Université de Strasbourg

PRISME
Laboratoire Pluridisciplinaire de Recherche
Ingénierie des Systèmes, Mécanique, Énergétique

UNIVERSITE D'ORLEANS

Rapport de stage

$2^{ème}$ année de master CSMI

Apport des méthodes d'apprentissage profond pour la reconnaissance des actes des énoncés oraux

---

Ouachour Hanane

Sous la direction de:

RAVIER Philippe et BOUGRINE Asma

Août 2021

REMERCIEMENTS

À la fin de ce travail, je tiens à exprimer ma profonde gratitude à mon directeur de stage **Mr. RAVIER Philippe** pour le suivi et le soutien qu'il n'a pas cessé de m'apporter tout au long du stage. J'ai beaucoup appris à ses côtés.

J'exprime mes plus vifs remerciements à **Mme. BOUGRINE Asma** pour l'aide technique et théorique qu'elle m'a apporté. Je tiens également à remercier tous **les responsables du Projets RAVIOLI** pour leurs conseils qui m'ont permis de surmonter les difficultés organisationnelles liées à la base de données.

Je tiens également à remercier tous ceux qui ont contribué de près ou de loin à ce stage. Le travail que j'ai accompli dans ce domaine est l'expression de mes remerciements les plus sincères.

Table des matières

Table des figures

Liste des tableaux

Chapitre 1

Introduction

Dans ce chapitre, nous allons présenter le contexte générale du projet, le laboratoire Prisme dans lequel le stage est déroulé, le projet Ravioli et ses objectifs.

1.1 Contexte

La parole est le mode de communication le plus naturel que les humains utilisent pour interagir entre eux. Ceci peut être justifié par le fait que le signal vocal de la parole permet la transmission intelligible d'une importante quantité d'informations. L'analyse de ce signal s'inscrit dans une succession de procédures, que ce soit pour la détermination du fondamental de la parole ou pour la synthèse vocale.

Une émotion peut être exprimée en choisissant les éléments lexicaux adaptés et la syntaxe appropriée. Si tel était le seul moyen d'y parvenir, l'expression verbale ne différerait de l'écriture que parce qu'elle est sonore. Mais la parole est l'émanation d'un être vivant. Le corps et la pensée vont non seulement personnaliser sa parole et la singulariser de toutes les autres, mais également témoigner, consciemment ou non, des émotions, des sentiments, des aspirations, des besoins, de la posture, de sensations d'affaiblissement physique ou de plus grande activité, de l'état de l'appareil phonatoire mais aussi de l'état général de l'organisme.

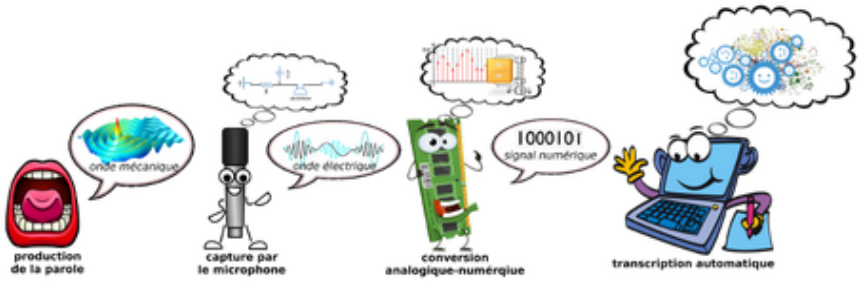

Fig 1.1 – Chaîne d'acquisition et de traitement du signal de la parole [19]

1.2 Présentation du laboratoire PRISME

Le laboratoire Pluridisciplinaire de Recherche en Ingénierie des Systèmes, Mécanique, Énergétique (PRISME) a été créé en 2008 et résulte du regroupement de plusieurs unités de recherche du domaine des sciences et technologies (sous-domaines : SPI et STIC). Le laboratoire PRISME est une unité propre de l'université d'Orléans (EA 4229). Il est structuré en deux départements : [7]

- Le département Fluides, Energie, Combustion, Moteur (FECP) qui développe des actions de recherche dans le domaine des transports et le domaine des systèmes énergétiques.
- Le département Images, Robotiques, Automatique et Signal (IRAuS) qui développe des actions de recherche concernant l'ingénierie des systèmes et les systèmes de traitement de l'information.

1.3 Présentation du projet RAVIOLI

Le projet RAVIOLI (Reconnaissance Automatique des Valeurs Injonctives à l'Oral, Langages en Interaction)s'appuie sur une collaboration régionale pour développer une thématique de recherche totalement innovante : celle du rapport entre prosodie, syntaxique, sémantique et pragmatique dans la perspective de l'analyse linguistique et de la détection automatique des phrases injonctives (e.g. phrases impératives) en dialogue oral spontané [26].

La conjonction de ces trois compétences (linguistique, informatique et traitement du signal) donne toute son originalité à cette approche multimodale qui vise à mesurer avec précision le rôle de la prosodie dans l'interprétation injonctive qui émerge au niveau des syntagmes et des phrases [26].

L'originalité de ce projet consiste à se saisir de données sonores authentiques et massives, en contournant l'enchevêtrement des paramètres par la conception d'outils capables d'opérer une discrimination automatique des emplois sur des bases statistiques [26].

1.4 Présentation du projet

Le signal de la parole est l'un des signaux les plus complexes à caractériser et à analyser, cette complexité est liée à sa production et à son aspect technologique. Le signal de la parole varie non seulement avec les sons prononcés, mais également avec le locuteur, l'âge, les émotions et l'environnement.

Dans le cadre de notre projet, on va s'intéresser à la classification des audio du projet RAVIOLI en classes injonctive et non injonctive.

L'injonction peut être définie comme un ordre ou un commandement précis non discutable, qui doit être obligatoirement exécuté et qui est souvent accompagné de menaces de sanctions. Dans ce contexte, l'impératif peut être considéré comme la manière typique d'exprimer l'injonction. Cependant, la forme impérative n'exprime pas uniquement la valeur injonctive et elle peut être utilisée à d'autres fins telles que l'expression de la condition ou d'une question, etc. Il en ressort qu'une classification automatique des injonctions dans les corpus oraux n'est pas une tâche simple car elle ne peut pas s'appuyer uniquement sur les travaux dédiés à la classification de la prosodie.

C'est là que l'apprentissage automatique et l'intelligence artificielle entrent en jeu pour la reconnaissance des actes des énoncés oraux et la classification des signaux audio vocaux en classes injonctive (INJ) et non injonctive (NINJ).

1.5 Objectif du projet

L'objectif de ce projet est de fournir des méthodes pour la classification des audio vocaux du projet RAVIOLI en classes injonctives et non injonctives.

Afin d'atteindre au mieux nos objectifs, nous organisons cela en différentes tâches :

- Fournir des outils et méthodes pour l'extraction et le traitement des données.
- Validation des résultats obtenus par Hacine-Gharbi et Ravier en 2020 par la méthode GMM en utilisant d'autres méthodes d'apprentissage automatique.
- Application des méthodes de classification sur une nouvelle base d'images.
- Mise en oeuvre d'une méthode d'apprentissage profond pour la classification.
- Confrontation des résultats "machine" avec des expertises linguistiques.

Ce travail permettra de disposer pour les linguistes d'un avis "machine" pour les injonctions au classement controversé.

Pur réaliser ces objectifs, nous allons analyser les bases de données à notre disposition et appliquer des méthodes d'apprentissage automatique tel que SVM et KNN et des méthodes d'apprentissage profond comme les réseaux de neurones récurrents.

Chapitre 2

Rappel et définitions

La parole est un atout que seul nous, êtres humains, possédons. La générer naturellement résulte d'une combinaison complexe de phénomènes physiques et d'interprétations psychoacoustiques. Nous allons définir les notions que nous utiliserons dans notre travail. Nous définissons la parole et le signal de la parole, ses modes de représentation et ses caractéristiques.

2.1 La parole

La parole humaine est un flux continu constitué d'une suite de mots, eux-mêmes étant constitués d'un enchaînement de phonèmes et de bruits articulatoires. La parole est très variable puisqu'un même phonème possède de nombreux paramètres qui sont fonction du locuteur : [28]

- intensité et hauteur de la voix.
- type de son émis par le locuteur (chuchotement, chant, parole).
- déformation du son dû à l'accent du locuteur.
- propriétés physio-acoustiques de l'appareil phonatoire du locuteur.
- émotion dans la voix du locuteur (serein, pleurant, en colère, gémissant,...).

2.2 Le signal

Un signal est l'observation des fluctuations ou des variations d'une grandeur physique (électrique, thermique, mécanique, chimique, lumineuse...) relative à un phénomène qui

joue le rôle de processus générateur. Le signal est donc un support physique d'une information, On distingue deux grands types de signaux : [28]

- **Le signal analogique** : est une variation continue d'une tension électrique en fonction du temps. Ces valeurs sont exprimées en volts et correspondent de façon analogique et proportionnelle à la forme d'onde de la source acoustique.
- **Le signal numérique** : est une variation discontinue de valeurs discrètes (distinctes les unes des autres) en fonction du temps. Ces valeurs sont exprimées sous forme de nombres (mots binaires 0 et 1) et constituent une représentation d'un signal analogique.

2.3 Le signal acoustique

Un signal acoustique (ou signal sonore) analogique est un signal continu qui représente un son. Lorsqu'on amplifie un tel signal pour exciter un haut-parleur, la membrane du haut-parleur oscille selon l'amplitude instantanée de ce signal. Cette vibration est transmise à l'air et produit un son qui se propage jusqu'à l'oreille.

La parole est un signal réel, continu, d'énergie finie et non stationnaire. Sa structure est complexe et variable avec le temps. Sa composition est la suivante [28] :

- Pseudo-périodique (D) : les sons voisés sont générés par vibration des cordes vocales, pseudo-périodiques ;
- Aléatoire (A) : les sons fricatifs sont représentés par des signaux permanents, non périodiques, contenant une grande bande de fréquence.
- Impulsionnel (C) : phase explosive des sons occlusifs produit par une ouverture soudaine laissant passer une bouffée d'air avec/sans vibration des cordes vocales. Ils sont représentés par des signaux impulsionnels, non périodiques, contenant une large bande de fréquences.
- le bruit (B).

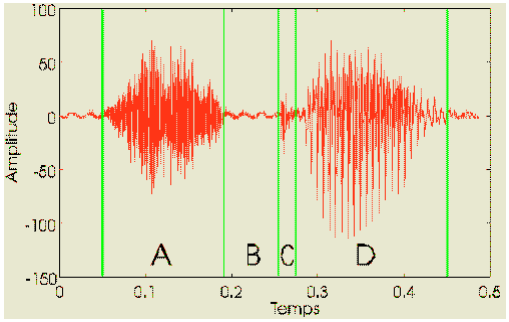

Fig 2.1 – Représentation temporelle du mot 'SKI'.[28]

2.3.1 Les modes de représentation d'un signal acoustique

L'étude du son a donné naissance à différents modèles de représentations ayant chacun un intérêt particulier [30] :

- **Représentation temporelle** : cette représentation montre l'évolution de l'intensité du signal sonore dans le temps.
- **Représentation fréquentielle (ou spectrale)** : représente l'intensité en fonction de la fréquence, cette représentation permet de visualiser la composition fréquentielle d'un son mais également l'intensité de chaque fréquence.
- **Représentation tridimensionnelle (le sonagramme ou spectrogramme)** : le temps est représenté sur l'axe X, la fréquence sur l'axe Y et le niveau de chaque fréquence, sur l'axe Z, est symbolisé par le niveau de gris. Il s'agit de la représentation temps-fréquence du son. On trace la répartition énergétique du son en fonction du temps et des fréquences. Le spectrogramme est très utilisé pour étudier le signal de parole.
- **Spectrogramme mel :** c'est un spectrogramme dans lequel les fréquences sont converties en échelle mel. La conversion d'hertz en mels se fait à l'aide de la formule ci-dessous où f représente la fréquence en Hz :

$$m = 2595.\log_{10}\left(1 + \frac{f}{700}\right)$$

La figure suivante représente les différents modes de représentation appliqués sur un signal.

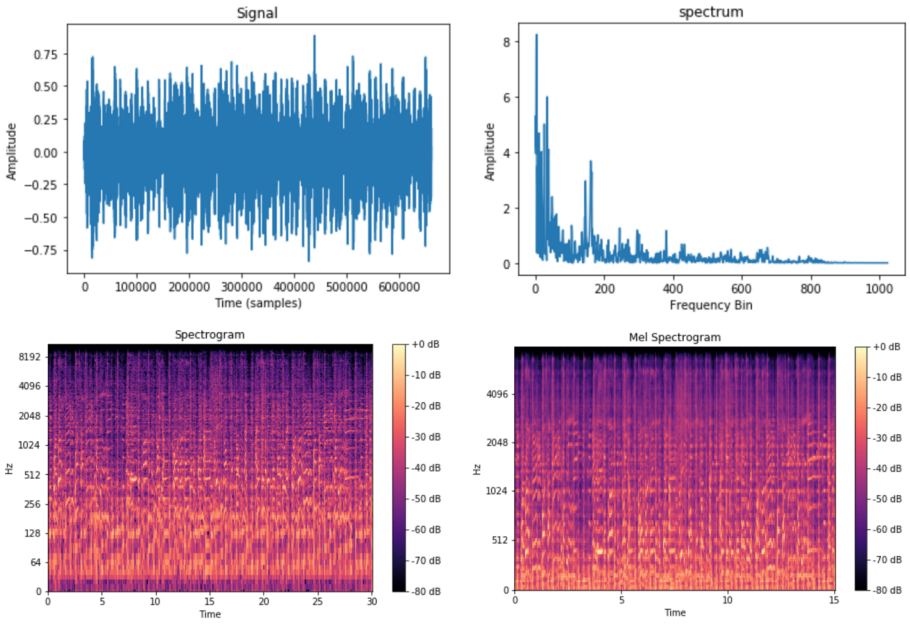

Fig 2.2 – Exemple d'une représentation temporelle, fréquentielle, spectrogramme et spectrogramme mel d'un signal [23]

2.3.2 Les caractéristiques du signal

Divers paramètres du signal de la parole ont été proposes en reconnaissance automatique du locuteur ou en classification. Ces paramètres doivent avoir une forte variabilité interlocuteurs et une faible variabilité intra-locuteur, permettant ainsi de discriminer plus facilement différents individus. De plus, ces paramètres doivent être robustes aux différents bruits et variations intersession. Plusieurs paramètres existent parmi lesquels on trouve [27] :

- **Les paramètres prosodiques** : les paramètres prosodiques peuvent impliquer de plus longs segments de parole (syllabe, mot, expression). Comme paramètres prosodiques, on trouve :
  - **l'intensité** qui dépend de l'amplitude de la vibration : plus elle est importante, plus le son est fort ; plus l'amplitude est faible, plus le son est faible. On l'exprime en décibel (dB) ;
  - **la fréquence** qui correspond au nombre de vibrations par seconde. Plus elle est élevée et plus le son paraîtra aigu, à l'inverse, il paraîtra grave, on la note f et elle a pour unité l'hertz (Hz). La fréquence la plus petite est appelée fréquence fondamentale (hauteur), elle se situe autour de :
    - 100 Hz chez l'homme.

- 200 Hz chez la femme.
- entre 200 et 300 Hz chez l'enfant. L'enfant a une voix beaucoup plus aiguë ; cela est dû au manque de maturité de ses cordes vocales ;

– **la période** qui est l'intervalle de temps au bout duquel la sinusoïde se reproduit à l'identique. Elle s'exprime en secondes et sa formule mathématique est : $T = \frac{1}{f}$ où f représente la fréquence en Hz.

– **le timbre** qui est défini par le nombre et l'intensité des harmoniques qui le compose et qui permet de reconnaître la personne qui parle ou l'instrument qui est joué.

– **l'énergie** qui est extraite directement du signal temporel sur une fenêtre d'analyse, sa variation est liee à l'intonation du locuteur. Elle est calculée par : $E = \sum_1^N s(n)^2$, généralement exprimée en decibels $E_{dB} = 10logE$.

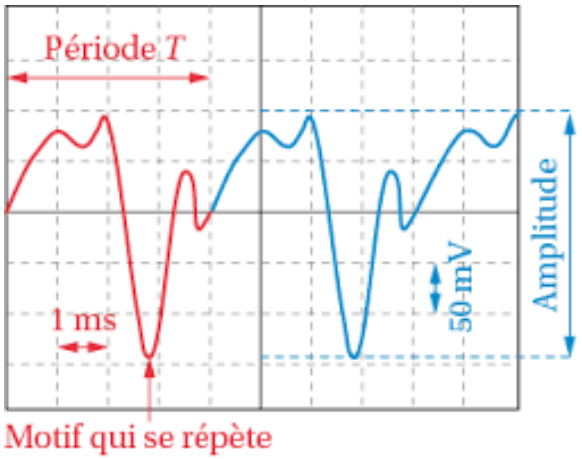

Fig 2.3 – Représentation schématique de la période et de l'apmplitude d'un signal. [35]

– **Les paramètres spectraux** : le signal de parole varie continuellement au cours du temps, en fonction des mouvements articulatoires. Par conséquent, sa paramétrisation doit être effectuée sur de courtes fenêtres d'analyse (typiquement de 10 à 30ms) où le signal est considéré comme quasi stationnaire. L'analyse utilise des fenêtres glissantes qui se recouvrent, à décalage régulier. Pour améliorer l'analyse et limiter les effets de bord, on pondère ensuite les trames du signal par une fenêtre temporelle aplatie aux extrémités, ce qui permet de réduire les discontinuités dans le signal. Différentes fenêtres ont été proposées : Hann, Blackman, Kaiser et Hamming. D'autres types de coefficients spectraux sont utilisés en reconnaissance automatique de la parole ou du locuteur [5] :

  - **MFCC** : les coefficients Cepstraux de Fréquence Mel sont basés sur le domaine fréquentiel en utilisant l'échelle Mel qui est basé sur l'échelle de l'oreille humaine. Ils accèdent aux informations qui caractérisent le conduit vocal.
  - **LPC** : le Code Prédictif Linéaire calcule un spectre de puissance du signal, il est utilisé pour l'analyse des formants. LPC est une méthode de codage et de représentation de la parole par un modèle de prédiction linéaire dont les coefficients sont calculés sur la base de la théorie de l'erreur quadratique moyenne la plus faible.
  - **PLP** : le modèle de Prédiction Linéaire Perceptive modélise la parole humaine en se basant sur le concept de la psychophysique de l'audition. La PLP élimine les informations non pertinentes de la parole et améliore ainsi le taux de reconnaissance de la parole.

