

## École Nationale des Sciences de l'Informatique

المدرسة الوطنيّة لعلوم الإعلامية

Cycle de formation des ingénieurs en informatique

**Rapport de stage**

Thème :

**plateforme smart pour la détection des agressions verbales**

Réalisé par :

**Montassar nawara**

Encadrée par :

### MMe **Yemna Sayeb**

Période : de 18/06/2025 à 31/07/2025



****Adresse** Campus Universitaire de la Manouba - 2010 Tunisie **T.**: 71 600 444 / **F.**: 71 600 449  
**Email** : [webmaster@ensi.rnu.tn](mailto:webmaster@ensi.rnu.tn)**

**R377+FGR Campus Universitaire de, Manouba 2010 École Nationale des Sciences de l'Informatique, Adresse**

[I. Introduction générale 1](#_Toc27094)

[II. Synthèse et structuration des parties 3](#_Toc14050)

[1. Analyse de technique 3](#_Toc774)

[1.1. Étude des bibliothèques de manipulation sonore 3](#_Toc6065)

[1.1.1. Pydub (Python) 3](#_Toc30105)

[1.1.2. Librosa (Python) 3](#_Toc12954)

[1.1.3. Sounddevice (Python) 3](#_Toc14364)

[1.1.4. OpenSMILE (C++/CLI via Python bindings) 4](#_Toc28658)

[1.1.5. SpeechRecognition (Python) 4](#_Toc13709)

[1.1.6. Web Audio API (JavaScript) 4](#_Toc2775)

[1.1.7. TarsosDSP (Java) 4](#_Toc4185)

[Résumé de cette section 4](#_Toc18591)

[1.2. Conception de étude sonore 5](#_Toc31386)

[1.2.1. Cartes embarquées (microcontrôleurs) 5](#_Toc29874)

[A. ESP32 5](#_Toc13570)

[B. Arduino 5](#_Toc3128)

[C. Raspberry Pi (3/4/Zero) 6](#_Toc4688)

[1.2.2. Les capteur sonore 7](#_Toc8266)

[A. Types de capteurs sonores 7](#_Toc2887)

[B. Conexion 8](#_Toc27263)

[C. Utilisation dans nos projet 8](#_Toc3964)

[1.3. Base de donnée 8](#_Toc12044)

[1.3.1. MySQL 8](#_Toc24950)

[1.3.2. PostgreSQL 9](#_Toc8409)

[1.3.3. MongoDB 9](#_Toc249)

[1.4. Logiciels et environnement de travail 9](#_Toc24369)

[1.4.1. AWS (Amazon Web Services) 10](#_Toc31855)

[1.4.2. GitHub 10](#_Toc8694)

[1.4.3. Server pre-local 10](#_Toc2761)

[1.5. Conclusion 10](#_Toc4231)

[1.6. Outils d’analyse de données 11](#_Toc6488)

[1.6.1. Mise en forme des données sonores 11](#_Toc24756)

[1.6.2. Transfert sécurisé et connecté des données vers le serveur 12](#_Toc5476)

[1.6.3. Modèles d’analyse des données audio 12](#_Toc1653)

[a. CNNs (Convolutional Neural Networks) 12](#_Toc14226)

[b. LSTM (Long Short-Term Memory) 12](#_Toc16961)

[c. ANNs (Artificial Neural Networks) 13](#_Toc1872)

[d. CRNN (Convolutional Recurrent Neural Network) 13](#_Toc2873)

[e. Transformers audio (e.g., Wav2Vec 2.0, Whisper) 13](#_Toc16107)

[f. SVM (Support Vector Machine) 13](#_Toc31616)

[g. Autoencoders / VAE 14](#_Toc15602)

[III. Étude quelques travaux similaires 15](#_Toc22645)

[1. Définition des concepts fondamentaux 15](#_Toc22037)

[1.1. Agressions verbales 15](#_Toc23983)

[1.2. Détection et analyse de données 15](#_Toc12964)

[1.3. Signal 15](#_Toc26858)

[1.4. Détection sonore 15](#_Toc32087)

[1.5. Détection visuelle 16](#_Toc22086)

[2. travaux similaires 16](#_Toc5384)

[2.1. Étude de article « BERSting at the Screams: A Benchmark for Distanced, Emotional and Shouted Speech Recognition » 16](#_Toc6019)

[2.1.1. Présentation et motivation 16](#_Toc16770)

[2.1.2. Méthodologie et démarche 17](#_Toc9762)

[2.1.3. Mise en œuvre et expérimentation 17](#_Toc11163)

[2.1.4. Apports 19](#_Toc12181)

[2.1.5. Synthèse 19](#_Toc6951)

[2.2. Étude de article « An Initial Machine Learning-Based Victim’s Scream Detection» 20](#_Toc5090)

[2.2.1. Présentation et motivation 20](#_Toc20877)

[2.2.2. Méthodologie et démarche 20](#_Toc21093)

[2.2.3. Mise en œuvre et expérimentation 21](#_Toc19975)

[2.2.4. Apports 21](#_Toc7570)

[2.2.5. Synthèse et contribution à notre projet 22](#_Toc28880)

[2.3. Étude de article « Enhancing the Prediction of Episodes of Aggression in Patients with Dementia Using Audio-Based Detection: A Multimodal Late Fusion Approach with a Meta-Classifier » 22](#_Toc21995)

[2.3.1. Présentation et motivation 22](#_Toc23755)

[2.3.2. Méthodologie et démarche 23](#_Toc28039)

[2.3.3. Mise en œuvre et expérimentation 23](#_Toc13167)

[2.3.4. Apports 24](#_Toc213)

[2.3.5. Contribution à notre projet 24](#_Toc25845)

[2.3.6. Synthèse 24](#_Toc5769)

[IV. Conclusion générale des travaux similaires 26](#_Toc24353)

[Enseignements tirés et positionnement de notre projet 26](#_Toc18383)

[V. Analyse et mise en place d une solution 28](#_Toc30487)

[1. Analyse et proposition de solution 28](#_Toc23352)

[1.1. Analyse initiale 28](#_Toc3870)

[1.2. Schéma de solution 28](#_Toc28674)

[1.2.1. Schéma Partie backend 29](#_Toc16194)

[1.2.2. Schéma Partie conception 30](#_Toc1192)

[1.3. Connexion entre capteurs et microprocesseur 31](#_Toc16433)

[1.3.1. Codage en C++ (ESP-IDF ou Arduino IDE) 31](#_Toc15066)

[1.4. Bibliothèque et modèles 31](#_Toc4546)

[1.4.1. Analyse de précision via MATplotlib 31](#_Toc3151)

[1.5. Base de données 32](#_Toc25358)

[1.6. Environnement de travail 32](#_Toc32525)

[Conclusion du chapitre 32](#_Toc12296)

[1.7. Autre outil que nos aident 32](#_Toc7457)

[VI. Bibliographie 32](#_Toc26855)



# Introduction générale

Dans notre société moderne, caractérisée par une vie quotidienne rythmée par les avancées technologiques et les interactions constantes, la question de la sécurité publique demeure cruciale. Bien que nous appartenions à une génération considérée comme l'une des plus avancées sur le plan intellectuel et technologique, des phénomènes de violence continuent à se manifester, que ce soit dans l’espace public ou privé.

La violence, sous ses différentes formes, reste une réalité préoccupante. Elle peut se traduire par des agressions physiques, verbales, des vols, des harcèlements ou toute autre atteinte aux droits fondamentaux des individus. Ces comportements agressifs témoignent d’un non-respect des lois et des valeurs humaines qui fondent la vie en société.

Historiquement, l’homme a toujours cherché à vivre en communauté pour assurer sa sécurité et celle des siens. Le regroupement urbain et l’organisation sociale ont pour objectif d’instaurer une paix durable. Cependant, malgré les efforts déployés par les États, les autorités locales et les forces de l’ordre, les actes d’agression persistent dans nos villes et nos quartiers.

Il ne s'agit pas là d’événements rares ou exceptionnels : chacun d’entre nous a déjà été témoin, victime ou au moins informé d’un acte de violence survenu dans son environnement proche. Cette banalisation de l’agression soulève une interrogation fondamentale : comment peut-on, à défaut d’éradiquer totalement ces actes, protéger les victimes potentielles et assurer que les agresseurs soient rapidement identifiés et traduits en justice ?

Le défi est de taille. De nombreux cas d’agressions se terminent en drames humains, et d'autres ne sont jamais résolus faute de preuves ou en raison de la fuite de l’agresseur. Ce constat met en évidence une nécessité impérieuse : renforcer les outils de détection, de prévention et d’intervention rapide face à ces actes.

Parallèlement, notre époque est marquée par une explosion des données et un développement fulgurant de l’intelligence artificielle. Cette évolution offre aujourd’hui des opportunités inédites pour relever le défi de la sécurité dans l’espace public. Grâce à l’analyse de données massives (big data), à la reconnaissance vocale et visuelle, et à la puissance des algorithmes intelligents, il devient possible de concevoir des systèmes capables de détecter automatiquement des situations à risque.

C’est dans cette optique que notre projet s’inscrit : proposer une solution technologique basée sur l’intelligence artificielle permettant la détection en temps réel des agressions verbales dans les espaces publics. Ce système reposerait sur l’usage de capteurs sonores et visuels implantés dans les zones sensibles des villes.

