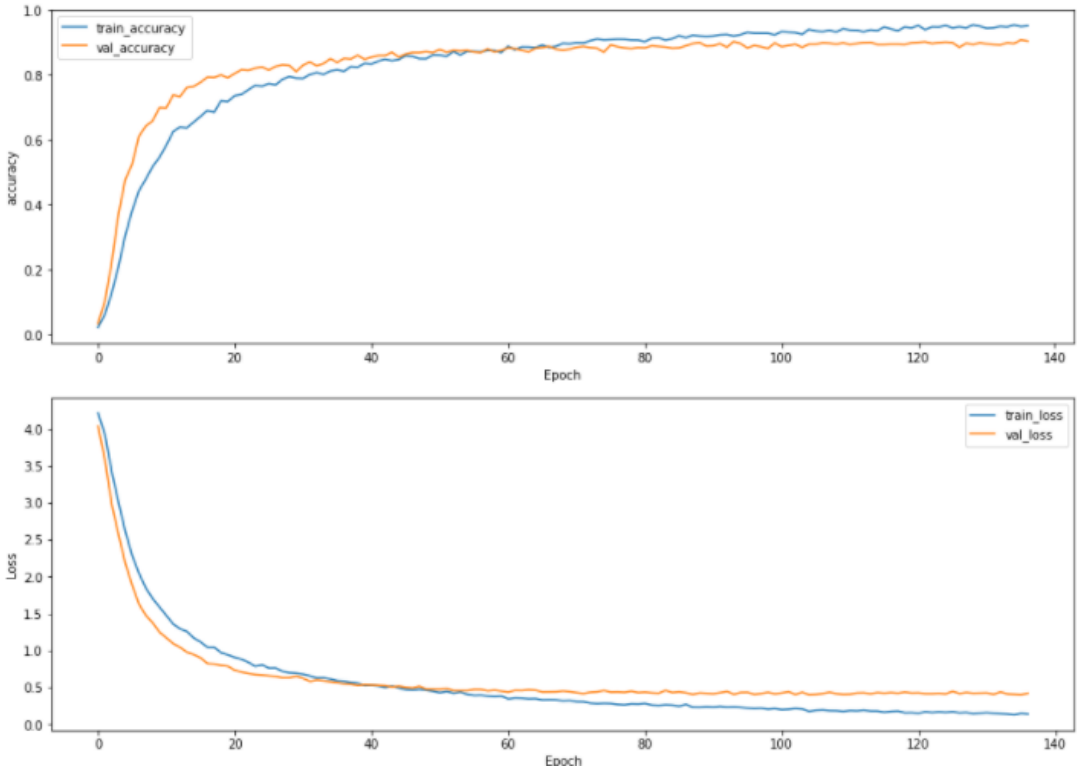


Figure 60: le suivi de l'accuracy et de Loss lors de l'entrainement de ANN avec les LPCs.

Vous trouverez ci-dessous les visualisations graphiques des diagrammes à barres pour chaque série d'expériences :