  Le PLP est identique au LPC sauf que ses caractéristiques spectrales ont été transformées pour correspondre aux caractéristiques du système auditif humain.
- **Les paramètres d'énergie obtenus par l'opérateur TEO (Teager-energy-operator)** : selon des études expérimentales réalisées par Teager, la parole est produite par un flux d'air non linéaire dans le système vocal. Dans des conditions de stress, la tension musculaire du locuteur affecte le flux d'air dans le système vocal, ce qui produit le son. Par conséquent, des caractéristiques non linéaires de la parole sont nécessaires pour détecter la parole dans le son. L'opérateur TEO s'appuye sur l'idée que la parole est le processus de détection de l'énergie. Pour un signal temporel discret $x[n]$, la TEO est définie comme :

$$\Phi\{x[n]\} = x^2[n] + x[n-1]x[n+1][4]$$

En définitive, les caractéristiques de la parole peuvent être d'ordre qualitatives ou quantitatives. Les indicateurs des caractéristiques quantitatives sont obtenus à l'aide d'opérateurs répertoriés en trois catégories, à savoir les opérateurs de détection, d'analyse et de codage, comme le montre la figure ci-dessous :

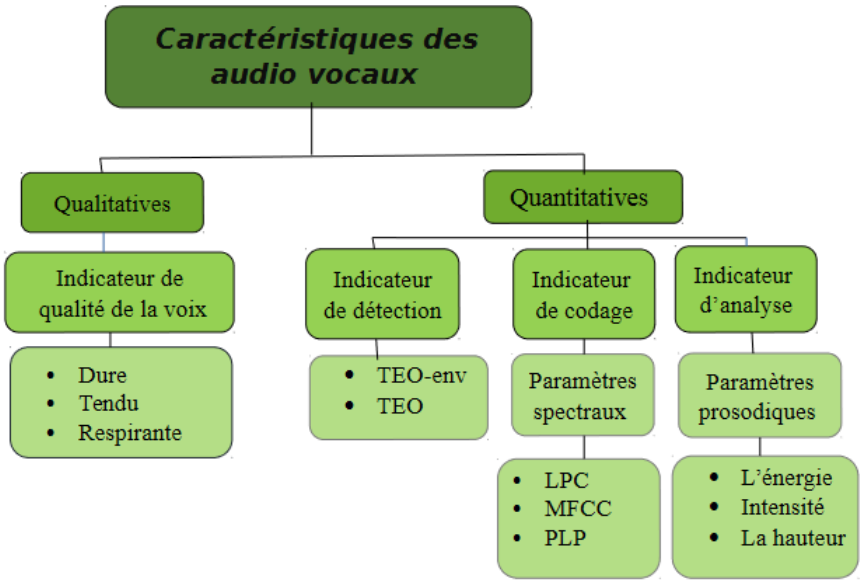

Fig 2.4 – Catégories des caractéristiques des audio vocaux

2.4 Conclusion

Dans ce chapitre, nous avons défini la parole et exposé les notions de base du signal vocal ainsi ses modes de représentation et ses caractéristiques.

Chapitre 3

Classification des signaux acoustiques

Cette partie constitue une phase importante dans la réalisation de notre projet. Dans cette partie, nous allons définir les différentes étapes utilisées pour récolter les données, ainsi les méthodes et les techniques exploitées pour le traitement et l'extraction de ces données. À la fin nous allons définir les différentes méthodes de classification que nous utiliserons pour classer les audio en classe injonctive et non injonctive.

3.1 Constitution des bases de données

Dans le cadre du projet RAVIOLI, une base d'énoncés injonctifs et non injonctifs a été collectée dans plusieurs environnements. Elle est caractérisée par une grande variabilité et complexité des signaux de parole spontanée (âge, sexe, locuteur, bruit, émotions, ...). Cette base de données est constituée d'énoncés injonctifs produits dans des interactions orales authentiques collectées à partir de la base de données ESLO2 (http://eslo.huma-num.fr/). [26]
L'ensemble des données est constitué des modules suivants :

- **éEole** : enregistrements divers au sein d'une école de l'agglomération orléanaise : cours, réunions, entretiens, repas, cours de récréation, déplacements (71h00).
- **Repas** : enregistrements dans un repas en famille ou entre amis réalisés par des locuteurs témoins (5h42).
- **Itinéraire** : demande d'itinéraire dans la rue. Le début est enregistré à micro discret, la suite à micro montré donne lieu à une reformulation de l'itinéraire (18h50).

– **24H** : enregistrement d'une journée entière d'un locuteur-auditeur, en utilisant un micro-cravate porté du lever au coucher par un témoin et captation des paroles entendues tout au long d'une journée (9h44).

Une fois les audio enregistrés, les linguistes appliqueront plusieurs opérations sur ces derniers :

– [•] Ils coupent les audio en sous audio en choisissant le point de coupure là où il y a du souffle ou il y a plus de parole,

- ils transcrivent les audio puis ils jugent s'ils sont exploitables et s'ils représentent des injonctions ou non,
- ils désignent la date de début et de fin ainsi que leurs durées,
- ils classent les audio injonctifs en différentes catégories : excuse, refus, apostrophe, stop, undefined, etc,
- ils identifient le locuteur dans chaque audio,
- les audio sont nommés par le nombre de secondes en premier puis par le numéro de la ligne du tableur suivi du nom de fichier son exemple : $0\_ligne6468\_ESLO2\_ECOLE\_1294$

La figure suivante représente la partie en-tête et quelques exemples du tableur fourni par les linguistes.

Nom du fichier	Locuteur	Injonctive?	Categorie	Start	End	Durée	Texte
24H_apresmiditravail_4.wav	NR390	oui	excuse	189.69	191.75	2.06	pardon je te le mets déjà sur les pieds je suis désolée
24H_apresmiditravail_4.wav	NR390	oui	apostrophe	346.9	347.33	0.43	Sarah
24H_apresmiditravail_4.wav	NR390	oui	excuse	390.97	391.86	0.89	pardon excusez -moi
24H_apresmiditravail_4.wav	NR390	oui	refus	498.53	500.93	2.4	bah vu comment il est cintré excuse -moi
24H_apresmiditravail_4.wav	collègue 4	oui	stop	551.36	552.15	0.79	attends attends attends attends attends
24H_apresmiditravail_4.wav	NR390	oui	__UNDEF__	574.24	574.45	0.21	non
24H_apresmiditravail_4.wav	NR390	oui	__UNDEF__	574.45	574.88	0.43	t' inquiète
24H_apresmiditravail_4.wav	collègue 4	oui	__UNDEF__	602.63	604.55	1.92	oui mais faut les serrer Pamela
24H_apresmiditravail_4.wav	collègue 4	oui	__UNDEF__	896.72	898.08	1.36	demain faut qu' on fasse les vitrines demain
24H_apresmiditravail_4.wav	NR390	oui	__UNDEF__	955.3	957.58	2.28	donc ça vous fait trente-neuf euros quatre-vingt-dix il vous plait
24H_apresmiditravail_4.wav	NR390	oui	__UNDEF__	984.32	985.71	1.39	voilà passez un bon week-end à bientôt
24H_apresmiditravail_4.wav	NR390	oui	__UNDEF__	1019.79	1023.04	3.25	après soit voilà vous échangez avec ce que vous voulez ou je vous fait un remboursement
24H_apresmiditravail_4.wav	cliente3	oui	__UNDEF__	1050.02	1054.82	4.8	non euh euh vous me faites un remboursement parce que là j' ai pas le temps de regarder sinon j' aurai pris autre chose
24H_apresmiditravail_4.wav	NR390	oui	__UNDEF__	1065.82	1067.93	2.11	profites -en bien hein pour une fois que tu as ton aprèm
24H_apresmiditravail_4.wav	NR390	oui	__UNDEF__	1095.03	1098.18	3.15	je vais juste vous demander de remplir cette petite partie avec une signature
24H_apresmiditravail_4.wav	NR390	oui	__UNDEF__	1198.82	1200.66	1.84	donc ça vous fait dix euros s' il vous plait
24H_apresmiditravail_4.wav	NR390	oui	__UNDEF__	1246.02	1247.1	1.08	oui donc du coup n' hésitez pas
24H_apresmiditravail_4.wav	NR390	oui	__UNDEF__	1247.1	1249.22	2.12	faudra juste bien garder l' étiquette sur le produit et le ticket
24H_apresmiditravail_4.wav	NR390	oui	__UNDEF__	1278.5	1281.39	2.89	donc ça vous fait cinquante-neuf euros quatre-vingt-quinze s' il vous plait
24H_apresmiditravail_4.wav	NR390	oui	__UNDEF__	1306.74	1308.98	2.24	voilà passez un bon week-end au revoir merci
24H_apresmiditravail_4.wav	NR390	oui	__UNDEF__	1396.0	1398.29	2.29	donc ça vous fait douze euros quatre-vingt-dix s' il vous plait
24H_apresmiditravail_4.wav	NR390	oui	__UNDEF__	1435.5	1436.97	1.47	passez un bon week-end à bientôt
24H_apresmiditravail_4.wav	NR390	oui	__UNDEF__	1445.27	1447.4	2.13	donc ça vous fait cinq euros quatre-vingt-quinze s' il vous plait
24H_apresmiditravail_4.wav	NR390	oui	__UNDEF__	1455.12	1455.93	0.81	Aliette
24H_apresmiditravail_4.wav	NR390	oui	__UNDEF__	1495.08	1497.2	2.12	donc cinq euros quatre-ving quinze s' il vous plait merci
24H_apresmiditravail_4.wav	NR390	oui	__UNDEF__	1568.99	1571.51	2.52	donc ça vous fait trente-quatre euros quatre-vingt-quinze s' il vous plait
24H_apresmiditravail_4.wav	NR390	oui	__UNDEF__	1571.97	1572.56	0.59	si excusez -moi
24H_apresmiditravail_4.wav	NR390	oui	__UNDEF__	1620.23	1621.1	0.87	je vous laisse vérifier
24H_apresmiditravail_4.wav	cliente11	oui	__UNDEF__	1752.03	1754.13	2.1	j' ai acheté un gilet tout à l' heure mais j' ai oublié de le passer
24H_apresmiditravail_4.wav	NR390	oui	__UNDEF__	1842.09	1843.65	1.56	donc ça vous fait quatre euros quatre-vingt-dix
24H_apresmiditravail_4.wav	NR390	oui	__UNDEF__	1900.89	1903.76	2.87	donc ça vous fait trente-quatre euros quatre-vingt-quinze s' il vous plait
24H_apresmiditravail_4.wav	NR390	oui	__UNDEF__	1903.76	1904.47	0.71	ah oui pardon
24H_apresmiditravail_4.wav	NR390	oui	__UNDEF__	1904.47	1905.53	1.06	excusez -moi

Fig 3.1 – Tableur des audio injonctifs.

Dans ce stage, on s'est basé sur le travail mené par Abdenour Hacine-Gharbi et al. financé par le projet RAVIOLI, dont l'objectif est le même (La classification des audio injonctifs et non injonctifs). [18] Nous allons dans un premier temps utiliser la

même base de données utilisée par les précédents auteurs, constituée d'un ensemble réduit d'injonctions et de non injonctions. Par la suite, nous travaillerons sur une deuxième base plus importante et plus complexe dans le but de valider les résultats trouvés avec la première base.

- **La première base** est la base utilisée par Hacine-Gharbi et al., les audio injonctifs de cette base sont constitués d'un discours spontané en français comprenant le mot "aller" (ou "allez"), qui est utilisé comme mot-clé pour classer les énoncés injonctifs. Cette base contient un total de 197 audio injonctifs et 198 audio non injonctifs.
- **La deuxième base** est la base qui contient des données sauvages (plusieurs locuteurs en même temps, bruit des fonds, non structurée), et ses audio injonctifs ne contiennent pas un mot-clé précis qui les désigne. Cette base contient 2237 audio injonctifs et 1215 audio non injonctifs.

3.2 Traitement de données

L'une des pierres angulaires de toute analyse est la qualité des données recueillies. Sans une compréhension de la qualité des données, il est difficile d'interpréter ou de faire confiance aux résultats qui en découlent. L'évaluation de cette qualité repose traditionnellement sur de la visualisation de données, notamment soutenue par des analyses statistiques.

Dans notre étude, les linguistes nous ont fourni une base de données qui contient 2237 audio injonctifs et 1215 audio non injonctifs. En examinant ces audio, on a remarqué que ces audio, malgré qu'ils ont été jugés exploitables, peuvent présenter beaucoup de difficultés comme par exemple, certains audio sont très courts, d'autres sont très bruités empêchant ainsi d'entendre les locuteurs, ou d'autres audio contiennent beaucoup de séquences de silence pouvant fausser les calculs, etc.

Pour ces raisons, nous avons appliquées pré-traitements à notre base, à savoir :

1. **échantillonnage des signaux**

   L'échantillonnage d'un signal continu est l'opération qui consiste à prélever des échantillons du signal pour obtenir un signal discret, c'est-à-dire une suite de nombres représentant le signal, dans le but de mémoriser, transmettre, ou traiter le signal.

La base de données vocales RAVIOLI a été enregistrée à l'origine à la fréquence d'échantillonnage de 44100 Hz. Nous avons sous-échantillonné les enregistrements à 16000 Hz en se basant sur le théorème de Shannon. Ce dernier exige que la fréquence d'échantillonnage doit être strictement supérieure à deux fois la plus grande fréquence présente dans le spectre du signal continu. Le contenu spectral étant quasiment absent au-delà de 8 kHz, il en résulte que toutes les informations présentes dans le signal original sont conservées dans les échantillons. L'objectif du sous échantillonnage est de réduire le coût de calcul.

Les audio de la base RAVIOLI sont composés de deux canaux qui représentent les sons entendus par les oreilles gauche et droite. On considérera les audio sous échantillonnés sur un seul canal comme la moyenne entre les deux canaux.

2. **Suppression des silences**

   En vérifiant les audio (soit INJ, NINJ) nous avons remarqué qu'il existait des audio contenant des trames dans les morceaux qui ne présentent pas de paroles. Afin de supprimer ces trames, on a utilisé deux méthodes différentes :

   – La détection d'activité vocale (en anglais Voice Activity Detection (VAD)) est définie comme étant le processus qui consiste, à partir d'un signal sonore, à différencier les portions qui contiennent de la voix de celles qui n'en contiennent pas.

     Le VAD découpe le signal audio d'entrée en trames. Une trame se définit comme une portion de signal de longueur fixe et de l'ordre de quelques millisecondes. Il va calculer pour chaque trame la probabilité de présence de la parole.

     À titre d'exemple, un résultat de VAD pour un extrait de parole est présenté dans la figure ci-dessous :

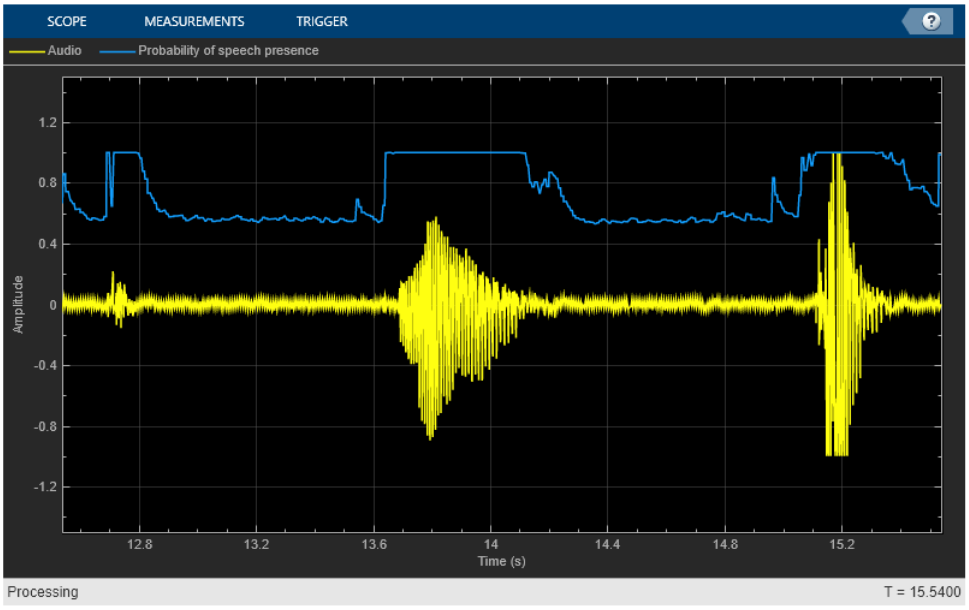

Fig 3.2 – Exemple des résultats obtenu par VAD.