Dans un premier temps, ces capteurs seraient capables d’identifier une agression verbale à travers l’analyse acoustique ou visuelle. Ensuite, en une fraction de seconde, les données captées seraient transmises à un système centralisé d’analyse qui évaluerait la gravité de la situation.

Selon le niveau de menace détecté, le système pourrait alors déclencher différentes réponses : envoi automatique d’un signal aux patrouilles les plus proches, activation d’effets sonores ou lumineux pour dissuader l’agresseur, ou encore enregistrement de la scène pour usage ultérieur par les autorités compétentes.

L’objectif final est de garantir une intervention rapide, dissuader les comportements agressifs, et renforcer la sécurité des citoyens dans les zones urbaines à forte densité. Ce projet s’inscrit donc à la croisée entre innovation technologique et responsabilité sociale.

En somme, notre démarche vise à mobiliser les outils modernes d’analyse et d’intelligence artificielle pour répondre à une problématique humaine ancienne : la violence urbaine. En combinant capteurs intelligents, traitement en temps réel, et systèmes d’alerte, nous espérons proposer une solution efficace, rapide et éthique. Protéger les individus vulnérables, prévenir les drames et renforcer la justice sont les moteurs de notre initiative. Ce projet n’est pas seulement technologique, il est profondément humain. Il reflète notre engagement envers une société plus sûre et solidaire.

# Synthèse et structuration des parties

## Analyse de technique

Dans cette partie, nous allons explorer les bibliothèques disponibles dans différents langages de programmation, principalement Python, qui permettent d’analyser des signaux audio en temps réel, de traiter et structurer les données sonores, ainsi que de manipuler, enregistrer et stocker ces données de manière efficace. L’objectif est de disposer d’un socle technologique solide pour développer un système intelligent de détection d’agressions verbales dans des environnements urbains.

### Étude des bibliothèques de manipulation sonore

#### Pydub (Python)

Pydub est une bibliothèque Python simple et efficace pour la manipulation de fichiers audio. Elle permet de lire, découper, combiner, convertir et exporter des sons dans divers formats (MP3, WAV, FLAC, etc.).

1. Utilité

traitement de base, conversion de formats, effets simples.

1. Limite

pas adaptée pour du traitement audio en temps réel ou des analyses acoustiques complexes.

#### Librosa (Python)

Librosa est une bibliothèque puissante pour l’analyse audio et la musique. Elle permet d’extraire des caractéristiques audio telles que les MFCC, le tempo, le pitch, les spectrogrammes, etc.

1. Utilité

idéale pour le machine learning audio, la reconnaissance vocale, la classification des émotions ou cris.

1. Limite

nécessite une bonne maîtrise des concepts audio pour une exploitation optimale.

#### Sounddevice (Python)

Sounddevice permet d’enregistrer ou de lire des flux audio en temps réel à partir du microphone ou des haut-parleurs. Elle est compatible avec NumPy pour un traitement direct des données sous forme de tableaux

1. Utilité

enregistrement en temps réel, interface simple avec le matériel.

1. Complémentaire

Librosa ou Pydub pour l’analyse après enregistrement.

#### OpenSMILE (C++/CLI via Python bindings)

OpenSMILE est un outil très puissant pour l'extraction de caractéristiques audio utilisées en reconnaissance d’émotions, analyse vocale, stress ou agressivité.

1. Utilité

extraction de très haut niveau de descripteurs vocaux pour les modèles ML.

1. Utilisé

echerche académique, projets de santé mentale, systèmes de détection émotionnelle.

#### SpeechRecognition (Python)

Cette bibliothèque permet de convertir l’audio en texte en utilisant divers moteurs (Google, Sphinx, etc.).

1. Utilité

econnaissance de mots ou phrases agressives à partir de la voix.

1. Intégration

très utile pour le NLP combiné avec des modèles de classification.

#### Web Audio API (JavaScript)

API native des navigateurs web pour la capture, manipulation et analyse du son directement dans le navigateur.

1. Utilité

utile pour des applications web embarquant de la détection ou des capteurs sonores. Mais elle est pour la navigateur uniquement

#### TarsosDSP (Java)

TarsosDSP est une bibliothèque Java utilisée pour l’analyse du son et la reconnaissance musicale. Elle permet l’extraction de pitch, MFCC, etc.

1. Utilité

alternative Java pour des projets Android ou embarqués.

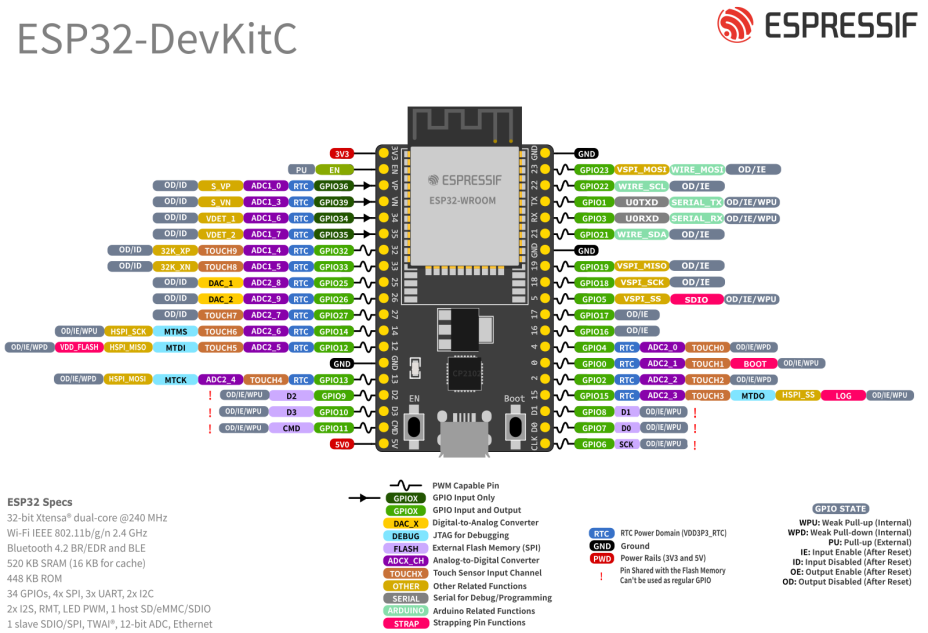
#### ****Résumé de cette section****

La combinaison de bibliothèques comme **Librosa**, **OpenSMILE**,**SpeechRecognition,** et **Sounddevice** offre une base technologique robuste pour capturer, analyser et interpréter les sons suspects dans l’espace urbain. Ces outils sont essentiels pour développer des modèles de détection d’agressions verbales, en temps réel ou différé, et pour assurer une précision dans l'identification du stress, de la colère ou de la violence dans la voix.

### Conception de étude sonore

#### Cartes embarquées (microcontrôleurs)

##### x4ESP32

* **Définition** : Microcontrôleur 32 bits avec Wi-Fi et Bluetooth intégrés.
* **Langage de programmation** : C/C++ (via Arduino IDE ou ESP-IDF), MicroPython.
* **Avantages** :
* Supporte l’**I2S** pour microphones numériques.
* Faible consommation d’énergie.
* Connectivité sans fil native.

**Utilité pour nos projet** : Idéal pour placer des capteurs sonores autonomes, capables de transmettre le signal en temps réel via Wi-Fi à un serveur centra

##### Arduino

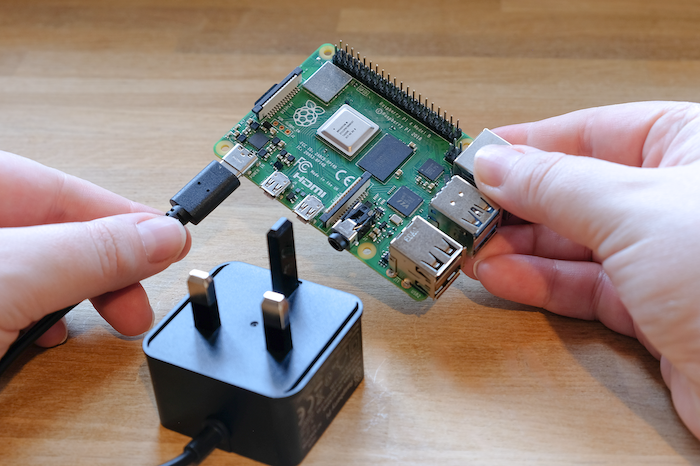
* **Définition** : Microcontrôleur simple, très utilisé pour les prototypes.
* **Langage de programmation** : C/C++ avec l’IDE Arduino.
* **Avantages** :
* Facilité d’utilisation et large communauté.
* Compatible avec de nombreux capteurs analogiques.

**Limites** :

* Pas de connectivité native (sauf si ajout de module Wi-Fi).
* Mémoire limitée, peu adapté au traitement audio complexe.

**Utilité pour nos projet** : utile en phase de prototypage ou pour des traitements simples (seuil sonore, transmission série...).

##### Raspberry Pi (3/4/Zero)

* **Définition** : Nano-ordinateur avec OS complet (Raspberry Pi OS).
* **Langages de programmation** : Python, C++, Node.js, etc.
* **Avantages** :
* Supporte les bibliothèques audio avancées comme pyaudio, librosa, pydub.
* Capable d'exécuter des modèles de machine learning légers.
* Supporte les connexions USB, I2S, ou Jack 3.5mm pour microphone.

**Utilité pour nos projet** : point de traitement central capable de recevoir des données de plusieurs capteurs (ESP32), analyser le son, détecter des agressions, et transmettre des alertes.