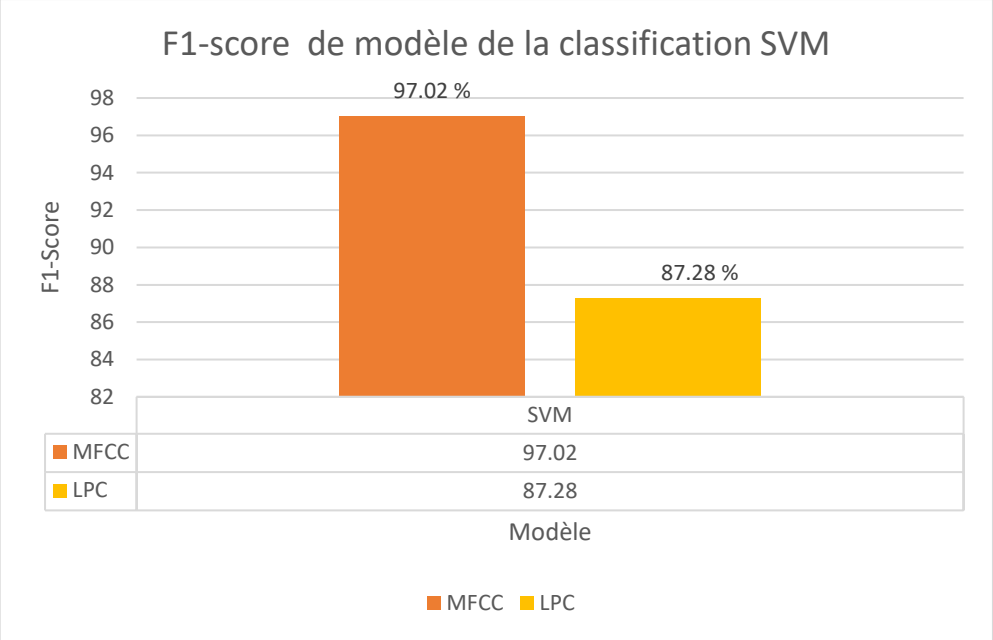


Figure 61:F1-score de modèle de la classification SVM

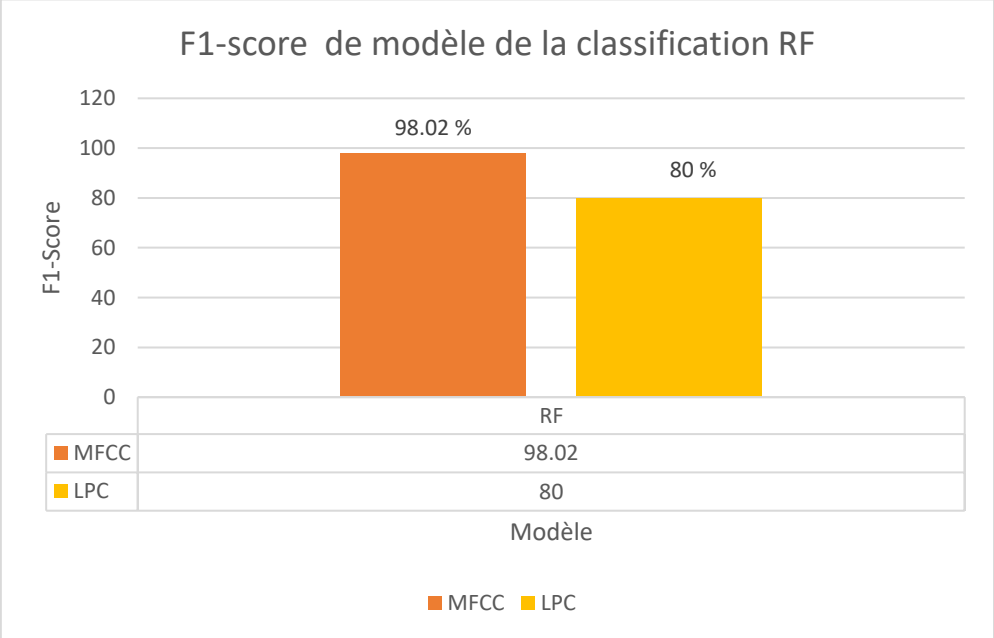


Figure 62: F1-score de modèle de la classification RF

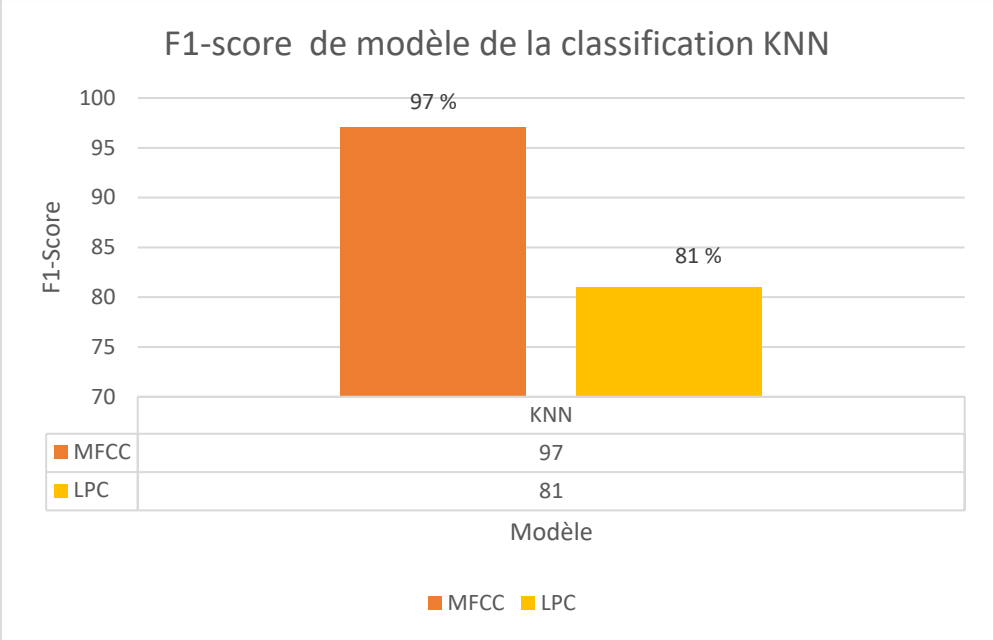


Figure 63 : F1-score de modèle de la classification KNN

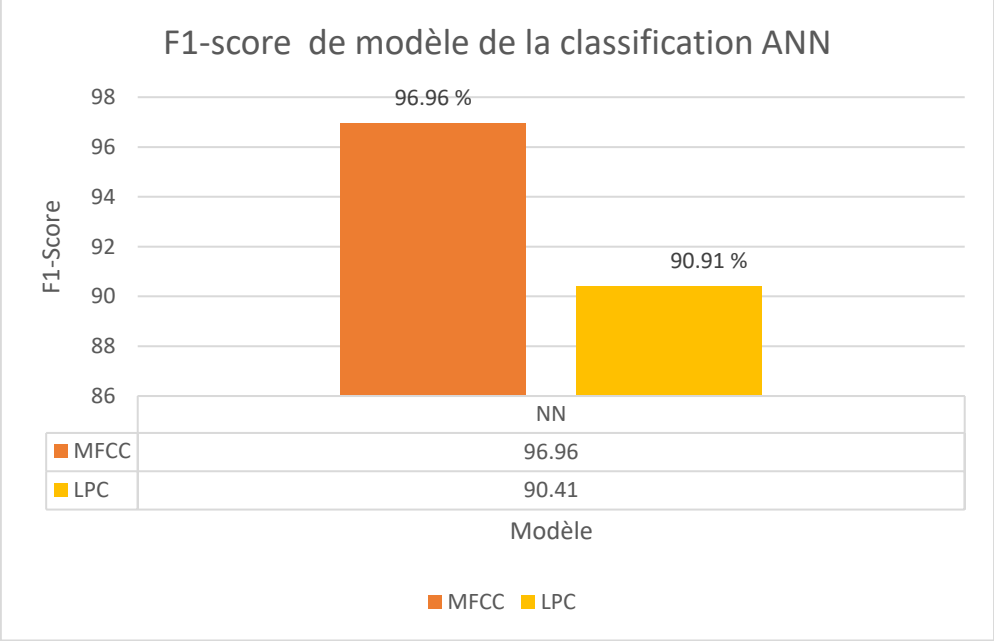


Figure 64: F1-score de modèle de la classification ANN.