– Méthode basée sur la hauteur : nous avons calculé la fréquence fondamentale pour chaque audio. Ensuite nous avons implémenté une fonction sous python qui nous permet de parcourir toutes les trames de l'audio et de supprimer celles où la valeur de la fréquence fondamentale est nulle.

3.3 Extraction de données

Un signal audio peut être observé grâce à sa forme d'onde, décrivant les variations d'amplitude de l'onde acoustique au cours du temps. Cependant, cette représentation brute ne permet pas une caractérisation suffisante pour pouvoir être utilisée par la suite en classification.

De ce fait, l'objectif de cette partie est d'extraire les descripteurs ou attributs, mettant en évidence certaines propriétés du signal jugées pertinentes pour permettre de discriminer les différentes classes audio. Nous avons découpé le signal en segments élémentaires non-recouvrantes (pas d'intersections), de taille de 10 millisecondes. Ce sont sur ces trames que sont extraits les descripteurs.

Comme nous avons précédemment mentionné, ce travail est la suite du travail réalisé par Abdenour Hacine-Gharbi [18]. L'étude précédente a permis de calculé 114 caractéristiques de la première base de données décrite dans le paragraphe 3.1, ces caractéristiques sont :

- les caractéristiques prosodiques statiques (l'énergie logarithmique (E), hauteur(PI)) et les caractéristiques dynamiques (la vitesse (D), accélération (A)) associées à chacune des caractéristiques prosodiques.
- les caractéristiques acoustiques : un ensemble de 12 caractéristiques est calculé pour chaque type : LPCC, MFCC et PLP en plus des caractéristiques dynamiques (A et D), ce qui donne des vecteurs de caractéristiques de 36 composantes pour chaque type.

D'après cette étude menée sur la première base, les auteurs ont montré que les caractéristiques prosodiques sont pertinentes pour la tâche de classification en classes injonction / non-injonction. Ils ont aussi conclu que les caractéristiques spectrales utilisées pour les tâches de reconnaissance automatique de la parole ne sont pas du tout appropriées.

C'est donc pour ces raison que pour cette étude nous allons nous intéresser seulement aux caractéristiques prosodiques et écarter le reste.

Afin d'extraire les différentes caractéristiques prosodiques du signal de la parole, nous avons implémenté un programme sur Matlab qui se base sur l'exécution des logiciels Praat et HTK.

- **Praat** : est un logiciel libre sous licence GPL destiné à manipuler, traiter, synthétiser des sons vocaux (phonétique) et réaliser des analyses acoustiques et prosodiques (spectrogramme, formants, intonation, intensité). Il est multiplateforme et écrit en language C, et sa configuration est basée sur des "objets" où chaque objet (son, courbe intonative, matrice...) peut être chargé, créé, enregistré, interrogé, modifié ou utilisé pour créer un nouvel objet. [24]
- **HTK** (Hidden Markov Model Toolkit) : est une boîte à outils logicielle propriétaire pour la gestion HMM. Il est principalement destiné à la reconnaissance de la parole, mais a été utilisé dans de nombreux autres domaines de la reconnaissance de formes pour des applications qui utilisent des HMM, y compris la synthèse de discours ou la reconnaissance de caractères et le séquençage ADN. HTK consiste en un ensemble de modules de bibliothèque et d'outils disponibles sous forme de source C. Il est largement utilisé par les chercheurs qui travaillent sur les HMM. [25]

Pour le calcul de ces caractéristiques, nous avons utilisé les logiciels Praat et HTK qui convertissent chaque audio en séquences de vecteurs de caractéristiques (la longueur de chaque séquence est fixé à $10ms$).

La figure suivante représente les valeurs de la fréquence fondamentale calculées par le logiciel Praat.

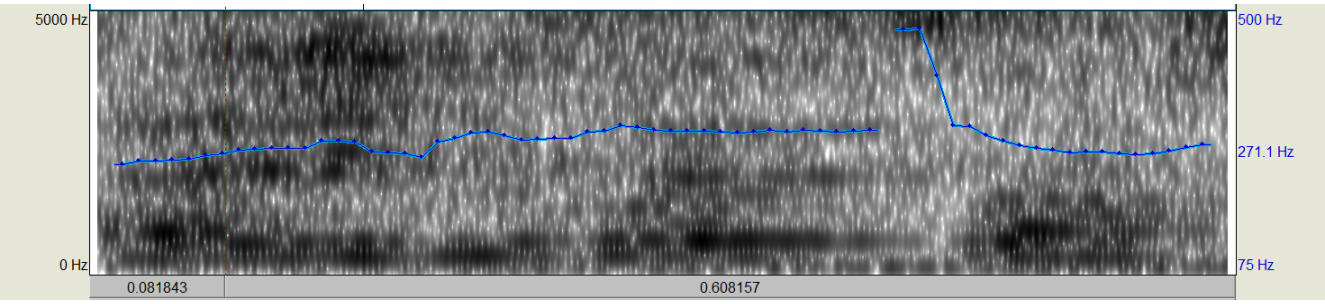

Fig 3.3 – Exemple des valeurs de la fréquence fondamentale obtenus par Praat.

Les valeurs de l'énergie et les valeurs des caractéristiques dynamiques (D et A) sont extraites par le logiciel HTK, en utilisant la commande 'Hcopy' de la bibliothèque HTK-Tools, cette commande prend ses options dans un fichier de configuration.

Le fichier de configuration utilisé pour le calcul de la vitesse et l'accélération de PI est le suivant :

```
TARGETKIND      = USER_D_A
```

Pour le calcul de E et ses caractéristiques dynamiques (D et A) nous avons utilisé le fichier de configuration ci-dessous. Ce dernier montre que les valeurs de E sont calculées à partir des fichiers 'wav' toutes les $10ms$ et sur des fenêtres d'analyse de Hamming toutes les $30ms$.

```
SOURCEKIND = WAVEFORM
SOURCEFORMAT = WAV
TARGETKIND = E_D_A
TARGETRATE = 100000.0 %Cibles de 10 ms
WINDOWSIZE = 300000.0 %Fenetre de 30 ms
SOURCERATE = 625.0 %Frequence d echantillonnagede 16KHz, en 100ns
NATURALREADORDER = F %Activer l'ordre de lecture
                     %naturel pour les fichiers HTK
NATURALWRITEORDER = F %Activer l ordre de l ecriture
                      %naturel pour les fichiers HTK
ZMEANSOURCE = T %Forme d onde source a moyenne nulle avant analyse
USEHAMMING = T %Utilise une fenetre hamming
PREEMCOEF = 0.97 %Premephasage de premier ordre
USEPOWER = T %Utiliser la puissance et non la magnitude dans
             %l analyse fbank
NUMCHANS = 22 %Filtrage sur 22 canaux
ENORMALIS = F %Normalisation de l intensite des donnees
```

3.4 Classification

Une fois qu'un sous-ensemble d'attributs a été sélectionné pour décrire le problème, un classifieur peut alors être modélisé afin de faire le lien entre les descripteurs et les classes cibles.

La classification est une étape importante pour l'analyse de données. Elle consiste à regrouper les objets d'un ensemble de données en classes homogènes. Il existe deux types d'approches : la classification supervisée et la classification non supervisée. Ces deux approches se différencient par leurs méthodes et par leur but.

3.4.1 Classification supervisée

La classification supervisée est basée sur un ensemble d'instances (appelé ensemble d'apprentissage) de classes connues, le but est de construire à partir de l'ensemble d'apprentissage un modèle qui permette de prédire la classe des nouvelles instances. Ceci revient à découvrir la structure des classes afin de pouvoir généraliser cette structure sur un ensemble de données plus large.

D'après le travail suscité de Abdenour Hacine-Gharbi (Hacine-Gharbi et Ravier 2020), les auteurs se sont intéressés à la contribution des descripteurs prosodiques pour la tâche de classification des injonctions vocales. Cette étude a porté sur des caractéristiques élaborées à la main, telles que les descripteurs prosodiques (hauteur, énergie) et d'autres

descripteurs typiques, tels que les caractéristiques spectrales : LPCC, PLP et MFCC avec leurs dérivées premières et secondes pour la modélisation dynamique. La tâche de classification a été réalisée par les modèles de mélange gaussien (GMM). Ce dernier a atteint le meilleur taux de classification de $86,22\%$ sur un sous-ensemble de 395 énoncés de la base de données RAVIOLI (197 énoncés INJ et 198 énoncés NINJ) en utilisant 8 gaussiennes et trois caractéristiques prosodiques (E, E_D, PI_D). Les auteurs ont montré aussi la prédominance de la caractéristique d'énergie logarithmique et une légère amélioration a été obtenue en utilisant d'autres caractéristiques dynamiques d'énergie et de hauteur.

Afin d'atteindre l'un des objectifs de notre projet, il est primordial de répondre à cette interrogation : est-ce-que les résultats obtenus par Abdenour Hacine-Gharbi et al. dépendent du classifieur utilisé ? Pour pouvoir répondre à cette question, nous avons étudié d'autres classificateurs tels que SVM (Support Vector Machine) et KNN (k Nearest Neigbors).

Plusieurs auteurs ont montré l'efficacité des algorithmes KNN et SVM pour la classifications des audio dont on peut citer :

- Tsang-Long et al.(PaoYu, ChenJun, YehYun, ChengYu, and Lin 2007) ont utilisé le KNN pour reconnaître cinq émotions, dont la colère, la joie, la tristesse, le neutre et l'ennui, à partir de discours émotionnels en mandarin. [1]
- Szymon et Bozena (2020) ont récupéré les descripteurs de la parole qui peuvent être utiles pour la classification des émotions dans le chant afin de les classer à l'aide d'un machine à vecteurs de support (SVM).[17]
- Ling He et al.(Margaret Lech,Namunu C.Maddagea, Nicholas B.Allenb 2011) ont utilisé les méthodes KNN et GMM pour la classification automatique du stress et des émotions dans la parole. Ces méthodes ont été testées sur deux bases de données contenant de la parole naturelle, SUSAS (annotée avec trois niveaux de stress différents) et les données de l'Oregon Research Institute (annotées avec cinq émotions différentes : neutre, en colère, anxieux, dysphorique et heureux). [3]

Dans cette étude, nous nous intéresserons à la classification supervisée car nos données sont étiquetées. De nombreux algorithmes d'apprentissage sont adaptés au problème de la classification supervisée et nous allons utiliser les machines à vecteurs de support (SVM) et les méthodes des K plus proches voisins (KNN).

3.4.1.1 Machines à vecteurs supports (SVMs)

La technique des machines à vecteurs supports (SVM) est une méthode d'apprentissage supervisé introduite par Vladimir Vapnik au début des années 90 qui a connu, cette dernière décennie, un grand développement en théorie et en application.

Les SVMs sont des classifieurs qui reposent sur deux idées clés : la notion de marge maximale et la notion de fonction noyau. La notion de marge maximale consiste à trouver l'hyperplan qui sépare les exemples positifs des exemples négatifs, en garantissant que la distance entre la frontière de séparation et les échantillons les plus proches (marge) soit maximale. Ces derniers sont appelés vecteurs supports.

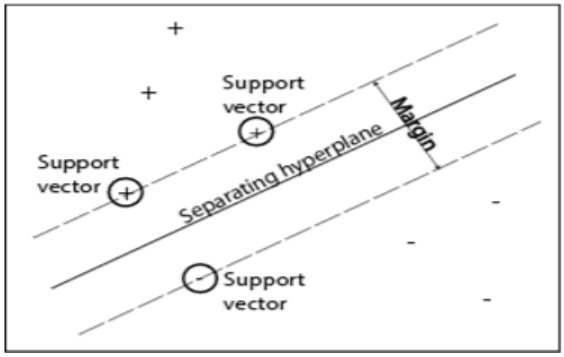

Fig 3.4 – Principe de la méthode SVM [36]

Dans le cas où les données ne sont pas linéairement séparables, les SVMs utilisent la technique des noyaux pour transformer l'espace de représentation des données d'entrées en un espace de plus grande dimension appelée espace de caractéristiques (possiblement de dimension infinie), dans lequel il est probable qu'il existe une séparatrice linéaire.

Cette fonction noyau ne nécessite pas la connaissance explicite de la transformation à appliquer pour le changement d'espace. Les fonctions noyau permettent de transformer un produit scalaire dans un espace de grande dimension, ce qui est coûteux, en une simple évaluation ponctuelle d'une fonction.

Les noyaux les plus utilisés pour les SVMs sont les noyaux polynomiaux, sigmoïdaux et à fonction de base radiale (Radial Basis Function,RBF).

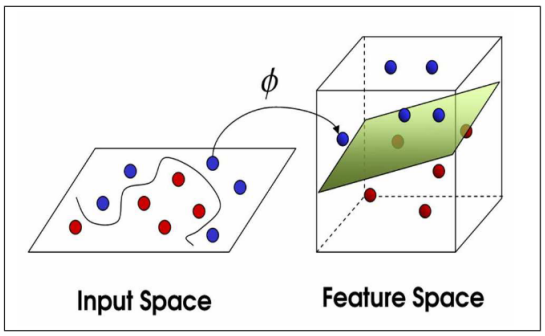

Fig 3.5 – Exemple de fonction noyaux SVM [36].

3.4.1.2 Algorithme des k plus proches voisins(KNN)

L'algorithme des K plus proches voisins en anglais (K Nearest Neighbors KNN) est un algorithme d'apprentissage supervisé basé sur la notion de proximité (voisinage) entre les instances et sur l'idée de raisonner à partir de cas similaires pour prendre une décision.

Le principe de base de l'algorithme KNN est de trouver les K instances les plus proches en calculant la similarité entre l'instance à classer et les instances déjà classées (les instances de la base d'apprentissage). On trie les instances par leurs distances et on prend les K premiers. Ensuite on cherche la classe majoritaire parmi les k instances les plus proches et affecter à instance non classée.

- ▷ **Distance euclidienne** : calcule la racine carrée de la somme des différences carrées entre les coordonnées de deux points :

$$D_e(x, y) = \sqrt{\sum_{j=1}^{n}(x_j - y_j)^2}$$

- ▷ **Distance Manhattan** : calcule la somme des valeurs absolues des différences entre les coordonnées de deux points :

$$D_m(x, y) = \sum_{j=1}^{n}|x_j - y_j|$$

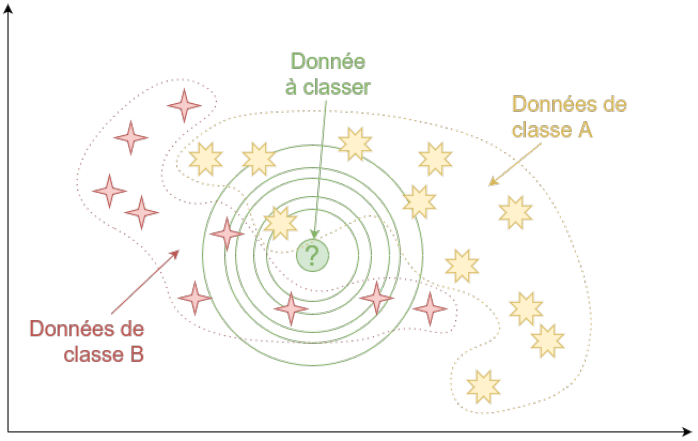

Fig 3.6 – Principe de la méthode KNN [21]

Algorithm 1: Les K plus proches voisins (KNN) [15]

1 **Initialisations :**

- D : un ensemble de données.
- d : une fonction distance.
- K : un entier.

Traitements

Pour une nouvelle observation X dont on veut prédire sa variable de sortie y

Faire :

- Calculer toutes les distances de cette observation X avec les autres observations du jeu de données D.
- Retenir les K observations du jeu de données D les proches de X en utilisation la fonction de calcul de distance d.
- Prendre les valeurs de y des K observations retenues :
  - [1.] Si on effectue une régression, calculer la moyenne (ou la médiane) de y retenues.
  - [2.] Si on effectue une classification , calculer le mode de y retenues.

  Retourner la valeur calculée dans l'étape (3) comme étant la valeur qui a été prédite par K-NN pour l'observation X.

Cette méthode dépend des éléments suivants :

- le nombre de voisins retenus (K) : on le prend comme un nombre impair.
- la distance entre deux instances : la distance euclidienne est généralement utilisée

lorsque les attributs sont numériques.

3.4.1.3 Modèle de melange de Gaussiennse (GMM)

Le modèle de mélange de Gaussiennes est un modèle statistique où la distribution des données est un mélange de plusieurs lois Gaussiennes. Le GMM est le modèle de référence en reconnaissance du locuteur.