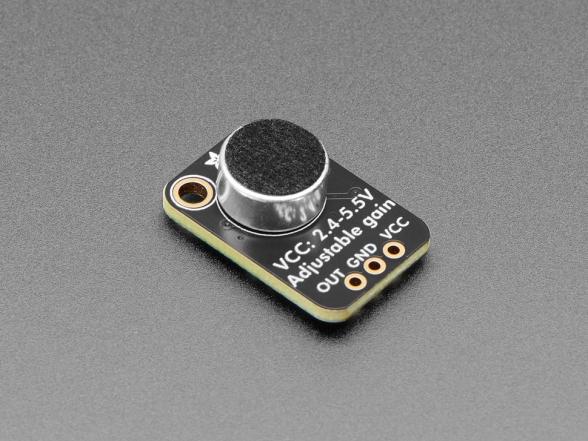
#### Les capteur sonore

La détection fiable des agressions verbales dans un environnement urbain repose avant tout sur l’acquisition d’un signal sonore de bonne qualité. Le choix du capteur est donc une étape critique dans la conception de notre système.

##### Types de capteurs sonores

Il existe plusieurs types de microphones adaptés à différents usages. Voici les plus pertinents pour notre projet :

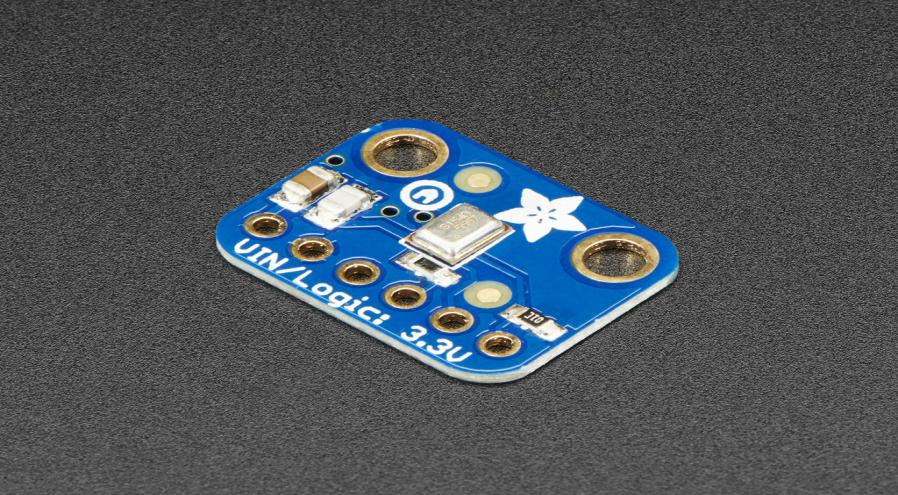
* **Microphones électret** : très utilisés pour les projets embarqués. Peu coûteux, compacts, mais sensibles aux bruits parasites.  
  Exemple :



Adafruit Electret Microphone Amplifier

* **Microphones MEMS (MicroElectroMechanical Systems)** : compacts, résistants et précis, ils offrent une bonne sensibilité et sont idéaux pour une intégration dans des environnements urbains.

Exemple :



SPH0645LM4H MEMS I2S Microphone

* **Microphones omnidirectionnels avec AOP intégré** : ils peuvent capter le son sur 360°, ce qui est utile pour des espaces publics ouverts.

##### Conexion

Les capteurs sonores peuvent être connectés à une carte via différents types d’interfaces :

* **Analogique (A0)** : sortie de tension continue à connecter à une entrée analogique de la carte (ESP32, Arduino).
* **Numérique (I2S, I2C, SPI)** : plus précis et adapté pour des données audio temps réel.

Exemple de câblage :

* Le **micro MEMS I2S** se connecte via I2S (Data In, Clock, Word Select).
* Le **micro analogique** se connecte directement sur une broche analogique + GND + VCC.

##### Utilisation dans nos projet

Dans notre système, les capteurs seront disposés dans des zones sensibles (rues, stations, places publiques). Ils doivent capter les sons en continu, filtrer les bruits ambiants, et transmettre uniquement les signaux suspects vers l’unité centrale de traitement (ESP32, Raspberry Pi, etc.).  
Des algorithmes de prétraitement pourront être embarqués (filtrage, détection de seuil, compression audio...).

### Base de donnée

La base de données constitue un élément central de notre système, permettant de **stocker**, **organiser** et **interroger** les données audio, les métadonnées, les résultats de détection, ainsi que les historiques d’événements. Plusieurs systèmes de gestion de bases de données (SGBD) peuvent être envisagés selon les besoins du projet : relationnel, NoSQL, hébergé ou local.

#### MySQL

* **Type** : Base de données relationnelle open-source.
* **Langage de requête** : SQL.
* **Avantages** :
* Large adoption et bonne documentation.
* Adapté pour la gestion de données structurées (utilisateurs, événements, métadonnées...).

**Cas d’utilisation dans nos projet** :Stockage des profils utilisateurs, journaux d’activité, logs d’alerte, etc.

#### PostgreSQL

* **Type** : Base de données relationnelle avancée, orientée objet.
* **Langage de requête** : SQL (avec extensions comme JSONB, PostGIS...).
* **Avantages** :
* Très robuste pour les traitements complexes.
* Support natif des données JSON, idéal pour stocker des résultats hybrides (audio + méta).

**Cas d’utilisation dans nos projet** : Centralisation des événements audio avec stockage des analyses statistiques ou modèles prédictifs.

#### MongoDB

* **Type** : Base de données NoSQL orientée documents.
* **Langage de requête** : BSON/JSON.
* **Avantages** :
* Flexible pour les données non structurées ou semi-structurées.
* Haute scalabilité, facile à déployer en cloud.

**Cas d’utilisation dans nos projet** :

Stockage de fichiers audio encodés, configurations des capteurs, historiques d’analyses sonores.

### Logiciels et environnement de travail

Pour le développement, le test, et le déploiement du système de détection des agressions verbales, un environnement de travail bien structuré est essentiel. Ce dernier inclut des plateformes de versionnement, d’hébergement, de simulation et de gestion des services cloud.

#### AWS (Amazon Web Services)

**Rôle** : Plateforme cloud utilisée pour héberger les bases de données, les API, les fichiers audio, ou déployer des modèles IA.

**Services pertinents** :

* **EC2** : machines virtuelles.
* **S3** : stockage des fichiers (audio dans nos cas).
* **Lambda** : fonctions serverless pour l’analyse en temps réel.

**Avantage** : Scalabilité, haute disponibilité, sécurité renforcée.

#### GitHub

**Rôle** : Plateforme de gestion de code source.

**Fonctionnalités** :

* Contrôle de version (Git).
* Collaboration en équipe.
* Déploiement automatisé via GitHub Actions.

**Avantage** : Large communauté, intégration CI/CD, visibilité du projet.

#### Server pre-local

**Définition :**

Serveur intermédiaire entre le développement local et la production.

**Utilité :**

* Tester le déploiement final avec configuration proche de la prod.
* Simuler les charges utilisateurs et vérifier la stabilité de l’API ou du système de détection.

**Hébergement possible :**

Machine virtuelle, container Docker, ou cloud privé.

### Conclusion

Le système pourra exploiter **PostgreSQL ou MongoDB** selon la structure des données audio.

L’environnement de développement s’appuiera sur **GitHub/GitLab pour le code**, **serveur local pour le test**, et éventuellement **AWS pour le déploiement final**.

Ces choix offrent une architecture fiable, évolutive, et collaborative pour le bon déroulement du projet.

### Outils d’analyse de données

L’analyse des données audio est une étape fondamentale dans notre projet, car elle permet de transformer des signaux sonores bruts en **données exploitables** pour l'entraînement de modèles d’intelligence artificielle. Cette phase comprend la structuration des données, leur transfert sécurisé, et le choix d'architectures de machine learning adaptées aux caractéristiques temporelles et fréquentielles du signal vocal.

#### Mise en forme des données sonores

Avant l’entraînement, les données audio doivent être **prétraitées et converties** dans un format adapté :

Extraction de caractéristiques acoustiques : **MFCCs** (Mel Frequency Cepstral Coefficients), **Spectrogrammes**, **Chroma features**, etc.

Normalisation des durées ou intensités.

Segmentation de séquences pertinentes (cris, mots-clés...).

Encodage dans des formats compatibles avec les frameworks IA : .npy, .csv, ou TFRecord (pour TensorFlow).

**Bibliothèques utiles** :

* [librosa](https://librosa.org/" \t "_new) : pour l’extraction audio en Python.



* torchaudio : pour le traitement audio avec PyTorch.
* [soundfile](https://pysoundfile.readthedocs.io/" \t "_new) : pour lire/écrire les fichiers audio.

#### Transfert sécurisé et connecté des données vers le serveur

Une fois les données extraites et structurées, elles doivent être **transmises à un serveur de traitement** :

* **Connexion API REST** : via Flask, FastAPI, ou Node.js pour recevoir les fichiers audio et stocker leurs méta-informations.
* **Connexion locale directe** : si l’analyse est effectuée à bord (edge computing) avant synchronisation.
* **Sécurité** : authentification par tokens (JWT), transferts chiffrés HTTPS.
* **Optimisation** : compression des fichiers audio, découpage par lots, base de données adaptée (MongoDB ou PostgreSQL avec liens aux fichiers sur S3).