Dans le cadre d'un modèle de mélange de gaussiennes GMM, $p(x|C_i)$ est modélisé par une combinaison linéaire de lois normales. On définit la densité de probabilité de la variable x appartenant à la classe C_i, où x est un vecteur de dimension D, suivant un modèle GMM d'ordre M par [6] :

$$p(x|C_i) = \sum_{m=1}^{M} \pi_m \mathcal{N}(x|\mu_m, \Sigma_m)$$

où

$$\mathcal{N}(x|\mu_m, \Sigma_m) = \frac{1}{(2\pi)^{D/2}|\Sigma_m|^(1/2)} \exp \frac{1}{2}(x-\mu_m)^T \Sigma_m^{-1}(x-\mu_m)$$

et

$$\sum_{m=1}^{M} \mu_m = 1, \mu_m \geq 0$$

avec

- $\mathcal{N}(x|\mu_m, \Sigma_m)$: appelée composante du mélange.
- μ_m : vecteur moyenne de dimension D.
- Σ_m : la matrice de covariance de dimension $D\ddot{O}D$ et $|\Sigma|$ est le déterminant de Σ.
- π_m : les coefficients de mélange et représentent la probabilité a priori que x soit généré par m^{eme} composante du modèle.

L'estimation de l'ensemble des paramètres $\lambda = \{\mu_m, \Sigma_m, \pi_m\}_{m=1}^{M}$, en se fait de manière itérative à l'aide de l'algorithme Espérance-Maximisation (EM, Expectation Maximization). Le choix du nombre de composantes M se fait généralement de manière empirique. De plus, la matrice de covariance peut être choisie comme pleine, diagonale ou sphérique. Ce choix est également à la charge de l'expérimentateur.

3.4.2 Les Réseaux de neurones artificiels

Les réseaux de neurones artificiels font partie des techniques d'apprentissage profond qui sont particulièrement adaptées au traitement des problèmes de régression ou de classification des données. Ils visent à reprendre le fonctionnement du neurone biologique.

Dans un réseau de neurones, plusieurs algorithmes travaillent ensemble pour effectuer des calculs sur les données d'entrée afin de produire une donnée de sortie. Ces données de sortie peuvent également aider le réseau de neurones à apprendre et à améliorer leur précision [31].

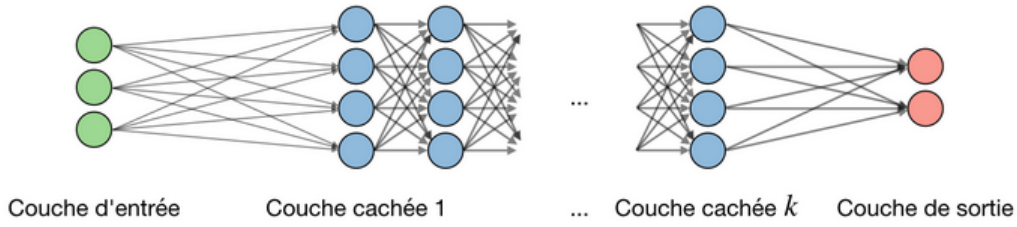

Fig 3.7 – Réseau de neuronnes [33].

Le réseau de neurones est composé des composants principaux suivants :

- **neurones** : Les réseaux de neurones sont constitués d'un ensemble de neurones (nœuds) connectés entre eux par des liens qui permettent de propager les signaux de neurone à neurone [31].
- **couches** : (ou layers) contiennent des neurones et aident à faire circuler l'information. Il existe au moins deux couches dans un réseau de neurones : la couche d'entrée (input layer) et la couche de sortie (output layer). Les couches, autres que ces deux couches, sont appelées les couches cachées (ou hidden layers). Plus il y aura de couches cachées, plus le réseau sera profond (deep learning) [31].
- **poids et biais** : sont des variables du modèle qui sont mises à jour pour améliorer la précision du réseau. Un poids est appliqué à l'entrée de chacun des neurones pour calculer une donnée de sortie [31].
- **fonctions d'activation** : lissent ou normalisent la donnée de sortie avant qu'elle ne soit transmise aux neurones suivants. Ces fonctions aident les réseaux de neurones à apprendre et à s'améliorer. Pour ce faire, la fonction d'activation doit être non linéaire car elle permet de séparer des données non linéaires. Les fonctions linéaires ne fonctionnent qu'avec une seule couche de neurones. Les fonctions non linéaires les plus utilisées sont dans la figure suivante[31] :

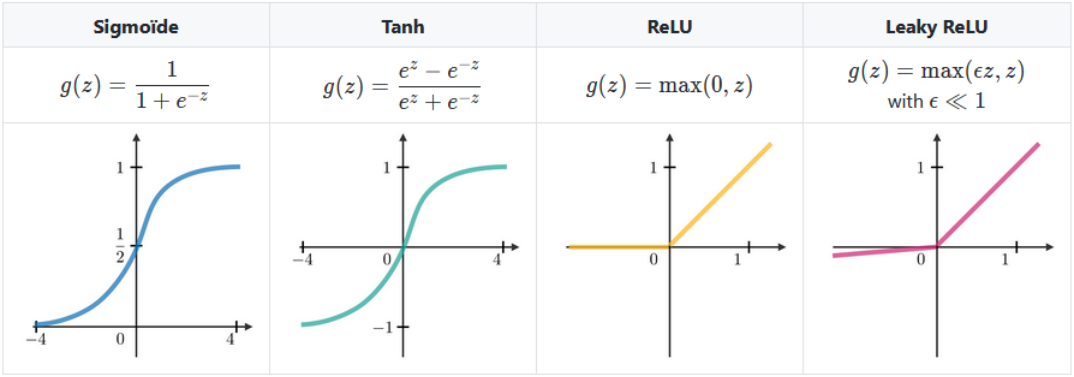

Fig 3.8 – Les principales fonctions d'activation [33].

- **fonction de perte :** est une méthode permettant d'évaluer la qualité de la modélisation des données par un algorithme spécifique. Si les prédictions s'écartent trop des résultats réels, la fonction de perte en crachera un très grand nombre. Elle prend en compte deux paramètres : la sortie prédite et la sortie réelle. Les fonctions de perte les plus utilisées actuellement en classification sont : entropie croisée et la marge maximal. [14].
- **Les optimiseurs** : mettent à jour les paramètres de poids afin de minimiser la fonction de perte qu'indique à l'optimiseur quand il va dans la bonne ou la mauvaise direction. Le taux d'apprentissage : il s'agit d'un très petit nombre, par lequel nous multiplions les gradients pour les mettre à l'échelle. En termes mathématiques, des taux d'apprentissages trop importants peuvent signifier que l'algorithme ne convergera jamais vers un optimum et des taux trop petites peuvent conduire notre optimiseur à converger vers un minimum local pour la fonction de perte, mais pas vers le minimum absolu. Les optimiseurs les plus utilisés sont [12] :
  - **la descente de gradient** : est l'algorithme d'optimisation le plus basique mais le plus utilisé en apprentissage automatique pour réduire la fonction de coût. Cette méthode a été utilisée dans de nombreux algorithmes en fonction de leurs spécificités : descente de gradient par lot, stochastique (SGD), mini-batch,etc. La descente de gradient est rapide, efficace, robuste et surtout flexible.
  - **momentum** : permet d'accélérer la descente de gradient (GD) lorsque nous avons des surfaces qui se courbent plus fortement dans une direction que dans une autre. Pour la mise à jour des poids, au lieu d'utiliser uniquement le gradient de l'étape actuelle pour guider la recherche, le momentum accumule éga-

lement le gradient des étapes passées pour déterminer la direction à prendre.

- **adaptive moment estimation (Adam)** utilise également le concept de momentum en ajoutant des fractions des gradients précédents au gradient actuel. Cet optimiseur est devenu assez répandu et son utilisation est pratiquement acceptée pour la formation de réseaux de neurones.
- **Adagrad** adapte le taux d'apprentissage spécifiquement aux caractéristiques individuelles ; cela signifie que certains des poids de l'ensemble de données auront des taux d'apprentissage différents des autres. Cela fonctionne très bien pour les ensembles de données éparses où beaucoup d'exemples d'entrée sont manquants.

- **Dropout** : est une technique qui est destinée à empêcher le sur-ajustement sur les données de training en abandonnant des unités dans un réseau de neurones. En pratique, les neurones sont soit abandonnés avec une probabilité p ou gardés avec une probabilité $1 - p$. [33]

Un réseau de neurones classique permet de gérer les cas ou les expériences sont indépendantes les unes des autres. Lorsque les expériences sont sous la forme de séquences temporelles, une nouvelle structure a été inventée : les réseaux de neurones récurrents. Cette nouvelle structure introduit un mécanisme de mémoire des entrées précédentes qui persiste dans les états internes du réseau et peut ainsi impacter toutes ses sorties futures.

3.4.2.1 Les réseaux récurrents (RNN)

Les réseaux récurrents (RNN) sont des réseaux de neurones dans lesquels l'information peut se propager dans les deux sens, notamment des couches profondes vers les couches précoces. Les (RNN) appartiennent aux algorithmes les plus prometteurs du moment car ils sont les seuls à avoir une mémoire interne. En raison de leur mémoire interne, les RNN sont capables de se souvenir de choses importantes concernant les données qu'ils ont reçues, ce qui leur permet d'être très précis dans la prédiction de ce qui va suivre. C'est la raison pour laquelle ils sont les algorithmes préférés pour les données séquentielles comme les séries temporelles, la parole, audio, vidéo, météo, etc [13].

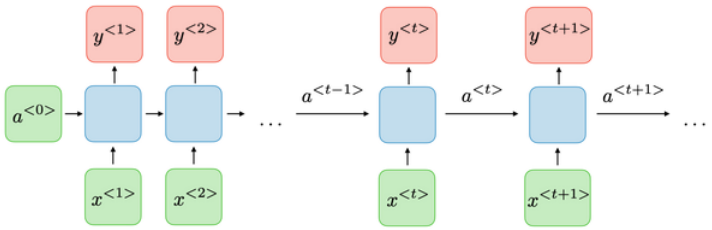

Fig 3.9 – Réseau de neurones récurrent[34].

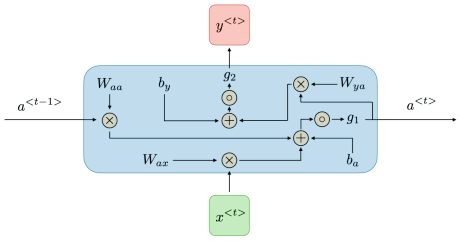

Fig 3.10 – Cellule d'un RNN[34].

Pour chaque pas de temps t, le vecteur de couche cachée $a^{<t>}$ et le vecteur de sortie $y^{<t>}$ sont exprimés comme suit [34] :

$$a^{<t>} = g1(W_{aa}a^{<t-1>} + W_{ax}x^{<t>+b_a}) \text{ et } y^{<t>} = g2(W_{ya}a^{<t>} + b_y)$$

avec :

- W_{aa}, W_{ya}, W_{ax} : matrices de poid.
- g_1, g_2 : fonctions d'activation.

Les réseaux neuronaux récurrents souffrent d'une mémoire à court terme. Si une séquence est suffisamment longue, ils auront du mal à transmettre les informations des étapes précédentes aux étapes suivantes. Pour résoudre ce problème, différents types de cellules de mémoire à long terme ont été introduits. La plus populaire de ces cellules de mémoire à long terme est la cellule LSTM.

3.4.2.2 Les réseaux longue mémoire à court terme (LSTM)

Un réseau longue mémoire à court terme (Long short-term memory (LSTM)) est un type unique de réseau neuronal récurrent (RNN) capable d'apprendre des dépendances à long terme, ce qui est utile pour certains types de prédictions qui nécessitent que le réseau retienne des informations sur de longues périodes, une tâche que les RNN traditionnels ont du mal à accomplir [13].

Les LSTM contiennent des informations en dehors du flux normal du réseau récurrent dans une cellule à porte. Grâce à cette cellule, le réseau peut manipuler l'information de plusieurs façons.

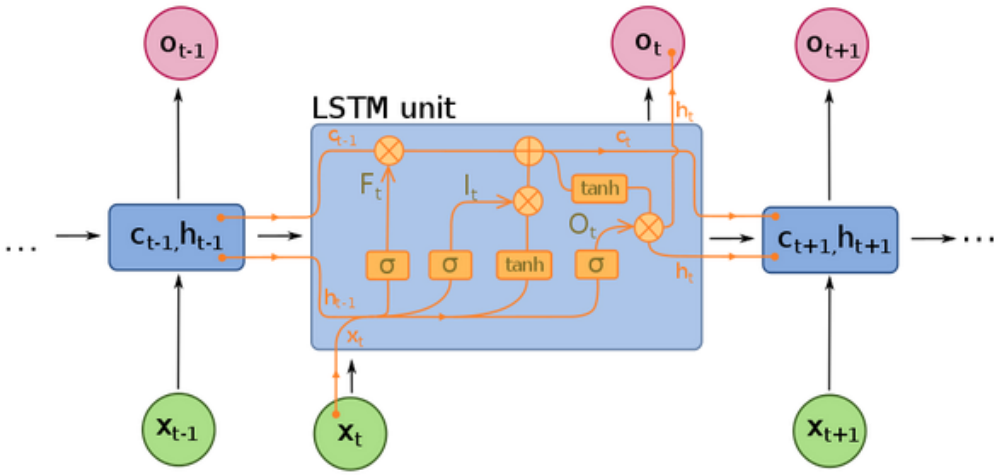

Fig 3.11 – Représentation d'une cellule LSTM [22].

Nous avons trois portes différentes qui régulent le flux d'information dans une cellule LSTM. Une porte d'oubli, une porte d'entrée et une porte de sortie. [22]

- **La porte d'entrée (i)** : décide quelle nouvelle information nous allons stocker dans l'état de la cellule.
- **La porte de sortie (o)** : détermine quelles parties de l'état de la cellule sont sorties.
- **La porte d'oubli (f)** : supprime les informations qui ne sont plus nécessaires à la réalisation de la tâche.

Les équations suivantes résument le fonctionnement d'une cellule LSTM à l'instant t [22]

$$
\begin{aligned}
F_t &= \sigma(W_F x_t + U_F h_{t-1} + b_F) \\
I_t &= \sigma(W_I x_t + U_I h_{t-1} + b_I) \\
O_t &= \sigma(W_O x_t + U_O h_{t-1} + b_O) \\
c_t &= F_t \otimes c_{t-1} + I_t \otimes \tanh\left(W_c x_t + U_c h_{t-1} + b_c\right) \\
h_t &= O_t \otimes \tanh\left(c_t\right) \\
o_t &= f(W_o h_t + b_o)
\end{aligned}
$$

avec :

- W_F, W_I, W_O, W_c : sont les matrices de poids de chacune des quatre couches pour leur connexion au vecteur d'entrée x_t.
- U_F, U_I, U_O, U_c : sont les matrices de poids de chacune des quatre couches pour leur connexion au vecteur d'entrée h_{t-1}

– b_F, b_I, b_O, b_c : sont les termes de biais pour chacune des quatre couches.

3.4.2.3 Les réseaux neuronaux convolutifs (CNN)

Les réseaux de neurones convolutifs (CNN en anglais Convolutional Neural Network) sont à ce jour les modèles les plus performants pour classer des images. Ils comportent deux parties bien distinctes. En entrée, une image est fournie sous la forme d'une matrice de pixels. Elle a deux dimensions pour une image aux niveaux de gris. La couleur est représentée par une troisième dimension, de profondeur 3 pour représenter les couleurs fondamentales [Rouge, Vert, Bleu]. Le CNN contient plusieurs couches différentes : [20]

– couche d'entrée : contient la matrice tridimensionnelle qui représente l'image d'entrée.
– couche de convolution : la composante clé des réseaux de neurones convolutifs. Son but est de repérer la présence d'un ensemble de features dans les images reçues en entrée.
– couche de pooling : ce type de couche est souvent placé entre deux couches de convolution. elle reçoit en entrée plusieurs feature maps, et applique à chacune d'entre elles l'opération de pooling qui consiste à réduire la taille des images, tout en préservant leurs caractéristiques importantes.
– couche relu : joue le rôle de fonction d'activation elle remplace toutes les valeurs négatives reçues en entrées par des zéros.
– couche entièrement connectée (Fully connected) : Une couche entièrement connectée implique des poids, des biais et des neurones. Il est utilisé pour classer les images entre différentes catégories par formation.
– couche Softmax/logistique : est la dernière couche de CNN. La logistique est utilisée pour la classification binaire et Softmax est pour la multi-classification.
– couche de sortie : contient l'étiquette de CNN.

3.4.2.4 État de l'art sur les méthodes d'apprentissage profond

De nombreux articles ont utilisé les réseaux neuronaux convolutifs (CNN) et les réseaux réseaux longue mémoire à court terme (LSTM) pour la classification des audio pour un objectif proche du notre à savoir la classification des émotions.

Harar et al. (Harár, Burget et Dutta 2017) ont appliqué un CNN profond directe-

ment sur les fichiers wav après l'élimination du silence et la normalisation, et ont obtenu un taux de reconnaissance élevé dans l'expérience de la base de données EMODB avec trois catégories d'émotions (en colère, neutre et triste). [8]

Lim et al. (Lim, Jang, and Lee 2017) ont proposé une architecture basés sur un CNN et un LSTM concaténés. Les auteurs ont transformé le signal vocal en représentation 2D qu'est analysé par des CNN et des architectures de mémoire à long terme (LSTM). Le CNN est composé de deux couches convolutionnelles et d'une couche de max-pooling. Ensuite, deux couches LSTM supplémentaires empilées séquentiellement avec 1024 nœuds chacune, apprennent séquentiellement les caractéristiques des CNN.[9]

Les auteurs de (J. Zhao, Mao et Chen 2019) ont proposé un algorithme basé sur CNN et LSTM. Un réseau CNN LSTM 1D et un réseau CNN LSTM 2D, ont été construits pour apprendre les caractéristiques locales et globales liées aux émotions à partir de la parole et du spectrogramme log-mel respectivement. Les deux réseaux ont la même architecture, consistant en quatre CNN et une couche LSTM. Le réseau CNN LSTM 2D se concentre sur la capture des corrélations locales et des informations contextuelles globales du spectrogramme, tandis que le réseau CNN LSTM 1D peut apprendre les caractéristiques locales en fonction de sa dynamique temporelle. [16]

3.5 Conclusion

Dans ce chapitre, nous avons présenté les différentes bases de données du projet RAVIOLI que nous allons utiliser dans notre étude ainsi leurs contenus et les différentes méthodes utilisées durant leurs traitements. Nous avons aussi exposé les méthodes et les logiciels utilisés pour l'extraction des caractéristiques (prosodiques et dynamiques) des audio. À la fin, nous avons présenté les méthodes des classifications supervisées (SVM et KNN) et les algorithmes de l'apprentissage profond (LSTM et CNN) que nous allons utiliser dans le chapitre suivant pour classer les audio des différentes bases en classes injonctive et non injonctive.

Chapitre 4

Implémentation et test

Dans ce chapitre, nous allons définir les techniques, les langages de programmation et les outils utilisés pour implémenter les méthodes SVM, KNN et LSTM. Nous allons par la suite présenter les différentes bases utilisées et les résultats de la classification obtenus pour chacune d'entre elles.

4.1 Outils informatiques

Concernant l'implémentation des méthodes SVM et KNN, nous avons utilisé le langage de programmation Python et ses différentes bibliothèques skit-learn, pandas, numpy, etc. Quant aux méthodes d'apprentissage profond, nous avons utilisé le langage Matlab.

- **Python** : est un langage de programmation général, interprété, interactif, orienté objet et de haut niveau. Le code source Python est disponible sous licence GNU ou General Public License (GPL).
- **Scikit-learn** : créée en 2007, scikit-learn est une bibliothèque open-source pour l'apprentissage automatique en Python. Bénéficiant d'une communauté extrêmement active, elle regroupe tous les principaux algorithmes d'apprentissage automatique (classification, régression, etc.). Elle est conçue pour s'harmoniser avec d'autres bibliothèques open-source, notamment NumPy et SciPy.[38]
- **Pandas** : est une bibliothèque permettant la manipulation et l'analyse des données. Elle propose, en particulier, des structures de données et des opérations de manipulation de tableaux numériques et de séries temporelles. [29]
- **Matlab** : est un logiciel interactif basé sur le calcul matriciel. Il est utilisé dans les

calculs scientifiques et les problèmes d'ingénierie parce qu'il permet de résoudre des problèmes numériques complexes en moins de temps requis par les langages de programmation courants, et ce grâce à une multitude de fonctions intégrées et à plusieurs outils testés et regroupés selon usage dans des dossiers appelés boites "toolbox". [20]

4.2 Critères d'évaluation de la classification

La dernière étape nécessaire à la mise en place d'un système de classification est son évaluation. C'est-à-dire savoir si le système a bien réussi son apprentissage ou non. Ceci revient à évaluer ses performances sur les données de test, ces données doivent être différentes des données utilisées pour l'apprentissage. Lors de la phase de test des classifieurs, nous obtenons des étiquettes attribuées à chacun des audio correspondantes aux classes apprises.

Nous allons évoquer, dans cette partie, les méthodes et les métriques scalaires qui permettent d'évaluer et d'analyser les performances des méthodes de classification.

4.2.1 La précision

La précision (en anglais : accuracy) est l'un des critères permettant d'évaluer la performance des modèles de classification. Elle désigne la proportion des prédictions correctes effectuées par le modèle et elle est définie ainsi :

$$précision = \frac{\text{Nombre de prédictions correctes}}{\text{Nombre total de prédictions}}$$

4.2.2 La validation croisée

La validation croisée (K-fold) est une procédure utilisée en apprentissage automatique pour évaluer la stabilité des performances d'un système d'apprentissage. Elle est utilisée pour estimer la capacité d'un modèle d'apprentissage automatique vis-à-vis de différentes répartitions des bases d'apprentissage et de test. Dans cette technique, les données sont divisées en K sous-ensembles différents. La méthode est ensuite répétée K fois, de sorte qu'à chaque fois, l'un des K sous-ensembles est utilisé comme ensemble de test et les $K-1$ autres sous-ensembles forment l'ensemble d'apprentissage. L'estimation de l'erreur

est moyennée sur l'ensemble des K essais pour obtenir l'efficacité totale du modèle [39].

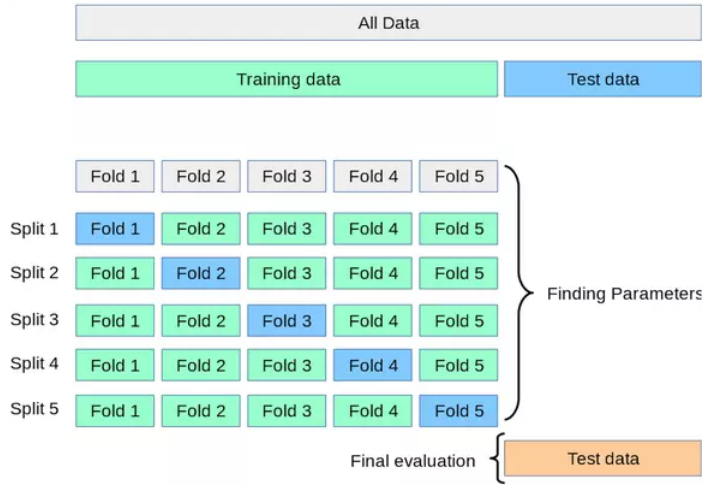

Fig 4.1 – Principe de la validation croisée [37]

4.2.3 La matrice de confusion

La matrice de confusion représente un résumé des résultats de prédictions sur un problème de classification. Les prédictions correctes et incorrectes sont mises en lumière et réparties par classes et elles sont ainsi comparées avec les valeurs réelles. Cette matrice permet de comprendre de quelle façon le modèle de classification est confus lorsqu'il effectue des prédictions.

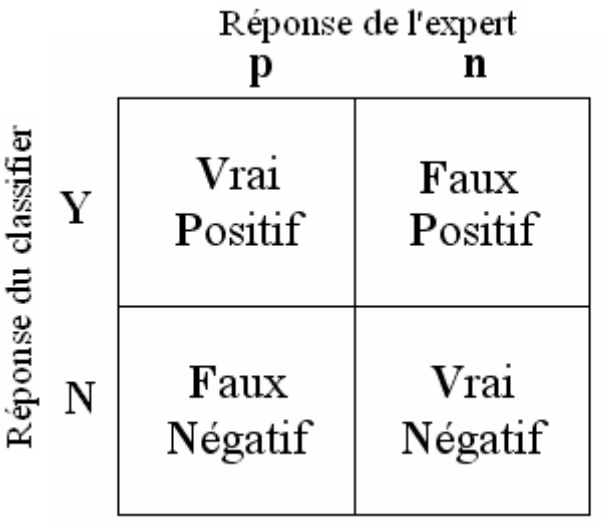

Fig 4.2 – Matrice de confusion[2]

Pour bien comprendre le fonctionnement d'une matrice de confusion, il convient de bien comprendre les quatre terminologies principales :

- Vrai positif : la prédiction est positive, et la valeur réelle est positive.
- Vrai négatif : la prédiction est positive, mais la valeur réelle est négative.
- Faux positif : la prédiction est positive, mais la valeur réelle est négative.
- Faux positif : la prédiction est négative, mais la valeur réelle est positive.

4.3 Implémentation

Dans cette partie nous allons présenter les méthodes et technique utilisés pour l'implémentation des différentes méthodes de classification que nous avons utilisé dans notre étude.

4.3.1 Méthodes SVM et KNN

Comme nous l'avons précisé dans le chapitre précédent, chaque audio est divisé en un ensemble de trames de taille fixé à $10ms$ et les caractéristiques prosodiques sont calculées sur chacune des trames. Les méthodes de classification permettent d'attribuer à chaque trame de l'audio une étiquette correspondante aux classes injonctives (INJ) ou non injonctive (NINJ). Pour définir l'étiquette globale de l'audio nous avons appliqué la méthode de vote majoritaire sur l'ensemble des trames de chaque audio, Ce qui est représenté par la figure suivante :

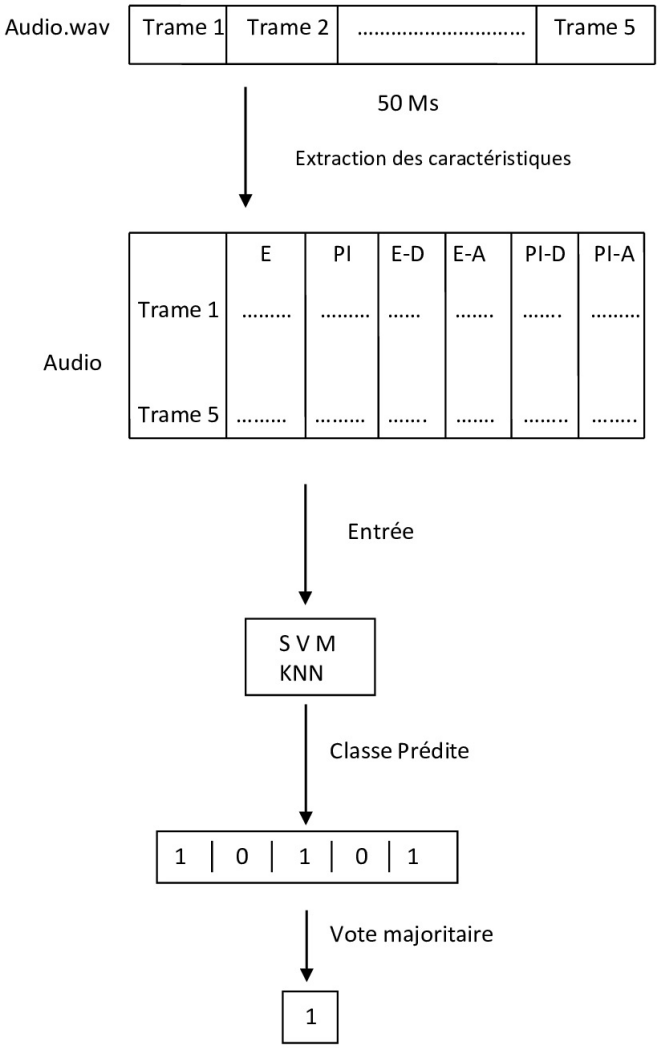

Fig 4.3 – Représentation schématique du pipeline de la classification

4.3.2 Méthode LSTM

Dans cette partie nous allons présenter la méthode LSTM que nous avons utilisé pour classer les audio. Cette méthode prend en entrée des segments d'audio et non pas des trames. Ceci est très utile pour le traitement d'audio car lorsque nous utilisons des couches appropriées d'incorporation et d'encodage dans le LSTM, le modèle sera capable de trouver la signification réelle de l'audio d'entrée et donnera la classe la plus précise.

Pendant la phase d'apprentissage, par défaut, le programme divise les données d'apprentissage en mini-lots (mini-batchs) puis il compresse les séquences de même lot pour qu'elles aient la même longueur. Pour ce faire, le modèle ajoute des zéros pour mettre tous les audio à la taille maximale des audio du lot. Ce remplissage non optimal peut

avoir un impact négatif sur les performances du réseau. Pour pallier à ce problème, nous avons trié les audio d'apprentissage par leur durées. De cette façon, nous avons permis au modèle de sélectionner les séquences d'un même mini-lot ayant des longueurs similaires optimisant ainsi l'apprentissage.

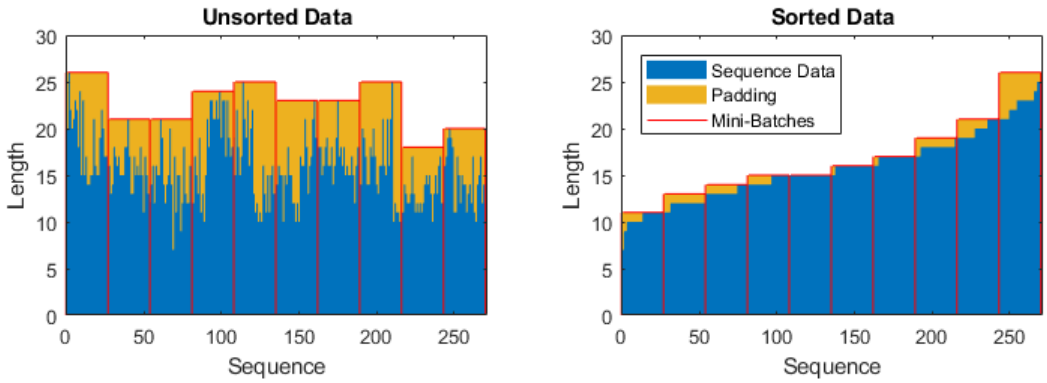

Fig 4.4 – Représentation de l'effet du remplissage des séquences avant et après le tri des données. Cette procédure permet de minimiser l'effet du zéro padding dans la phase d'apprentissage. [32]

4.4 Tests

Dans le cadre de notre projet, nous avons utilisé deux bases de données différentes : la première base est la base utilisée par Abdenour Hacine-Gharbi et al [18] et la deuxième base c'est celle que nous avons générée durant notre étude. Afin de classer les audio de ces bases en classes injonctives et non injonctives nous avons employé les méthodes SVM, KNN et LSTM.

4.4.1 Méthodes SVM et KNN

Dans cette partie nous allons présenter les taux de classification (TC) optimaux trouvés par les méthodes SVM et KNN après plusieurs tests :

- pour la méthode KNN nous avons fait varier le nombre des plus proches voisins (N : tous les nombres impairs entre 1 et 200 pour la première base et entre 1-400 pour la deuxième base)
- pour la méthode SVM nous avons choisi le noyau RBF (Radial Basis Function) et nous avons changé les valeurs du paramètre de régularisation ($C = [0.01, 0.1, 0.5, 1, 1.5, 5, 10]$) et les valeurs de coefficient du noyau ($\gamma = [0.01, 0.1, 1, 10, 100]$)

Dans cette étude nous avons noté les différents types de configuration de descripteur comme suit : $TYPE_{DA}$ où TYPE est PI (pitch ou la hauteur) ou E (énergie logarithmique). Le paramètre D représente la dérivée (vitesse) et A est la double dérivée (accélération).

4.4.1.1 Première base

Le premier objectif de cette étude est de confirmer les résultats de classification précédents trouvés avec la méthode GMM (prédominance de la caractéristique d'énergie logarithmique) [18]. Pour ceci, nous avons appliqué les méthodes KNN et SVM sur la même base de données utilisée par Abdenour Hacine-Gharbi et al.

Cette base a été divisée en un ensemble de données d'apprentissage qui contient 100 audio injonctifs et 99 audio non-injonctifs et un ensemble de données de test composé de 97 audio injonctifs et 99 audio non injonctifs.

Les tableaux suivants représentent les meilleurs taux de classification (précision) obtenus sur les différentes configurations ainsi que les paramètres optimaux utilisés pour chaque méthode.

Configs	E	E,E$_D$	E,E$_A$	E,E$_{DA}$	PI	PI,PI$_{DA}$	PI,E	E,E$_D$, PI_D	PI,E,E$_{DA}$, PI_{DA}
N	51	5	3	5	27	39	3	89	11
TC (%)	**71.42**	65.75	69.89	65.30	65.81	66.83	69.83	70.42	65.81

TABLE 4.1 – Résultats obtenus avec la méthode KNN

Configs	E	E,E$_D$	E,E$_A$	E,E$_{DA}$	PI	PI,PI$_{DA}$	PI,E	E,E$_D$, PI_D	PI,E,E$_{DA}$, PI_{DA}
C,γ	10,1	0.1,10	10,1	1,1	10,1	10,0.01	10,1	1,1	10,1
TC (%)	**82**	76.02	73.46	77	63	65	65	70.91	67

TABLE 4.2 – Résultats obtenus avec la méthode SVM

Discussion : après l'étude des résultats de la classification des audio de la première base par les méthodes SVM et KNN on déduit que :

- la méthode SVM donne un taux de classification optimale de 82% en utilisant l'énergie logarithmique, $C = 10$ et $\gamma = 1$.
- un taux de classification optimale de 71.42% obtenu par la méthode KNN avec l'énérgie logarithmique et $N = 51$.