Modèles d’Pour entraîner notre système à reconnaître des agressions verbales, nous utilisons des **modèles d’apprentissage automatique spécialisés dans le traitement du son**, capables de capter les variations de fréquence, d’intensité et de ton dans les cris ou insultes.

#### Modèles d’analyse des données audio

##### a. CNNs (Convolutional Neural Networks)

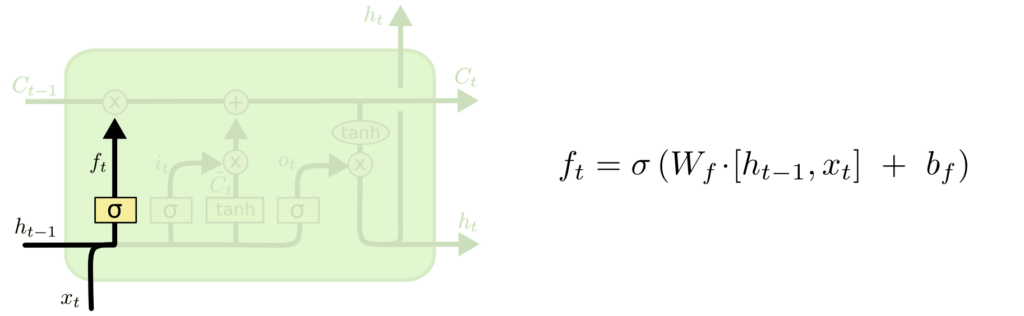
**Utilité :** très efficace pour extraire des caractéristiques spatiales sur les spectrogrammes.

**Exemple :** classification de cris à partir d’un spectrogramme.

##### b. ****LSTM (Long Short-Term Memory)****

**Utilité :** capte les dépendances temporelles dans un signal, idéal pour les séquences vocales longues.

**Application :** reconnaissance de phrases agressives, ou ton de la voix.



##### c. ****ANNs (Artificial Neural Networks)****

**Utilité :** modèles simples mais efficaces pour des jeux de données bien formatés (vecteurs MFCC...).

**Limite :** moins performants que CNN ou LSTM sur les séquences complexes.

<https://scikit-learn.org/stable/modules/neural_networks_supervised.html>

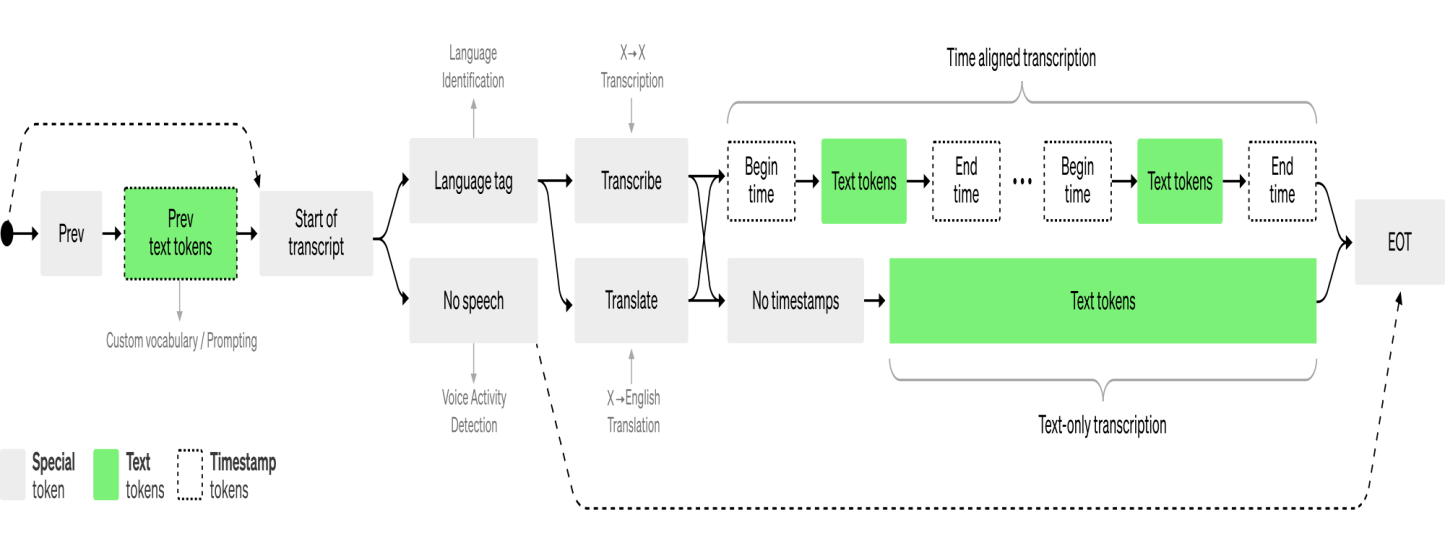
##### d. ****CRNN (Convolutional Recurrent Neural Network)****

**Utilité** : combine les avantages des CNN (extraction spatiale) et LSTM (séquence temporelle).

Performant pour les tâches de détection de cris ou anomalies vocales.

<https://arxiv.org/abs/1609.04243>

##### e. ****Transformers audio (e.g., Wav2Vec 2.0, Whisper)****



**Utilité** : architectures basées sur les Transformers pour transcrire ou classifier la parole directement.

**Avantages** : performances SOTA (state-of-the-art), peu de prétraitement nécessaire.

<https://openai.com/index/whisper/>

<https://github.com/openai/whisper/blob/main/notebooks/LibriSpeech.ipynb>

##### f. ****SVM (Support Vector Machine)****

**Utilité** : modèle supervisé classique, très bon pour les petits jeux de données et la classification linéaire.

**Application** : utilisé dans certains projets de détection de cris en situation d'urgence.

<https://scikit-learn.org/stable/modules/svm.html>

##### g. ****Autoencoders / VAE****

**Utilité** : apprentissage non supervisé pour la détection d’anomalies sonores ou la compression de données audio.

**Application** : détection automatique de sons suspects ou anormaux par reconstruction.

<https://arxiv.org/abs/1711.00520>

# Étude quelques travaux similaires

Dans un premier temps, il est essentiel d’examiner des travaux de recherche existants dans le domaine, à l’échelle mondiale, afin d’identifier les approches déjà proposées pour répondre à des problématiques similaires à la nôtre. Cette revue de littérature nous permettra :

* D’analyser les idées développées dans des projets antérieurs
* De repérer les points forts et les éléments compatibles avec notre propre solution
* D’utiliser certains projets comme base de référence pour construire et améliorer notre approche.

L’objectif est donc de nous appuyer sur ces travaux pour consolider notre démarche, éviter de repartir de zéro, et garantir que notre projet soit à la fois innovant et ancré dans une réalité technologique et scientifique déjà explorée.

## ****Définition des concepts fondamentaux****

Avant d’aborder les projets existants, il convient de bien définir les notions clés qui structurent notre sujet. Cela facilitera la compréhension des travaux étudiés et permettra de mieux situer notre projet dans le paysage technologique actuel.

### ****Agressions verbales****

Les agressions verbales désignent tout acte de communication violent ou menaçant, qui vise à nuire psychologiquement à une personne. Elles incluent les insultes, cris, menaces, propos dégradants ou discriminatoires, et peuvent provoquer un sentiment d’insécurité important, même en l’absence de contact physique.

### ****Détection et analyse de données****

Il s’agit de l’ensemble des techniques permettant de collecter, traiter et interpréter des données issues de différentes sources (audio, vidéo, etc.) dans le but d’identifier des situations suspectes ou à risque. Ces analyses peuvent être basées sur des algorithmes de machine learning, de traitement du signal ou de reconnaissance d’événements.

### ****Signal****

Le signal, dans le contexte de notre projet, désigne l’alerte transmise automatiquement lorsqu’une agression est détectée. Ce signal peut être dirigé vers les autorités (ex. : patrouille de police) ou vers des dispositifs dissuasifs (sirènes, lumières, messages audio...).

### ****Détection sonore****

La détection sonore repose sur l’analyse de l’environnement acoustique afin d’identifier des anomalies vocales comme des cris, des insultes ou des bruits violents. Des microphones intelligents associés à des algorithmes d’analyse permettent de reconnaître ces événements en temps réel.

### ****Détection visuelle****

La détection visuelle utilise des caméras et des techniques de vision par ordinateur pour analyser les comportements humains, les expressions faciales ou les mouvements suspects. Couplée à la détection sonore, elle renforce la fiabilité du système en apportant une confirmation visuelle des incidents détectés.

## **travaux similaires**

### ****Étude de article «**** BERSting at the Screams: A Benchmark for Distanced, Emotional and Shouted Speech Recognition »

#### **Présentation et motivation**

Cet article présente le corpus BERSt, développé conjointement par des chercheurs de la Simon Fraser University (Canada), SUPMICROTECH/CNRS (France) et la société Enchanted Tools. Ce corpus inédit constitue un benchmark vocal conçu pour évaluer la reconnaissance automatique de la parole (ASR) dans des conditions extrêmes, telles que les cris, les émotions intenses et la distance entre le locuteur et le capteur. L’objectif principal est de combler un manque important dans les bases de données actuelles, qui ne couvrent pas suffisamment ces situations réalistes et difficiles.