Nous avons aussi réalisé une étude comparative des taux de la classification obtenus

par les méthodes SVM, KNN et GMM (les résultats obtenus par Hacine-Gharbi et al[18]) appliqué sur différentes caractéristiques. Les résultats sont présentés sur la figure ci-dessous :

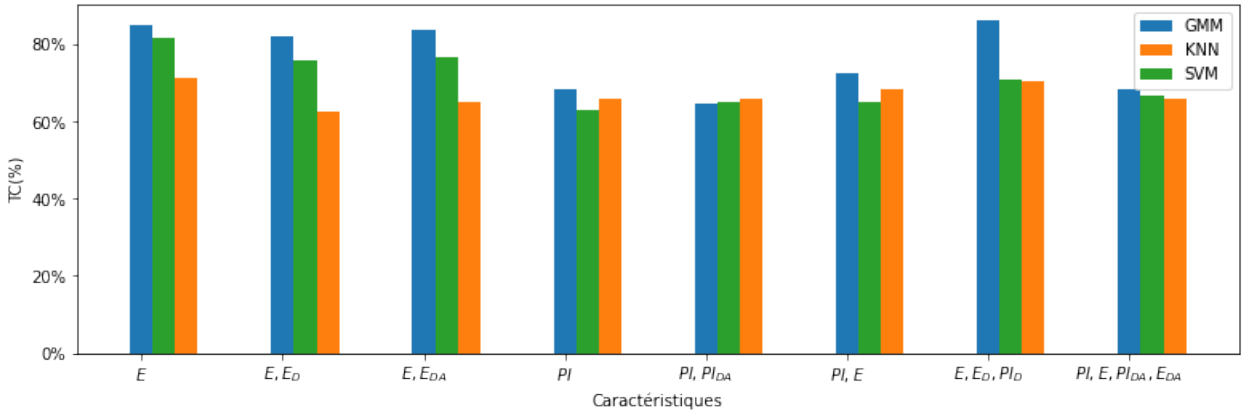

Fig 4.5 – Les résultats obtenus par les méthodes SVM, KNN et GMM. [21]

– [] **Discussion** : en examinant les résultats présentés sur la figure ci-dessus nous avons tiré les conclusions suivantes :
  – les trois méthodes montrent clairement l'importance de la caractéristique prosodique d'énergie logarithmique, qu'elle soit utilisée dans sa version statique ou dynamique.
  – la méthode GMM donne de meilleurs résultats comparée aux autres méthodes (SVM et KNN)

Afin d'estimer au mieux ces résultats, nous avons utilisé la validation croisée (K-fold dans ce cas : K=2).

Les tableaux ci-dessous donnent les valeurs des taux de classification et les valeurs optimales de N (resp C et γ) des méthodes KNN (resp SVM) en fonction des configurations des types de descripteurs.

Configs	E	E,E$_D$	E,E$_{DA}$	PI	PI,PI$_{DA}$	E,E$_D$, PI_D	PI,E,E$_{DA}$, PI_{DA}
N	3	5	7	27	5	123	11
TC (%)	**[50.75, 69.89]**	[50.75, 62.75]	[53.76, 61.73]	[49.74, 65.81]	[49.24, 68.36]	[49.24, 68.36]	[49.50, 65.81]

TABLE 4.3 – Résultats obtenus avec la méthode KNN en utilisant 2-fold([1-fold1,2-fold]).

Configs	E	E,E$_D$	E,E$_{DA}$	PI	PI,PI$_{DA}$	E,E$_D$, PI_D	PI,E,E$_{DA}$, PI_{DA}
C,γ	0.01,0.01	0.01,0.01	0.1,0.01	0.1,0.01	10,0.01	0.01,0.01	0.5,0.01
TC (%)	**[49.74, 71.93]**	[49.74, 64.28]	[49.74, 61.73]	[48.24, 56.63]	[34.18, 50.25]	[33.36, 51.25]	[51.25, 71.93]

TABLE 4.4 – Résultats obtenus avec la méthode SVM en utilisant 2-fold([1-fold,2-fold]).

Discussion : l'application de la validation croisée sur les méthodes SVM et KNN nous

a montré qu'il y avait un grand écart entre les taux de classification des deux groupes (folds) et cela peut être justifié par la différence des durées temporelles des audio utilisés pour chaque groupe, ce que montre le tableau suivant :

	INJ	NINJ
1-fold (minutes)	1.2	0.909
2-fold (minutes)	1.3215	3.463

TABLE 4.5 – La durée en minute des audio dans chaque fold.

À partir du tableau ci-dessus qui représente la durée des audio utilisés par chaque groupe (fold) on déduit que les durées des audio injonctifs et non injonctifs sont similaires pour le 1-fold contrairement aux 2-folds où la durée des audio non injonctives est presque trois fois plus importante que celle des injonctifs. Lorsqu'on utilise les audio de 2-folds en apprentissage, les méthodes apprennent mieux les audio non injonctifs et elles ont tendance à privilégier un classement de tous les audio en non injonctifs, ce que montrent les matrices de confusion présentées ci-dessous :

	Méthode KNN		Méthode SVM	
Configs	groupe1	goupe2	groupe1	goupe2
E,E$_{DA}$, PI, PI_{DA}	$\begin{pmatrix} 0 & 100 \\ 1 & 98 \end{pmatrix}$	$\begin{pmatrix} 44 & 53 \\ 14 & 85 \end{pmatrix}$	$\begin{pmatrix} 1 & 99 \\ 0 & 99 \end{pmatrix}$	$\begin{pmatrix} 54 & 43 \\ 11 & 88 \end{pmatrix}$
E	$\begin{pmatrix} 2 & 98 \\ 0 & 99 \end{pmatrix}$	$\begin{pmatrix} 31 & 66 \\ 7 & 92 \end{pmatrix}$	$\begin{pmatrix} 1 & 99 \\ 0 & 99 \end{pmatrix}$	$\begin{pmatrix} 25 & 72 \\ 1 & 98 \end{pmatrix}$
PI	$\begin{pmatrix} 0 & 100 \\ 0 & 99 \end{pmatrix}$	$\begin{pmatrix} 43 & 54 \\ 13 & 86 \end{pmatrix}$	$\begin{pmatrix} 13 & 87 \\ 15 & 84 \end{pmatrix}$	$\begin{pmatrix} 26 & 71 \\ 15 & 84 \end{pmatrix}$
PI,E$_D$, PI_D	$\begin{pmatrix} 0 & 100 \\ 1 & 98 \end{pmatrix}$	$\begin{pmatrix} 39 & 58 \\ 4 & 95 \end{pmatrix}$	$\begin{pmatrix} 4 & 96 \\ 1 & 98 \end{pmatrix}$	$\begin{pmatrix} 13 & 84 \\ 46 & 53 \end{pmatrix}$

TABLE 4.6 – Les matrices de confusion obtenues par la méthode KNN et SVM dans le cas de 2-folds appliqué sur la première base. Pour la phase d'apprentissage, le groupe2 est équilibré sur les deux classes INJ / NINJ alors que le groupe1 ne l'est pas.

Une base déséquilibrée pour l'apprentissage peut donc être la cause d'une instabilité des performances obtenues lors de la procédure K-fold. Nous chercherons à valider cette hypothèse avec une base équilibrée, détaillée dans la partie suivante.

4.4.1.2 Deuxième base

Afin de valider notre travail et les résultats obtenus avec la première base nous avons testé les méthodes SVM et KNN sur la deuxième base. Cette base contient 910 audio

injonctifs et 910 audio non injonctifs lesquels ne contiennent pas de silence. De plus la durée totale des audio injonctifs est égale à celle des non injonctifs et cela afin d'éviter le problème d'écart des taux de la classification rencontré avec la première base.

– **La base complète**

Dans ce cas nous avons appliqué les méthodes SVM et KNN sur toute la base en utilisant la validation croisée (K = 2 et 5). Dans le cas de 2-fold nous avons utilisé des ensembles de test et d'apprentissage qui contiennent 455 audio injonctifs et 455 audio non injonctifs. Concernant le cas 5-folds nous avons utilisé un ensemble de test qui contient 182 audio injonctifs et 182 audio non injonctifs et un ensemble d'apprentissage qui contient 728 audio injonctifs et 728 audio non injonctifs.

Les tableaux ci-dessous représentent les taux de classification (TC : précision) pour les meilleures configurations trouvées par les méthodes SVM et KNN en utilisant la validation croisée (K=2 et 5).

	2-folds			5-folds		
Configs	N	TC(%)	moyen TC(%)	N	TC(%)	moyen TC(%)
E,PI,E$_{DA}$, PI_{DA}	21	[65.82, 61.20]	**63.51**	7	[63, 64, 65.50, 62, 64]	**64.17**
E	95	[55.81, 56.48]	56.15	21	[51.37, 56.04, 55.21, 59.34, 58.79]	56.15
E, E$_D$	49	[58.79, 60.43]	59.61	77	[55.76, 59.89, 59.06, 56.31, 65.93]	59.39
E, E$_A$	45	[58.79, 62.96]	60.87	91	[58.51, 62.08, 58.79, 62.08, 63.73]	61.04
E,E$_{DA}$	69	[53.07, 55.93]	54.50	77	[59.06, 63.73, 59.89, 59.89, 64.83]	61.48
PI	9	[60.95, 55.71]	58.35	9	[56.59, 60.98, 56.86, 56.04, 56.59]	57.41
PI, PI$_D$	33	[61.64, 56.70]	59.175	43	[54.67, 62.36, 57.96, 57.14, 59.89]	58.40
PI, PI$_A$	55	[60.54, 56.48]	59.17	25	[54.94, 62.08, 61.81, 57.41, 53.84]	58.02
PI, E	63	[54.84, 54.84]	54.84	13	[56.31, 62.91, 60.71, 58.24, 54.12]	58.46
PI, PI$_D$, E_D	85	[62.19, 58.13]	60.16	9	[56.86, 59.89, 57.69, 58.24, 59.34]	58.40

TABLE 4.7 – Résultats obtenus avec la méthode KNN en utilisant 2-folds et 5-folds ([1fold,...,5fold]) appliqués sur la base complète

	2-folds				5-folds			
Configs	C	γ	TC	moyen TC	C	γ	TC	moyen TC
E,PI,E$_{DA}$, PI_{DA}	1	1	[58.58, 63.49]	**61.04**	0.1	10	[62, 63.5 64.5, 63, 51]	**63**
E	1	1	[56.15, 56.59]	56.37	10	10	[51.64, 58.79, 54.67, 57.69, 60.71]	56.70
E, E$_D$	10	10	[55.91, 59.34]	58.62	10	10	[53.57, 59.06, 56.31, 60.71, 62.91]	58.51
E, E$_A$	10	10	[56.92, 58.35]	57.63	10	10	[53.29, 58.24, 56.04, 57.41, 61.81]	57.36
E,E$_{DA}$	1	10	[57.25, 59.23]	58.24	1	10	[55.49, 59.34, 57.40, 58.24, 61.53]	58.40
PI	10	10	[57.58, 52.74]	55.16	0.1	10	[52.47, 57.69, 56.31, 53.29, 54.67]	54.89
PI, PI$_D$	10	1	[59.67, 55.93]	57.80	1	1	[56.04, 60.71, 59.06, 56.86, 58.24]	58.18
PI, PI$_A$	10	1	[60.30, 58.02]	59.17	1	10	[55.49, 59.89, 59.34, 56.31, 54.94]	57.19
PI, E	1	10	[60, 57.80]	58.90	1	1	[53.02, 59.34, 55.49, 55.76, 53.57]	55.43
E, PI$_D$, E_D	0.1	10	[60, 57.80]	58.90	1	1	[58.79, 59.89, 58.51, 59.06, 60.16]	59.28

TABLE 4.8 – Résultats obtenus avec la méthode SVM en utilisant 2-folds et 5-folds ([1fold,...,5fold]) appliqués sur la base complète

Discussion : les résultats obtenus avec les méthodes KNN et SVM montrent que :

– [•] les taux de classification maximaux sont obtenus avec la combinaison des

caractéristiques $\{E, PI, E_D, E_A, PI_D, PI_A\}$ dans les deux cas : 2-folds et 5-folds.

- [•] la méthode KNN donne un TC maximal de 63.51% avec $N = 21$ dans le cas 2-folds et un $TC = 64.17\%$ avec $N = 7$ dans le cas de 5-folds.
- [•] un taux de classification maximal de 61.04% (resp 63%) avec $C = 1$ et $\gamma = 1$ (resp $C = 0.1$ et $\gamma = 10$) est obtenu par la méthode SVM dans cas de 2-folds (resp 5-folds).
- [•] les méthodes sont stables puisqu'il n'y a pas une très grande variation de résultat entre les différents folds.

Pour mieux visualiser la performance des méthodes KNN et SVM, nous avons calculé les matrices de confusion de certaines configurations. Ces matrices sont représentées dans le tableau ci-dessous :

	Méthode KNN		Méthode SVM	
Configs	groupe1	goupe2	groupe1	goupe2
E,E$_{DA}$, PI, PI_{DA}	$\begin{pmatrix} 321 & 134 \\ 177 & 278 \end{pmatrix}$	$\begin{pmatrix} 277 & 178 \\ 175 & 280 \end{pmatrix}$	$\begin{pmatrix} 190 & 265 \\ 112 & 340 \end{pmatrix}$	$\begin{pmatrix} 229 & 226 \\ 106 & 349 \end{pmatrix}$
E	$\begin{pmatrix} 265 & 190 \\ 212 & 243 \end{pmatrix}$	$\begin{pmatrix} 282 & 173 \\ 223 & 232 \end{pmatrix}$	$\begin{pmatrix} 277 & 178 \\ 221 & 234 \end{pmatrix}$	$\begin{pmatrix} 273 & 182 \\ 213 & 242 \end{pmatrix}$
E,E$_{DA}$	$\begin{pmatrix} 70 & 385 \\ 42 & 413 \end{pmatrix}$	$\begin{pmatrix} 89 & 366 \\ 35 & 420 \end{pmatrix}$	$\begin{pmatrix} 303 & 152 \\ 237 & 218 \end{pmatrix}$	$\begin{pmatrix} 312 & 143 \\ 228 & 227 \end{pmatrix}$
PI	$\begin{pmatrix} 291 & 164 \\ 191 & 264 \end{pmatrix}$	$\begin{pmatrix} 271 & 184 \\ 219 & 236 \end{pmatrix}$	$\begin{pmatrix} 332 & 123 \\ 263 & 192 \end{pmatrix}$	$\begin{pmatrix} 293 & 162 \\ 268 & 187 \end{pmatrix}$
PI,PI$_{DA}$	$\begin{pmatrix} 55 & 400 \\ 18 & 432 \end{pmatrix}$	$\begin{pmatrix} 44 & 411 \\ 23 & 432 \end{pmatrix}$	$\begin{pmatrix} 295 & 160 \\ 207 & 248 \end{pmatrix}$	$\begin{pmatrix} 277 & 178 \\ 225 & 230 \end{pmatrix}$
PI,E	$\begin{pmatrix} 60 & 395 \\ 25 & 430 \end{pmatrix}$	$\begin{pmatrix} 70 & 385 \\ 38 & 417 \end{pmatrix}$	$\begin{pmatrix} 336 & 119 \\ 240 & 215 \end{pmatrix}$	$\begin{pmatrix} 290 & 165 \\ 258 & 197 \end{pmatrix}$

TABLE 4.9 – Les matrices de confusion obtenues par la méthode KNN et SVM dans le cas de 2-folds

— **Discussion** : on note une certaine instabilité des résultats fournis par la méthode KNN avec une prédominance de classement en NINJ pour 4 configurations sur les 6 étudiées, ceci quelque soit le groupe. La méthode SVM en revanche est plus robuste aux changements de configuration.

– Les bases école et repas

Comme la base de données RAVIOLI est constituée de plusieurs modules (école,

repas, 24H, itinéraire), nous avons réalisé une étude plus fine par module après avoir divisé cette base en sous bases (école et repas, 24H et itinéraire). L'idée est de chercher à savoir si les résultats dépendent des modules aux conditions d'acquisition et d'enregistrement différents. Nous avons utilisé les bases, école et repas, uniquement parce que les deux autres ne contiennent pas un nombre de données suffisant. La base repas contient 326 audio injonctifs et 326 audio non injonctifs et la base école contient 559 audio injonctifs et 559 audio non injonctifs. Les tableaux ci-dessous présentent les valeurs des taux de classification optimaux des bases écoles et repas appliquées sur les méthodes SVM et KNN et les paramètres optimaux utiliser avec chacune des méthodes.