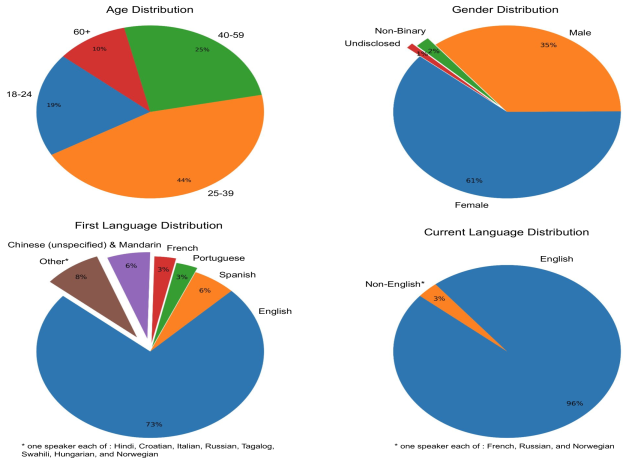
#### x2 **Méthodologie et démarche**

Le travail se concentre sur la création d’une ressource dédiée à l’analyse de la parole émotionnelle, criée et distante, enregistrée dans des environnements naturels au moyen de smartphones. Les principaux objectifs sont les suivants :

* Évaluer les performances des systèmes de reconnaissance automatique de la parole (ASR) dans ces conditions complexes,
* Détecter les cris au sein des enregistrements,
* Reconnaître les émotions véhiculées par la voix (Speech Emotion Recognition - SER).

#### **Mise en œuvre et expérimentation**

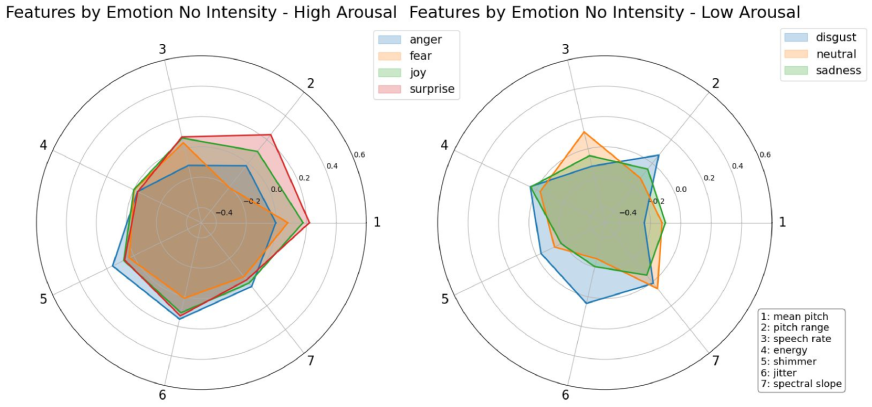
Le corpus comprend 4526 extraits audio, totalisant environ 4 heures d’enregistrements, collectés auprès de 98 locuteurs exprimant sept émotions différentes, notamment la joie, la peur ou la colère. Les enregistrements ont été réalisés à l’aide de smartphones placés dans 19 positions différentes, au sein de 98 configurations variées d’environnement et de distance, afin de garantir un réalisme et une diversité maximale des conditions d’acquisition.



* **Modèles et outils testés** :

• ASR : Whisper Turbo, SpeechBrain, etc.  
• SER : DAWN, Wav2Small  
• Analyse fine des performances selon la distance, le type de cri ou d’émotion.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Architecture** | **# Param.** | **Dataset** | **# h** |
| Whispermedium.en | sequence-to- sequence | 769 M | OpenAI custom\*\* | 680,000 |
| Whisper-turbo | sequence-to-sequence\* | 798 M | OpenAI custom\*\* | 680,000 |
| Ne Mo Quartz net | CNN† with residual connections | 18.9 M | Libri Speech, Common Voice | 3,500 |
| Ne Mo Fast Conformer Trans ducer | Auto regressive conformer | 114 M | NeMo ASRSet | 24,500 |
| Wav2Vec2-Base960h | CNN-based encoder transformer | 94.4 M | LibriSpeech | 960 |

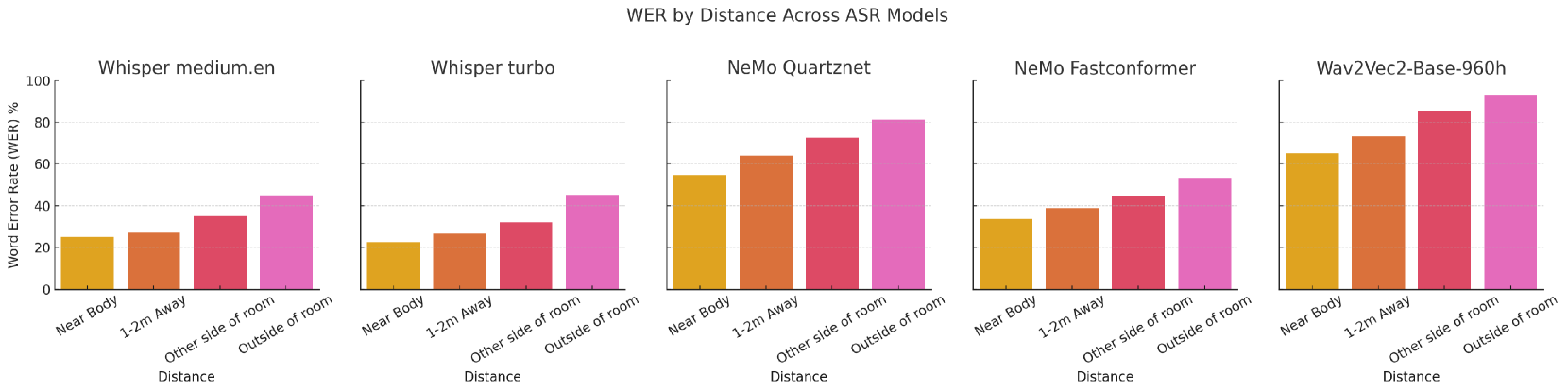


#### **Apports**

Les résultats démontrent que les performances des modèles restent limitées dans ces conditions difficiles :

* Le taux d’erreur de mots (Word Error Rate - WER) est d’environ 30 % pour Whisper Turbo,
* La précision non pondérée en reconnaissance émotionnelle (SER) atteint seulement 32 % avec DAWN.

Ces chiffres révèlent un écart important entre les performances des modèles sur des données standards et leur efficacité en situations réelles et bruyantes, soulignant l’importance d’un corpus tel que BERSt. Celui-ci offre un cadre unique pour mesurer la robustesse des systèmes de reconnaissance vocale face à des données complexes mêlant distance, cris et émotions.



#### **Synthèse**

Le corpus BERSt constitue une contribution majeure à notre problématique, car il cible précisément les contraintes auxquelles nous sommes confrontés : reconnaissance vocale en situation tendue (cris, agressions verbales) en milieu urbain, avec des dispositifs simples tels que des smartphones ou des capteurs acoustiques.

Il fournit une base solide tant sur le plan technique que méthodologique pour construire ou entraîner un système de détection d’agressions vocales dans l’espace public. Ce corpus peut ainsi servir :

* De référence pour comparer nos propres algorithmes,
* De fondement pour tester et améliorer la robustesse des modèles,
* De point de départ pour étendre les travaux existants en intégrant des réactions automatiques (alertes, interventions), une fonctionnalité absente dans l’étude originale.

### ****Étude de article «**** An Initial Machine Learning-Based Victim’s Scream Detection»

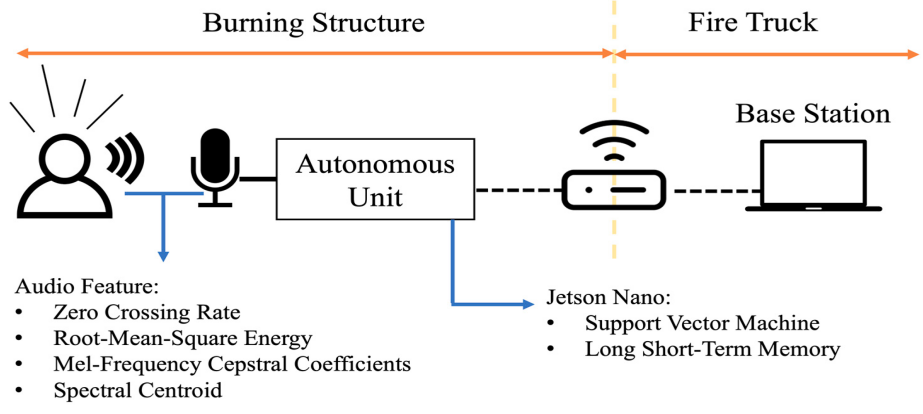
#### x1Présentation et motivation

Cet article présente un système automatisé de détection des cris de victimes, spécifiquement développé pour des situations d’incendies. Les auteurs justifient l’importance de cette recherche par la nécessité d’améliorer la rapidité d’intervention des secours en détectant rapidement les personnes piégées grâce à leurs cris. Ce système est conçu pour être intégré à un véhicule autonome de secours, afin d’opérer dans des environnements extrêmes, notamment à faible visibilité, où la détection visuelle est impossible ou difficile.