	Méthode KNN			Méthode SVM			
Configs	N	TC	Moyen TC	C	γ	TC	MoyenTC
E,PI,E$_{DA}$, PI_{DA}	35	[61.35, 63.07]	**62.21**	10	0.01	[61.60, 62]	**61.80**
E	43	[56.60, 55.19]	55.90	0.1	1	[55.89,v57.88]	56.88
E,E$_D$	29	[57.67, 62]	59.84	1	10	[57.14, 59.49]	58.32
E,E$_A$	45	[58.92,56.98]	57.95	10	10	[56.25, 58.42]	57.33
E,E$_{DA}$	151	[58.75, 63.62]	61.18	10	10	[57.14, 59.85]	58.49
PI	3	[55.71, 60.03]	57.87	0.1	0.1	[56.07, 57.52]	56.79
PI, PI$_A$	31	[56.78, 61.64]	59.21	10	0.1	[55.89, 60.57]	58.23
PI, PI$_D$	29	[57.85, 60.21]	59.03	1	0.1	[55.71, 58.06]	57.51
PI, PI$_{DA}$	49	[56.60, 61.11]	58.85	1	0.1	[55.89, 60.75]	58.32
PI, E	15	[56.96, 60.93]	58.94	1	0.1	[56.07, 56.80]	56.44
E, E$_D$, PI_D	201	[56.96, 62.36]	59.66	0.1	10	[59.99, 57.80]	58.90
E, E$_D$, PI_{DA}	201	[57.32, 64.33]	60.82	1	10	[56.42, 58.24]	57.33
E, E$_{DA}$, PI_{DA}	21	[60.71, 62.90]	61.80	1	10	[56.60, 57.88]	57.24

TABLE 4.10 – Résultats obtenus avec la base **école** par les méthodes KNN et SVM en utilisant 2-fold

	Méthode KNN			Méthode SVM			
Configs	N	TC	moyen TC	C	γ	TC	TCmoyen
E,PI,E$_{DA}$, PI_{DA}	15	[61.65, 64.41]	**63.03**	0.1	0.1	[60.20, 64.88]	**62.54**
E	201	[50, 53.37]	51.68	0.1	0.1	[51, 52]	51.50
E,E$_D$	23	[55.21, 53.37]	54.29	1	10	[50.92, 54.60]	52.76
E,E$_A$	91	[52.14, 54.90]	53.52	1	0.1	[50.50, 51.50]	51
E,E$_{DA}$	251	[57.05, 57.97]	57.51	1	10	[50.92, 54.60]	52.76
PI	7	[57.97, 65.44]	57.20	0.1	0.01	[59.5, 60.12]	59.81
PI, PI$_A$	47	[58.89, 60.73]	59.81	10	1	[58.58, 63.49]	61.04
PI, PI$_D$	79	[58.58, 61.04]	59.81	0.1	0.01	[59.81, 62.57]	61.19
PI, PI$_{DA}$	29	[60.42, 63.49]	61.96	0.1	0.01	[59.81, 64.11]	61.96
PI, E	5	[58.89 , 57.36]	58.12	0.1	0.01	[59.50, 60.12]	59.81
E, E$_D$, PI_D	25	[57.05, 57.97]	57.51	1	1	[56.74, 57.36]	57.05
E, E$_D$, PI_{DA}	51	[56.74, 61.34]	59.04	0.1	1	[56.74,62.26]	59.50
E, E$_{DA}$, PI_{DA}	11	[58.89, 61.96]	62.07	0.1	1	[56.75,62.28]	59.51

TABLE 4.11 – Résultats obtenus avec la base **repas** par les méthodes KNN et SVM en utilisant 2-fold

Discussion : les tableaux des résultats de la classification des bases école et repas avec les méthodes SVM et KNN en utilisant la validation croisée illustrent que :

- les deux bases donnent un taux de classification maximal en combinant les caractéristiques $\{E, PI, E_D, E_A, PI_D, PI_A\}$;
- avec la base **école** on obtient un $TC = 62.21\%$ (resp $TC = 61.80\%$) avec $N = 35$ (resp $C = 10$ et $\gamma = 0.01$) appliqué sur les méthodes KNN et SVM respectivement ;
- avec la base **repas** on obtient un $TC = 63.03\%$ (resp $TC = 62.54\%$) avec $N = 15$ (resp $C = 0.1$ et $\gamma = 0.1$) appliqué sur les méthodes KNN et SVM respectivement.

Le partage de la base en deux sous-bases d'exemples appartenant au même domaine n'apporte pas de résultats de performances sensiblement différents. Les résultats sont aussi du même ordre de grandeur qu'avec la base complète.

Afin d'améliorer les résultats, nous avons ensuite cherché à travailler sur une base de meilleure qualité. La qualité d'une base de données comporte de nombreux aspects, notamment la cohérence, l'intégrité, l'exactitude et l'exhaustivité. On considère des données de haute qualité si elles sont adaptées aux utilisations auxquelles elles ont été prévues et si elles représentent correctement la construction du monde réel auquel elles se réfèrent, donc nous pouvons dire que la qualité de la base a un grand impact sur les résultats de la classification. Pour cela dans la partie suivante nous allons construire une base

de meilleure qualité en analysant notre base avec un classifieur non supervisé comme k-means.

4.4.1.3 Classification non supervisée k-means sur la deuxième base

Pour améliorer les résultats précédemment obtenus avec la deuxième base nous avons utilisé la méthode k-means afin de diviser les données en K sous-groupe (clusters) cohérent et construire une base de qualité.

La méthode de k-means permet d'effectuer une classification d'un ensemble de données en k clusters. Un cluster regroupe plusieurs concepts similaires. Chaque cluster est décrit par son centre. Les centres des clusters sont mobiles au cours de l'exécution de l'algorithme. L'algorithme peut se présenter comme suit [10] :

- le nombre de clusters, le paramètre k est fourni au départ.
- un ensemble de k centres est choisi aléatoirement dans l'ensemble des données.
- les k clusters sont formés en regroupant dans chaque centre l'ensemble des données plus proches du centre courant que de tout autre centre.
- le centre de chaque cluster est calculé et devient le nouveau centre.
- l'algorithme boucle alors sur l'étape précédente : les données sont réaffectées en fonction de ces nouveaux centres et la condition d'arrêt est que les centres deviennent immobiles.

De manière générale, l'algorithme de clustering du K-means est efficace, mais présente quelques faiblesses comme le nombre de clusters k qui doit être fixé a priori, ce qui oblige à initialiser les centres des clusters. L'initialisation de ces centres conditionne le résultat final.

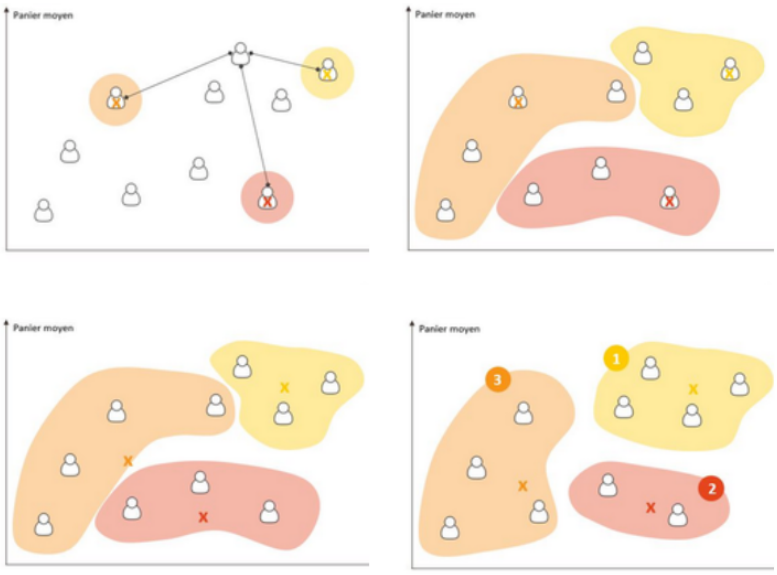

Fig 4.6 – Principe de la méthode k-means [11].

Dans notre étude, nous avons appliqué la méthode k-means sur l'ensemble des valeurs injonctives et non injonctives de chaque configuration.

La figure suivante représente la répartition des valeurs injonctives et non injonctives de E et PI en 2-clusters avec la méthode k-means.

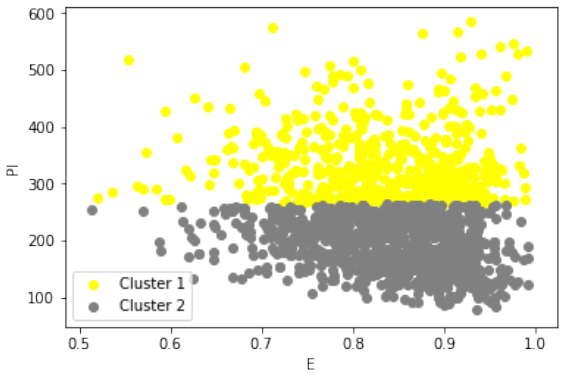

Fig 4.7 – Répartition des valeurs injonctives de E et PI avec 2-means

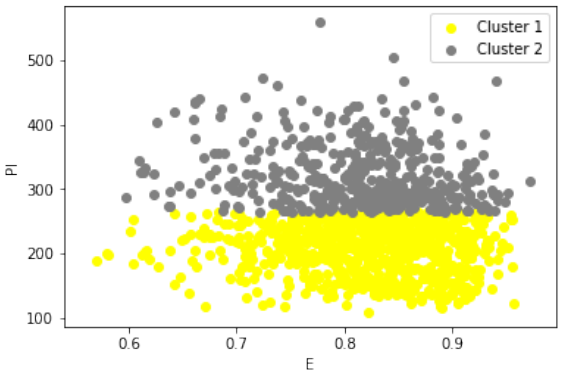

Fig 4.8 – Répartition des valeurs non injonctives de E et PI avec 2-means

Pour la création de la sous-base nous avons choisi pour chaque configuration les audio injonctifs et non injonctifs qui appartiennent au cluster majoritaire (celui qui contient plus de données) et cela pour ne pas perdre beaucoup d'informations. Cette sous-base contient 520 audio injonctives et 520 audio non injonctives.

Les taux de classification de cette sous-base avec les méthodes SVM et KNN en

utilisant la validation croisée ($K = 2$) sont présentés dans le tableau ci-dessous :

	Méthode KNN			Méthode SVM			
Configs	N	TC	moyen TC	C	γ	TC	TCmoyen
E,PI,E_{DA}, PI_{DA}	13	[67.10, 71]	69.05	10	10	[67.57, 71.07]	68.03
E	17	[74, 74.50]	**74.25**	0.1	10	[71.25, 70.07]	**70.66**
E,E_D	15	[73.62, 70.86]	72.24	0.1	10	[70.86, 70.07]	70.47
E,E_A	7	[75.39, 71.85]	73.62	0.1	0.1	[71.45, 68.89]	70.17
E,E_{DA}	11	[74, 71]	72.50	0.1	10	[65, 71.07]	68.03
PI	3	[57, 55]	56	10	10	[65.21, 63.81]	67.75
PI, PI_A	21	[65.57, 66.66]	66.12	10	1	[64.49, 66.12]	65.30
PI, PI_D	93	[67.02, 65.45]	66.24	1	10	[64.49, 65.21]	64.85
PI, PI_{DA}	31	[66, 67]	66.50	10	1	[64.67, 66.30]	65.48
PI, E	23	[65 , 67]	65.50	0.1	1	[64.67, 64.67]	64.67
E, E_D, PI_D	59	[61.89, 60.87]	61.38	1	1	[56.74, 57.36]	57.05
E, E_D, PI_{DA}	87	[62.83, 60.95]	61.89	0.1	1	[56.74,62.26]	59.50
E, E_{DA}, PI_{DA}	77	[63.93, 64.49]	64.21	0.1	1	[56.74,62.26]	59.50

TABLE 4.12 – Résultats obtenus avec les méthodes SVM et KNN testées sur les données obtenues par 2-means

Discussion : on constate à partir des résultats de la classification de la base obtenus par 2-means avec les méthodes SVM et KNN que :

- nous avons obtenu des meilleurs résultats avec cette base par rapport aux autres (deuxième base complète, base repas et base école).
- les deux méthodes SVM et KNN montrent l'importance de l'énergie logarithmique, qu'elle soit utilisée dans sa version statique ou dynamique.
- la méthode KNN fournit un TC maximal de 74.25% avec $N = 17$.
- la méthode SVM donne un taux de classification maximal de 70.66%.
- il n'y a pas une très grande variation de résultat entre les différents folds, on est donc en présence d'un modèle stable.

Les SVM et KNN sont des méthodes non neuronales qui utilisent la classification par trames pour classer un audio. Elles prennent comme entrées des trames d'audio séparées sans tenir compte de l'ordre temporel d'apparition des trames pour les classer dans l'une ou l'autre des catégories. Dans la prochaine partie nous allons utiliser un réseau neuronal qui utilise la classification par segment.

4.4.2 Méthode LSTM

Dans cette étude, nous avons utilisé la méthode LSTM pour classer les audio de la première et la deuxième base. Nous avons choisi cette méthode car elle est efficace pour mémoriser les informations importantes.

4.4.2.1 Première base

Pour la classification de la première base nous avons utilisé la méthode LSTM qui est constituée d'un modèle séquentiel qui est une pile linéaire de couches. Sa première couche est une couche LSTM avec 62 unités de mémoire qui renvoie le dernier pas de temps de la séquence. Il s'ensuit une couche d'exclusion (Dropout) avec une probabilité de 0.5, utilisée pour éviter le sur-ajustement du modèle. Enfin, la dernière couche est une couche entièrement connectée avec une fonction d'activation *softmax* et des neurones égaux au nombre de classe prédite (2).

Pour ajuster cette méthode nous avons utilisé :

- 100 époques (epoch) où une époque est définie comme un passage sur l'ensemble des données ;
- des lots de taille 10 (mini-batch), cela signifie qu'on prend à chaque itération un ensemble d'apprentissage qui contient 10 audio ;
- l'optimisateur *adam* avec $lr = 0.01$ afin de réduire les pertes en modifiant le poids et le taux d'apprentissage de ce modèle ;
- la fonction de perte *cross entropy* pour mesurer l'erreur de la méthode.

Nous avons testé cette méthode sur la première base que nous avons divisé en trois ensembles :

- ensemble d'apprentissage : contient 80% des données de la base.
- ensemble de validation : contient 10% des données de la base.
- ensemble de test : contient 10% des données de la base.

La figure suivante représente la progression de la précision de la classification et la perte d'entropie croisée pour chaque mini-lot pour les ensembles de test et de validation.

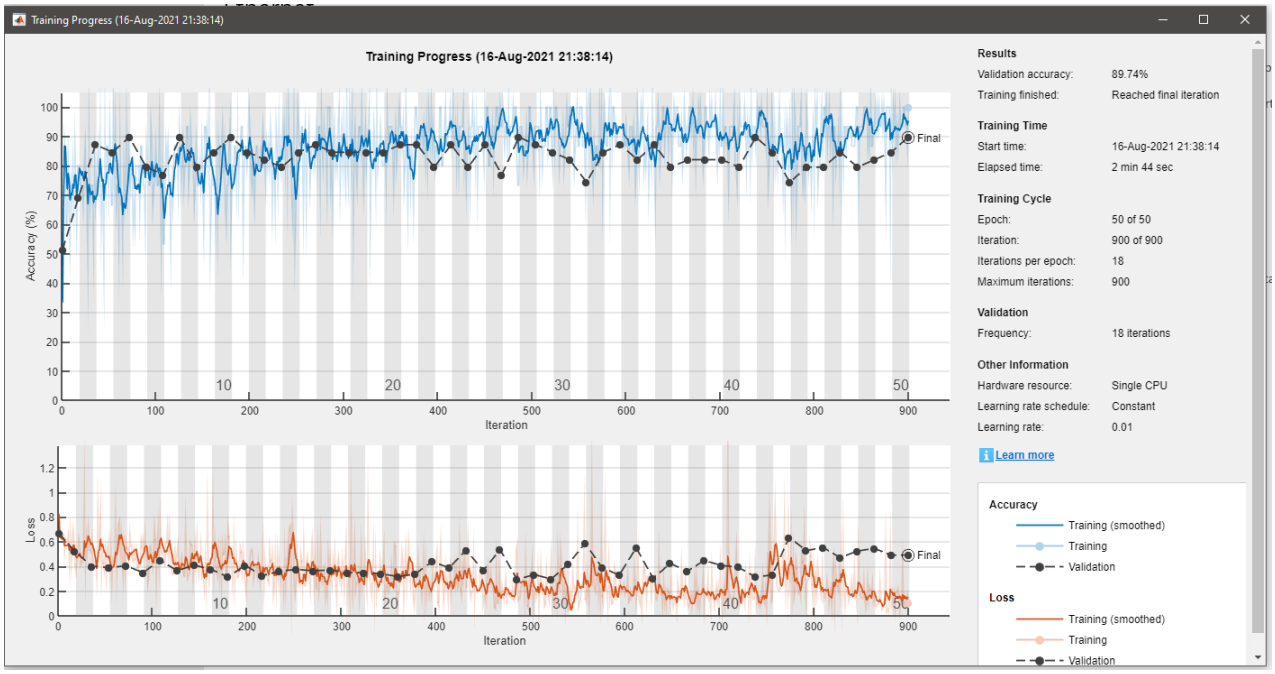

Fig 4.9 – Représentation de la fonction de perte et de la précision des ensembles de validation et de test de la première base

On remarque que les deux graphiques ont le même comportement :

- La courbe de la perte de l'ensemble de validation converge vers 0.15 et la courbe de test converge vers 0.44
- la courbe de progression de la précision montre que les deux ensembles atteignent un taux de classification de 89%
- le taux de classification obtenu avec la base de test est de 75%

Nous avons effectué plusieurs tests afin d'améliorer plus les résultats trouvés mais dans la plupart des cas on tombe sur le problème de sur-apprentissage (overfitting) qui survient lorsque le modèle essaye de trop coller aux données d'entraînements. Parmi les raisons de sur-apprentissage on trouve le manque des données d'entraînement. Cela justifie la poursuite du travail sur la deuxième base constituée de plus de données.

4.4.2.2 Deuxième base

Dans cette partie, nous avons utilisé la méthode LSTM pour la classification des audio de la deuxième base en classes injonctives et non injonctives dont l'architecture du modèle utilisé est la suivante :

- une couche LSTM avec 62 neurones en entrée ;
- une couche de dropout de 20% ;

- une fonction de perte *cross entropy* et l'optimiseur *adam* avec lr=0.001 ;
- un modèle ajusté pour 100 époques d'entraînement avec une taille de mini-lot de 80.

Nous avons appliqué cette méthode sur un ensemble d'apprentissage qui contient 1076 audio injonctifs et non injonctifs et un ensemble de test de validation qui contiennent chacun 364 audio injonctifs et non injonctifs. Les résultats obtenus avec cette méthode sont présentés sur la figure ci-dessous qui représente la progression de la fonction de perte et la précision des ensembles d'apprentissage (bleu et orange sur la figure) et de validation (noir sur la figure) :

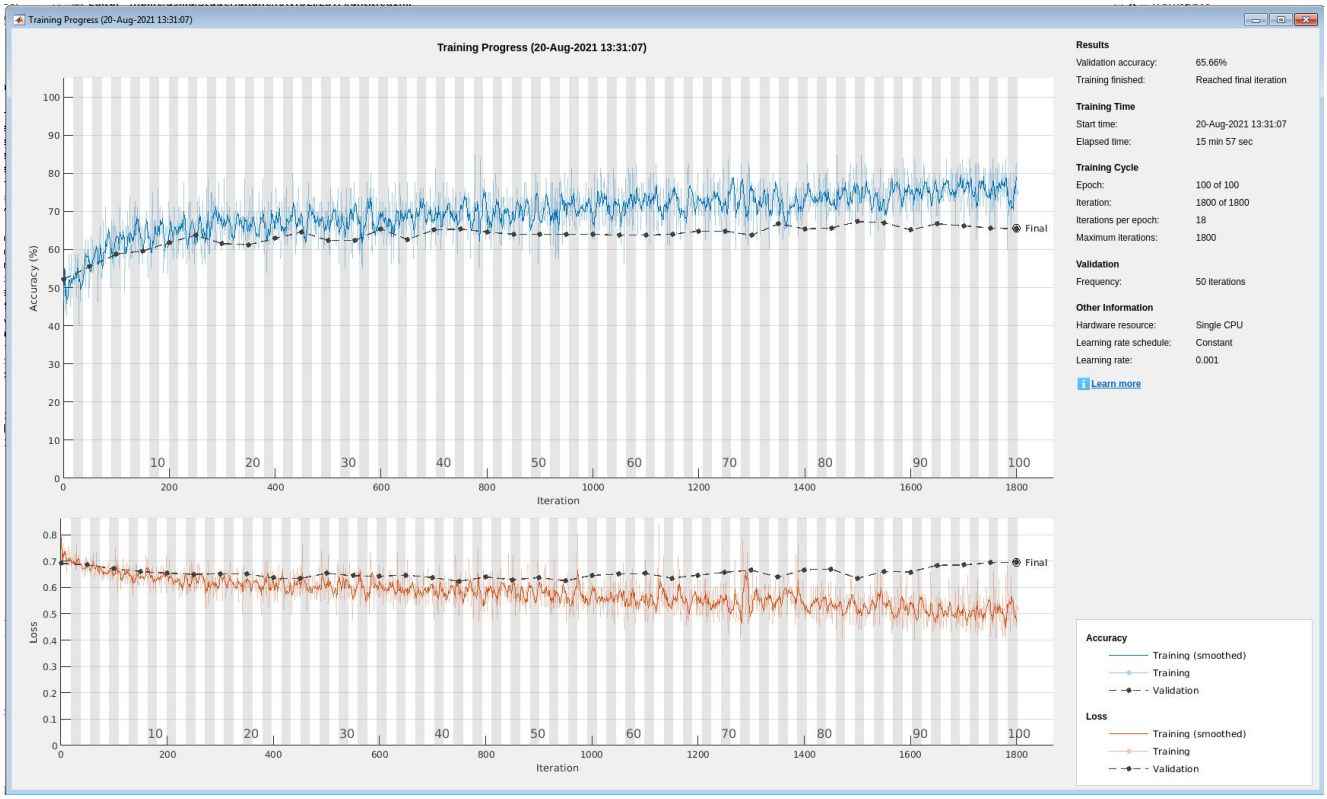

Fig 4.10 – Représentation de la fonction de perte et de la précision des ensembles de validation et de test de la deuxième base.

En explorant ces résultats nous avons déduit que :

- les courbes d'apprentissage et de validation ont le même comportement ;
- la courbe de précision montre que les taux d'apprentissage et de validation atteignent 65% ;
- la courbe de la fonction de perte converge jusqu'à 40%
- le taux final de classification est de 64%.

La méthode LSTM appliquée sur la deuxième base n'a pas amélioré les résultats

trouvés par les méthodes KNN et SVM, pour cela nous allons utiliser dans la prochaine partie les méthodes CNN.

4.4.3 Méthode CNN

Dans la littérature, plusieurs auteurs ont montré l'efficacité des méthodes CNN pour la classification des audio. Pour cela nous avons décidé d'appliquer cette méthode sur la deuxième base en utilisant les mels spectrogramme comme données d'entrée au format image au CNN.

Dans cette étude, nous avons utilisé un CNN dont son architecture est la suivante :

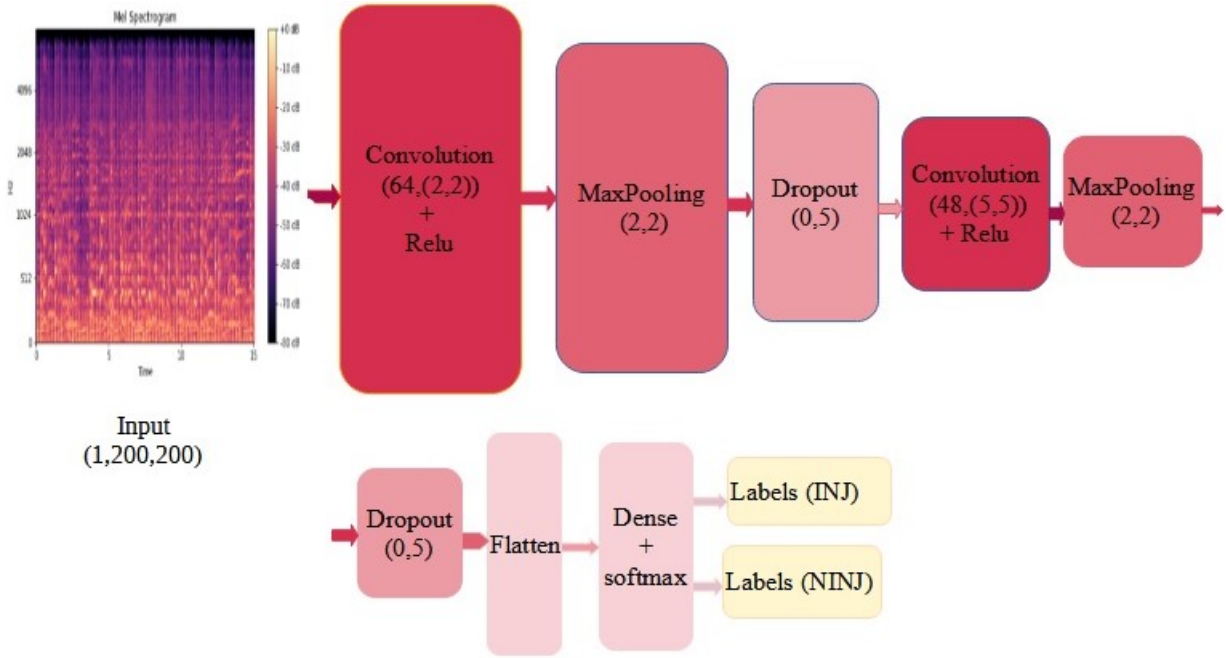

Fig 4.11 – Architecture proposée du CNN

Ce modèle s'appuie sur la plupart des modèles trouvés dans la littérature, avec deux couches de convolution. Le modèle sera ajusté sur 50 époques d'entraînement avec une taille de mini-lot de 50 et en utilisant l'optimiseur *adam*.

Les résultats de la progression de la fonction de perte et de précision des ensembles d'apprentissage et de test obtenus par la méthode CNN sont représentés sur la figure ci-dessous :

Fig 4.12 – Représentation de la fonction de perte et de la précision des ensembles de validation et de test de la deuxième base (modèle CNN).

En examinant ces résultats on déduit que :

- les courbes de test et de validation ont le même comportement ;
- les courbes des pertes pour le test et la validation convergent vers 0.4 ;
- la courbe de précision de l'ensemble d'apprentissage converge vers 80% et celle de test vers 75%.

Le modèle CNN fournit de très bons résultats car il affiche de meilleurs taux de classification avec 10% d'écart avec les meilleurs résultats précédemment obtenus. Il reste cependant à travailler sur la fonction perte qui est encore élevée, ce qui laisse place à une élaboration d'un modèle plus représentatif des données pouvant générer de meilleurs résultats. Ceci passe par la recherche d'autres architectures plus adaptées.

4.5 Conclusion

Dans ce chapitre, nous avons évalué la capacité des méthodes SVM et KNN pour la classification des énoncés provenant d'un corpus de données sauvages de discours oral en classe injonctive et non injonctive. Cela nous a permis de constater qu'avec la première base nous avons obtenu des taux de classification maximaux avec l'énergie logarithmique ce qui confirme les résultats obtenus par Hacine-Gharbi et al[18] avec la méthode GMM.

Nous avons également remarqué que lorsque nous testons ces méthodes sur la deuxième base et ses sous bases (école et repas) les meilleurs taux de classification sont obtenus avec la combinaison des caractéristiques $\{E, ED, EA, PI, PID, PIA\}$. L'application de

k-means sur la deuxième base a permis de réduire la base de données en un ensemble plus homogène. Les meilleurs taux de classification ont été obtenus avec l'énergie logarithmique, indicateur qui avait déjà été identifié comme pertinent au travers des résultats obtenus sur la première base.

Nous avons aussi conclu que le déséquilibre de la base d'apprentissage a une influence sur la stabilité des méthodes de classification.

Par la suite nous avons utilisé la méthode LSTM où nous avons acquis des meilleurs taux de classification avec la première base par rapport aux méthodes SVM et KNN, avec une augmentation d'environ 65% à 75%. Cependant avec la deuxième base elle n'a pas donné de meilleurs résultats. Pour cela, nous avons utilisé la méthode CNN basée sur les mels spectrogrammes où nous avons amélioré les résultats trouvés, avec une augmentation de 64.17% à 75%.

Conclusion générale

Ainsi, nous concluons ce stage de master qui a été présenté dans ce rapport et où un ensemble de tâches a été réalisé.

Nous avons tout d'abord analysé et traité les audio de la base de données RAVIOLI. L'analyse a permis d'identifier la présence d'exemples injonctifs aux durées très courtes (pas forcément exploitables), de cas très bruités avec des audios peu compréhensibles et des enregistrements comportant la présence simultanée de plusieurs locuteurs. Le traitement a permis de supprimer des silences présents dans certains enregistrements.

Par la suite nous avons utilisé les logiciels HTK et Praat pour l'extraction des caractéristiques prosodique (énergie logarithmique et la fréquence fondamentale) ainsi leurs paramètres dynamiques (vitesse et accélération) des audio afin de les utiliser dans la classification.

Nous avons réalisé une étude bibliographique sur les différentes méthodes et techniques déjà utilisés dans la littérature pour la classification des audio. Cela nous a permis de déduire que les méthodes SVM, KNN, LSTM et CNN sont les plus performantes pour cette classification, donc nous avons testé ces méthodes sur deux bases différentes.

Au cours de ce stage, nous avons eu l'occasion de mettre en pratique les cours assimilés pendant mon master CSMI, il a également contribué à l'élargissement de mes compétences et à l'acquisition de nouvelles technologies.

Notre travail est une première version, il est à conforter pour une comparaison totalement objective des performances entre les méthodes. Les LSTM et CNN ont été appliqués sur une configuration de la base répartie en 80 / 10 / 10 alors que les SVM et KNN l'ont été sur une répartition 80 / 20 avec procédure 2-fold ou 5-fold.

Il reste aussi ouvert pour des travaux de comparaison et/ou d'hybridation avec d'autres méthodes de classification, notamment :

- utiliser les méthodes de la classification basée sur la classification par séquence et non par trame des audio comme hybridation CNN+LSTM.
- extraire d'autres caractéristiques (MFCC, PLP....) des audio.
- utiliser une autre base de données plus grande et mieux structurée, extraite à partir du corpus RAVIOLI ; ceci nécessite un travail approfondi avec les linguistes,

encore en cours à l'issue de ce stage.

Bibliographie

[1] YehYun PaoYu ChenJun et ChengYu. "A comparative study of different weighting schemes on KNN-based emotion recognition in Mandarin speech. Advanced Intelligent Computing Theories and Applications". In : (2007), p. 997-1005.

[2] Wannous.H. "Multi view classification of color regions application of the 3D assessment of chronic wounds". In : (2008).

[3] Namunu C.Maddagea Ling He Margaret Lech et Nicholas B. "Study of empirical mode decomposition and spectral analysis for stress and emotion classification in natural speech". In : Biomedical Signal Processing and Control 6 (2011), p. 139-146.

[4] M.S. Kamel M.El Ayadi et F.Karray. "Survey on speech emotion recognition: Features, classification schemes, and databases". In : Pattern Recognition 44 (2011), p. 572-587.

[5] Namrata D. Feature Extraction Methods LPC, PLP and MFCC In Speech Recognition. Gujarat Technology University of INDIA. G H Patel College of Engineering, juil. 2013.

[6] Flocon-Cholet J. Classification audio sous contrainte de faible latence. Rapp. tech. Université Rennes, 2016.

[7] Cosnard M et Bataille F. "Évaluation de l'unité : Laboratoire Pluridisciplinaire de Recherche en Ingénierie des Systèmes, Mécanique, Énergétique PRISME". In : (2017).

[8] Radim Burget Harár Pavol et Malay Kishore Dutta. "Speech Emotion Recognition with Deep Learning". In : (2017).

[9] Et Taejin Lee. “Speech Emotion Recognition Using Convolutional Neural Networks”. In : (2017).

[10] MBAO.M. Distance sémantique et carte conceptuelle. Rapp. tech. Université Montpellier II, 2017.

[11] Marie-Jeanne Vieille.V. “k-means, comment ça marche?” 2017.

[12] Algorithmia. “Introduction to optimizers”. 2018.

[13] HOURRANE.O. “Réseaux neuronaux récurrents et LSTM”. In : (2018).

[14] Stacey.R. “Deep Learning: Which Loss and Activation Functions should I use?” 2018.

[15] Benzaki Y. Introduction à l’algorithme K Nearst Neighbors (K-NN). Rapp. tech. 2018.

[16] Daeyoung Jang Zhao Wootaek et Taejin Lee. “Speech Emotion Recognition Using Deep 1D 2D CNN LSTM Networks”. In : (2019).

[17] zymon et Bozena. “Ranking Speech Features for Their Usage in Singing Emotion Classification”. In : International Symposium on Methodologies for Intelligent Systems.Foundatio (2020), p. 225-234.

[18] A Hacine-Gharbi et Ravier P. “Automatic Classification of French Spontaneous Oral Speech into Injunction and No-Injunction Classes”. In : (2020).

[19] authot. La détection de phonèmes, étape clé de la reconnaissance de la parole. https://www.authot.com/fr/2017/11/22/la-detection-de-phonemes/.

[20] Boughaba.M et Boukhris.B. L’apprentissage profond (Deep Learning) pour la classification et l Rapp. tech. UNIVERSITE KASDI MERBAH OUARGLA.

[21] cfaury. Les K plus proches voisins. https://htk.eng.cam.ac.uk/.

[22] Comment le LSTM améliore le RNN. https://ichi.pro/fr/comment-le-lstm-ameliore-le-rnn-34021890806049.

[23] Comprendre le spectrogramme Mel. https://ichi.pro/fr/comprendre-le-spectrogramme-mel-277775661583955.

[24] Gaëlle F. Traitement de l’oral. Rapp. tech. Université de nante.

[25] HTK. https://htk.eng.cam.ac.uk/.

[26] Abouda L. Traitement automatique de RAVIOLI. https://tln.lifat.univ-tours.fr/version-francaise/projets-en-cours/ravioli.

[27] Chaix P et Ducourneau J Lorenzi A. Son. http://www.cochlea.eu/son.

[28] RAVIER P. Traitement du signal en Sciences du Langage. Rapp. tech. Polytech d'Orléans.

[29] Pandas. https://fr.wikipedia.org/wiki/Pandas.

[30] Représentation du son. http://www.cochlea.eu/son/representation-du-son?fbclid=IwAR1tWIHD5mtL09kG6T5C8mjWV7Yf_9CggXXlCqBtKfoWYcOL6GA3kVqD7HM.

[31] Rod. Comprendre les réseaux de neurones. https://moncoachdata.com/blog/comprendre-les-reseaux-de-neurones/.

[32] Sequence Classification Using Deep Learning. https://fr.mathworks.com/help/deeplearning/ug/classify-sequence-data-using-lstm-networks.html.

[33] Afshine.A et Shervine.A. "Pense-bête d'apprentissage profond".

[34] Afshine.A et Shervine.A. "Pense-bête de réseaux de neurones récurrents".

[35] Signal sonore périodique. https://www.annabac.com/revision-bac/signal-sonore-periodique.

[36] SVM. https://dataanalyticspost.com/Lexique/svm/.

[37] Validation croisée K-Fold pour le Deep Learning à l'aide de Keras. https://ichi.pro/fr/validation-croisee-k-fold-pour-le-deep-learning-a-l-aide-de-keras-69014279685432.

[38] Benoit Y. From scikit-learn to Spark ML. https://blog.engineering.publicissapient.fr/2015/10/08/from-scikit-learn-to-spark-ml/.

[39] Khandelwal Y. Cross Validation and its types! https://www.kaggle.com/discussion/204878.