#### **Méthodologie et démarche**

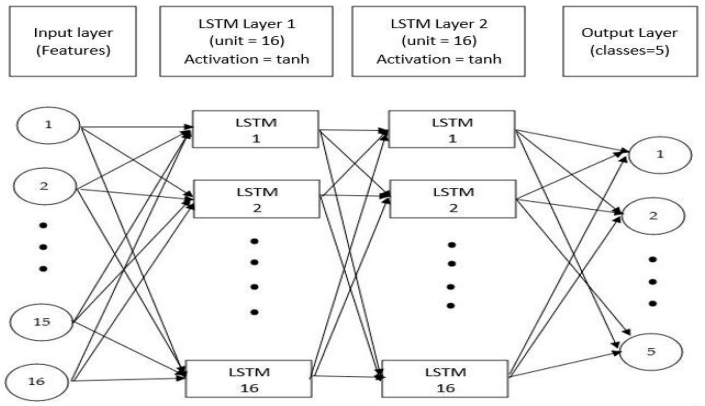
La méthodologie repose sur l’utilisation de techniques de machine learning appliquées à des signaux audio captés sur le terrain. Deux modèles principaux sont comparés :

* Le SVM (Support Vector Machine), reconnu pour sa simplicité et sa rapidité d’exécution,
* Le LSTM (Long Short-Term Memory), un réseau de neurones récurrent capable de modéliser des séquences temporelles complexes.

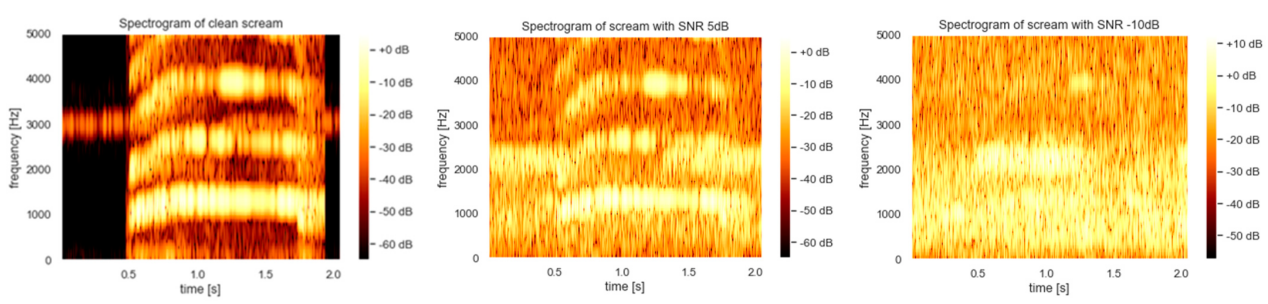
Le traitement audio comprend une phase d’extraction de caractéristiques spécifiques aux cris, qui permet d’alimenter les modèles pour optimiser leur précision. Le système est conçu pour être embarqué dans un dispositif autonome (AESV) destiné aux opérations de secours.

#### **Mise en œuvre et expérimentation**

Les expérimentations ont consisté à entraîner et tester les modèles SVM et LSTM sur des bases de données de cris de victimes, dans des conditions simulant les environnements d’incendie. Le choix des technologies et leur implantation sur un système embarqué ont été évalués afin de garantir une exécution en temps réel, indispensable pour une intervention efficace.

Le SVM s’est distingué par ses performances en termes de rapidité et d’efficacité, facilitant son déploiement sur des plateformes à ressources limitées.

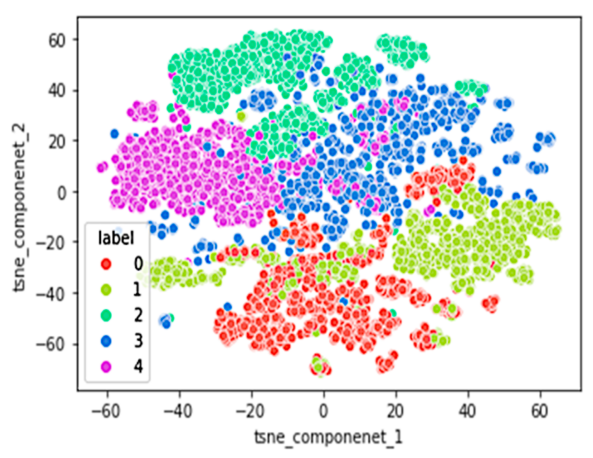
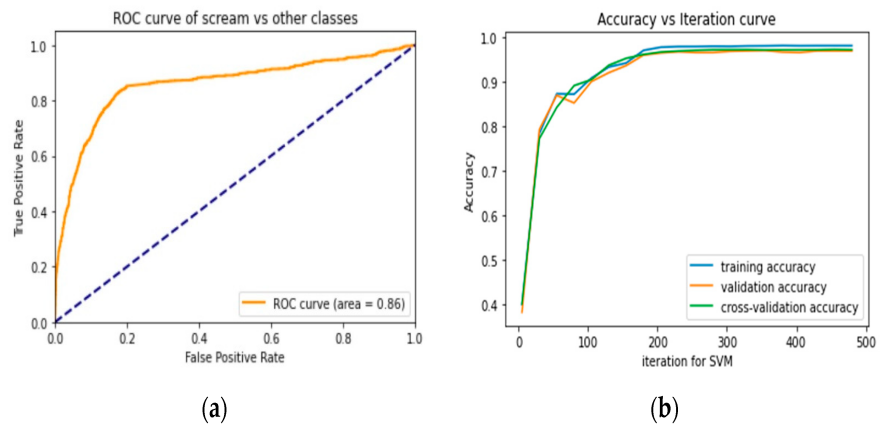




#### Apports

L’étude apporte plusieurs contributions majeures :

* Une très bonne précision dans la détection des cris de détresse, attestant de la pertinence de l’approche audio,
* La démonstration que le modèle SVM, plus léger et rapide, est plus adapté que le LSTM pour une exécution temps réel sur systèmes embarqués,
* La confirmation que le système peut fonctionner sans recours à la vision, un atout essentiel dans les cas de fumée dense ou faible visibilité,
* Une amélioration potentielle significative de la rapidité des secours en situations critiques grâce à cette détection automatisée.



#### Synthèse et contribution à notre projet

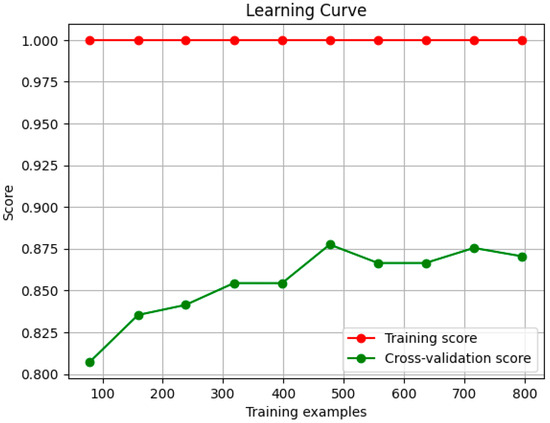
Cet article constitue une base technique solide pour notre problématique, notamment sur deux points essentiels :

* Il valide l’utilisation conjointe de données audio et d’algorithmes de machine learning pour la détection de signaux d’alerte humains,
* Il démontre la faisabilité d’un traitement en temps réel, soutenant ainsi l’idée d’un déclenchement automatique d’alertes vers les patrouilles ou services de secours.

Ainsi, cette étude appuie la pertinence de notre approche et fournit un cadre méthodologique et technologique à exploiter, tout en mettant en lumière la nécessité d’optimiser le compromis entre précision et rapidité dans les environnements contraints.

### ****Étude de article «** Enhancing the Prediction of Episodes of Aggression in Patients with Dementia Using Audio-Based Detection: A Multimodal Late Fusion Approach with a Meta-Classifier »**

#### **Présentation et motivation**

Cet article propose une méthode novatrice pour prédire les épisodes d’agressivité chez les patients atteints de démence, en s’appuyant sur la détection de signaux audio. L’enjeu est de permettre une anticipation proactive des comportements violents, afin d’intervenir à temps dans des établissements de soins. L’approche repose sur l’exploitation de signaux non intrusifs (principalement audio), combinés à d'autres informations contextuelles, afin de renforcer la précision des prédictions. Cette démarche s’inscrit dans un objectif de sécurité et de prévention, aussi bien pour les patients que pour le personnel soignant..

#### **Méthodologie et démarche**

La méthode adoptée par les auteurs repose sur une architecture de fusion multimodale, dite late fusion, dans laquelle différentes sources de données (signaux audio, variables contextuelles) sont traitées séparément avant d’être combinées à l’aide d’un **méta-classifieur**. Cette fusion tardive permet de tirer parti des spécificités de chaque modalité tout en conservant une interprétabilité et une flexibilité accrues.  
Les étapes principales incluent :

* La détection et l’extraction de caractéristiques audio pertinentes,
* L’analyse d’indicateurs contextuels (par exemple, heure, environnement, antécédents),
* L’apprentissage supervisé à l’aide de plusieurs modèles,
* La combinaison finale via un méta-classifieur pour générer une prédiction consolidée.

#### **Mise en œuvre et expérimentation**

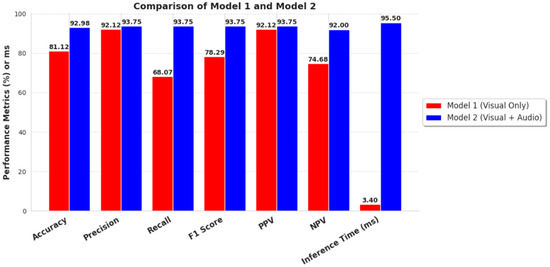
Le système a été testé dans un environnement médical réel, avec une collecte de données audio en continu et une annotation des épisodes d’agression observés. L’étude a mis en œuvre :

* Des techniques avancées de traitement du signal pour isoler les indices audio indicatifs de tension ou de pré-agression,
* Des algorithmes d’apprentissage automatique pour entraîner des modèles sur des séquences temporelles,
* Une fusion tardive (late fusion) des résultats individuels à l’aide d’un méta-classifieur, permettant une prise de décision plus robuste.

L’objectif étant de construire un système capable de s’adapter à la diversité des contextes tout en respectant les contraintes éthiques et techniques liées au domaine médical.

#### Apports

L’article apporte plusieurs contributions significatives :

* Une **amélioration mesurable de la précision de la prédiction** des épisodes agressifs par rapport aux approches basées sur une seule modalité,
* L’**utilisation de données non intrusives** (audio) respectant la vie privée des patients,
* Une **architecture flexible et modulaire**, capable de s’adapter à différents types de contextes médicaux,
* Un **potentiel d’intégration concret** dans des systèmes de surveillance automatisés en établissements de santé, favorisant la réactivité du personnel soignant.

#### Contribution à notre projet

Ce travail est particulièrement pertinent pour notre projet dans la mesure où il partage un objectif similaire : la **détection précoce d’un comportement agressif via l’audio**, dans le but de déclencher une réaction rapide. Bien que le cadre d’application soit différent (milieu médical vs. espace public urbain), les **contraintes et enjeux sont comparables** : garantir la sécurité de populations vulnérables à l’aide de dispositifs discrets et automatisés.

Les similitudes techniques (traitement audio, machine learning, système embarqué, logique temps réel) valident notre propre orientation, et l’approche multimodale constitue une **piste intéressante d’évolution** que nous pourrions envisager pour enrichir notre système, en y intégrant par exemple des données contextuelles (heure, lieu, densité sonore ambiante).

#### Synthèse

Cette étude démontre la **pertinence des signaux audio** comme indicateurs prédictifs de comportements agressifs, en particulier lorsqu’ils sont intégrés dans une **architecture multimodale intelligente**. Elle met en lumière l’importance d’une **analyse fine et contextualisée** des sons pour anticiper les incidents critiques. Le recours à une fusion tardive et à un méta-classifieur offre des perspectives concrètes d’amélioration pour tout système de détection basé sur l’audio, et fournit une **base de réflexion précieuse pour l’enrichissement de notre propre approche**.

## Conclusion générale des travaux similaires

L’étude comparative des trois articles sélectionnés permet de dégager une vision claire de l’évolution des systèmes intelligents de détection acoustique, appliqués à des contextes critiques tels que la sécurité urbaine, les catastrophes naturelles ou encore les soins médicaux.

Le **premier article** a introduit un **benchmark de référence (BERSt)** pour la reconnaissance vocale dans des conditions extrêmes (cris, émotions, distance), en mettant l’accent sur la robustesse des modèles face aux environnements bruyants et non contrôlés. Ce travail fondateur a permis d’évaluer objectivement les limites actuelles des systèmes de reconnaissance de la parole (ASR) et de détection des émotions (SER), tout en fournissant une base de données exploitable à des fins de recherche.

Le **deuxième article** a démontré la **faisabilité opérationnelle** d’un système embarqué de détection de cris en milieu dangereux (incendies), en comparant les performances de deux modèles d’apprentissage automatique : **SVM**, pour sa légèreté et sa rapidité, et **LSTM**, plus précis mais plus coûteux en ressources. Ce travail met en lumière l’importance de l’efficacité en temps réel dans des contextes où chaque seconde peut sauver une vie.

Enfin, le **troisième article** étend cette approche au **domaine médical**, en proposant un système prédictif d’épisodes d’agressivité chez les patients atteints de démence. À travers une **approche multimodale** fondée sur la fusion tardive de données audio et contextuelles, ce travail renforce la pertinence de l’analyse acoustique dans des environnements sensibles où la prévention est cruciale.

### ****Enseignements tirés et positionnement de notre projet****

Ces travaux confirment trois constats majeurs :

* **La pertinence** de l’audio comme signal primaire pour détecter des situations critiques ou des comportements agressifs.
* **La faisabilité technique** de systèmes intelligents, capables de fonctionner dans des conditions réelles et complexes.
* **L’impact sociétal** de telles solutions, qui contribuent à renforcer la sécurité des individus sans recourir à des méthodes intrusives.
* Notre projet s’inscrit dans cette continuité, en ciblant spécifiquement l’espace public urbain. Il vise à concevoir un système de **détection en temps réel des agressions verbales**, basé sur des capteurs simples (comme des microphones) et des algorithmes d’intelligence artificielle. Contrairement aux approches médicales ou robotiques, notre solution met l’accent sur la **réactivité immédiate**, par exemple par le déclenchement automatique d’alertes vers les forces de sécurité.

Par ailleurs, les apports méthodologiques (benchmark BERSt, comparaison SVM/LSTM, fusion multimodale) issus des travaux analysés serviront de **références directes** pour construire, tester et améliorer notre propre système.

# Analyse et mise en place d une solution

Cette section vise à proposer une architecture fonctionnelle complète pour la détection d’agressions verbales à l’aide de capteurs sonores, d’un traitement intelligent du signal audio et d’un système décisionnel basé sur des modèles d’intelligence artificielle. L’ensemble du système s’articule autour de plusieurs modules interconnectés, depuis la détection sonore jusqu’à la prise de décision automatisée.

## Analyse et proposition de solution

### Analyse initiale

Le problème que nous cherchons à résoudre est la détection automatique d’agressions verbales dans des environnements publics (rue, transport, etc.). L’enjeu majeur réside dans la capacité à capter un signal sonore pertinent (cris, hausse de voix, etc.), à le traiter en temps réel, et à évaluer son niveau de dangerosité.

### Schéma de solution

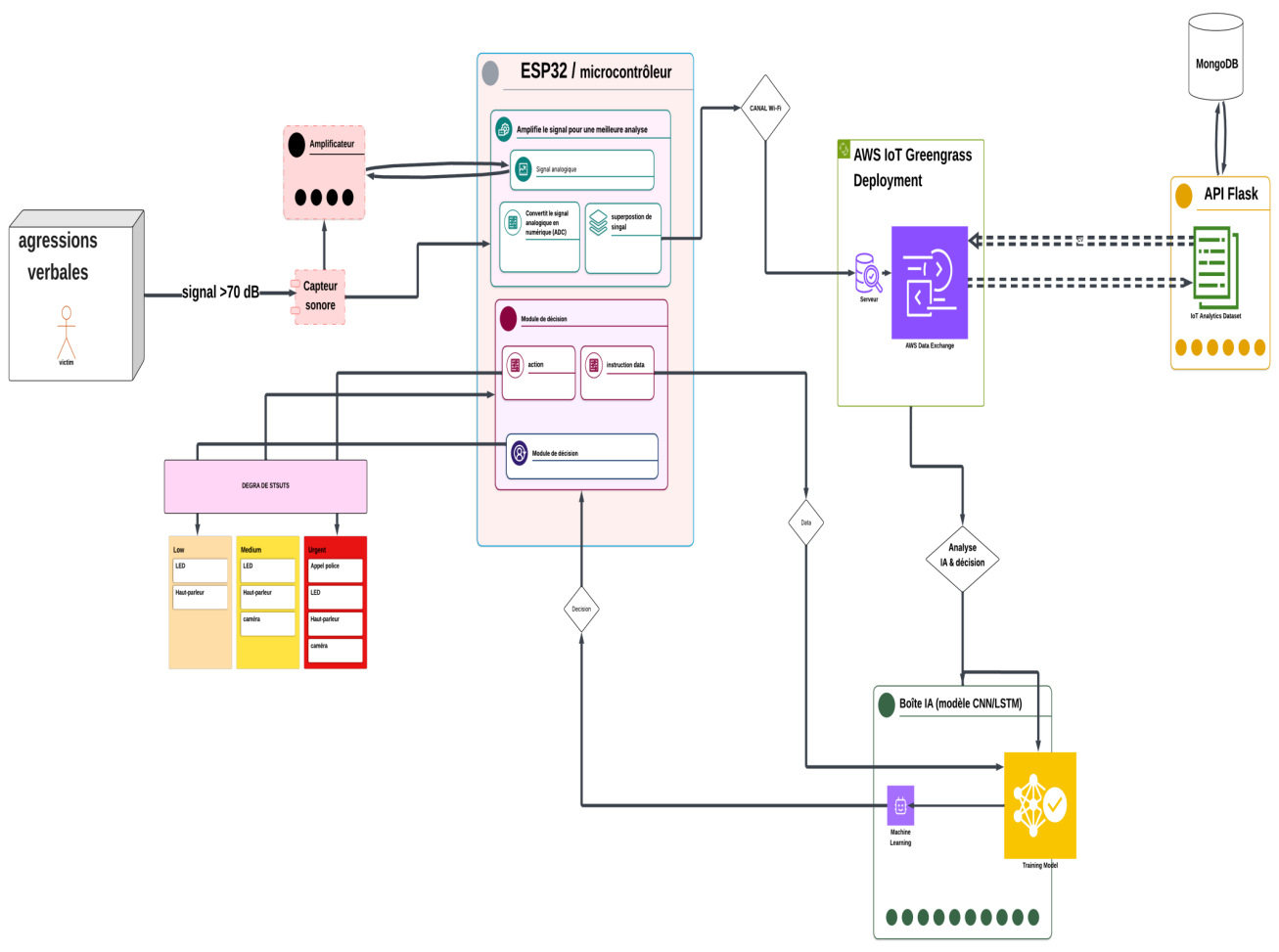
Lorsqu’une agression verbale se produit, le système commence par détecter un son dont l’intensité dépasse un seuil prédéfini (en dB) à l’aide de capteurs acoustiques. Ce signal est ensuite amplifié par un module d’amplification analogique pour améliorer la qualité d’analyse.

Ensuite, ce signal est transféré vers un microcontrôleur (ESP32 ou Arduino) doté de connectivité sans fil (Wi-Fi ou Bluetooth). Ce microcontrôleur transmet les données vers un serveur distant ou local. Le signal est alors pré-traité pour être converti en données numériques exploitables.

Ces données sont ensuite analysées par un modèle d’intelligence artificielle hébergé sur le serveur. Le modèle estime la probabilité qu’il s’agisse d’une agression verbale réelle. En fonction de cette évaluation, une ou plusieurs actions sont déclenchées :

* Allumage d’un voyant LED,
* Emission d’un signal sonore via haut-parleur,
* Déclenchement d’un enregistrement vidéo,
* Envoi d’un message ou appel automatique vers les autorités locales.

<https://lucid.app/lucidchart/b16608ea-8dcb-4c9f-8736-c57f6ade20b0/edit?viewport_loc=-691%2C-1598%2C4034%2C1598%2C0_0&invitationId=inv_dac62dfb-d4fd-46a9-ae1b-15470eb7e814>



#### Schéma Partie backend

**Composants principaux :**

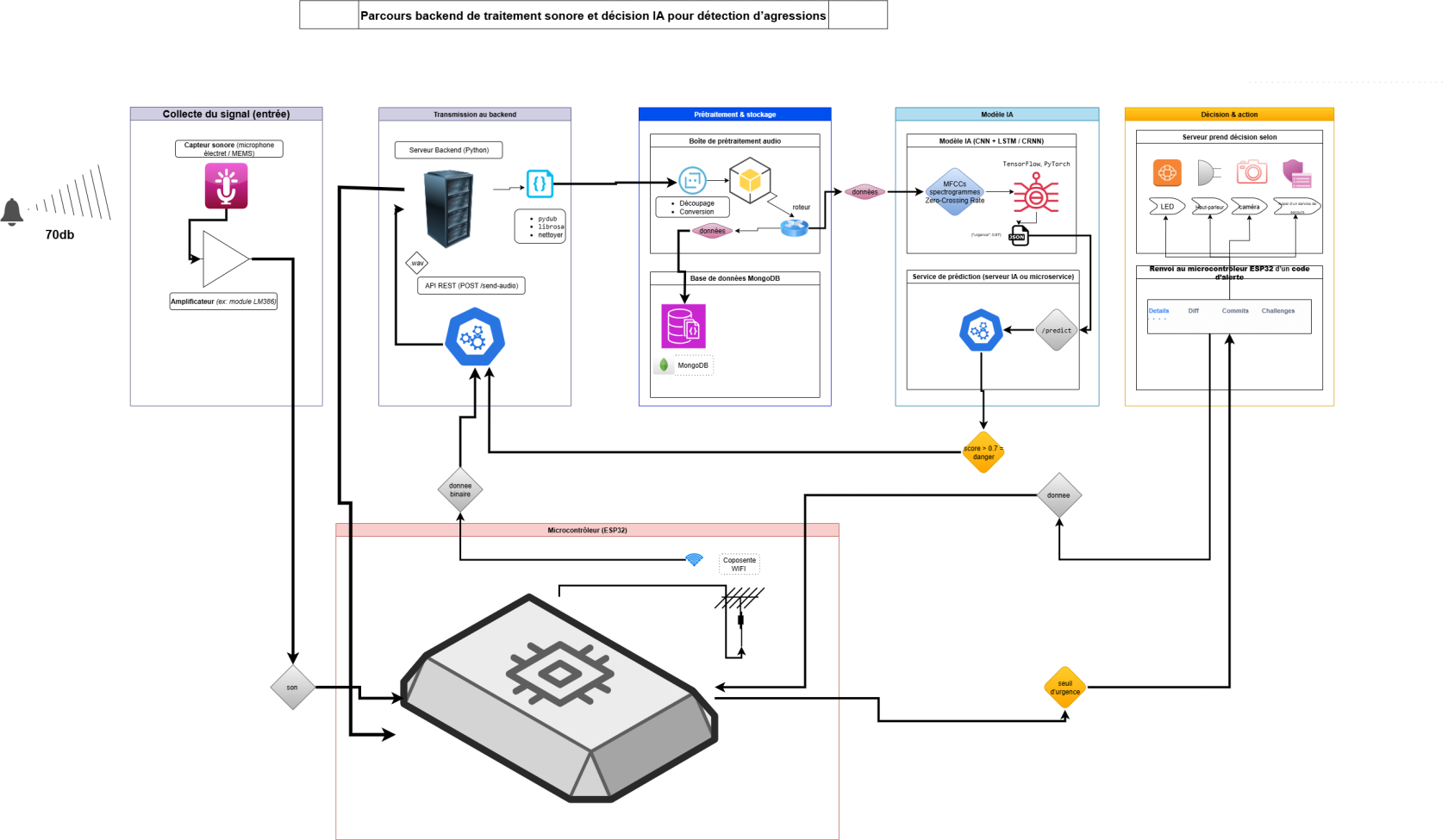
**Serveur :** (local ou AWS),

**API REST :** (Python + Flask ou FastAPI),

**Base de données :** MongoDB pour le stockage des événements détectés,

**Environnement de traitement :** (Docker ou Anaconda),

**Librairies IA :** TensorFlow, PyTorch.



**Modèle IA utilisé :**

**CNN + LSTM + CRNN :** pour combiner la reconnaissance de motifs spectrogrammiques et la dépendance temporelle.

Données entrantes sous forme de spectrogrammes log-mel ou MFCC.

#### Schéma Partie conception

**Choix des composants :**

**Microprocesseur :** ESP32, pour sa connectivité Wi-fi intégrée et sa faible consommation énergétique.

**Capteurs audio :** microphones électret ou MEMS, capables de détecter un signal >70 dB avec bonne sensibilité.

<https://wokwi.com/projects/305569599398609473>

**Modules complémentaires** :

Amplificateur audio (type MAX9814),

LED, haut-parleur, relais pour action de sortie,

### Connexion entre capteurs et microprocesseur

L’intégration se fera via :

- Entrée analogique du Microcontrôleur pour le son

- Connexion série (UART, I2C) vers les modules complémentaires,

- Transmission via Wi-fi des données vers serveur/API.

#### Codage en C++ (ESP-IDF ou Arduino IDE)

Lecture des données audio via ADC.

Condition : si dB > seuil ⇒ capture + envoi via Wi-Fi.

API POST vers /analyse-son (traitée côté Python).

Gestion des GPIO pour action LED, haut-parleur, etc.

### Bibliothèque et modèles

**Exemples de bibliothèques à utiliser** :

* **Pydub** / **Librosa** : traitement audio en Python,
* **Sounddevice** : enregistrement et lecture,
* **scikit-learn**, **Keras**, **TensorFlow**, **PyTorch** : entraînement IA,
* **OpenSMILE** : extraction d’attributs audio.

#### Analyse de précision via MATplotlib

MATLAB peut être utilisé pour la visualisation et la comparaison des performances du modèle (accuracy, F1-score, confusion matrix). On peut importer le modèle entraîné (.h5 ou .pb) et l’analyser avec des fonctions de classification.

### Base de données

**MongoDB** : stocke les événements, les métadonnées (heure, lieu, intensité),

Intégration via **PyMongo**,

**Test API :** POST /event, **GET** /events?date=today.

### Environnement de travail

**GitHub** : versioning et travail collaboratif,

**AWS EC2** : déploiement du serveur,

**Docker** : pour contenir l’API + modèle IA,

**VS Code / Jupyter** pour développement local.

### Conclusion du chapitre

Taw3 ba3ed hathi\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* en 7 line max

### Autre outil que nos aident

# Bibliographie

[https://arxiv.org/html/2505.00059v1?](https://arxiv.org/html/2505.00059v1?utm_source)

[https://www.aimodels.fyi/papers/arxiv/bersting-screams-benchmark-distanced-emotional-shouted-speech?](https://www.aimodels.fyi/papers/arxiv/bersting-screams-benchmark-distanced-emotional-shouted-speech?utm_source)

<https://www.mdpi.com/2076-3417/15/10/5351>

<https://www.mdpi.com/2076-3417/11/18/8425>

<https://www.adafruit.com/product/1063>

<https://learn.adafruit.com/adafruit-i2s-mems-microphone-breakout>

<https://docs.espressif.com/projects/esp-idf/en/latest/esp32/>

<https://www.raspberrypi.com/documentation/>

<https://dev.mysql.com/>

<https://www.postgresql.org/docs/>

<https://aws.amazon.com/fr/>

<https://www.mongodb.com/docs/>

<https://github.com/>

<https://towardsdatascience.com/using-lstm-in-twitter-sentiment-analysis-a5d9013b523b/>

<https://towardsdatascience.com/neural-networks-for-real-time-audio-stateful-lstm-b534babeae5d/>

<https://github.com/Alec-Wright/CoreAudioML/blob/823a4727f4578aa434e715eae302e0e930576074/dataset.py>

<https://www.tensorflow.org/tutorials/audio/simple_audio?hl=fr>

<https://towardsdatascience.com/using-lstm-in-twitter-sentiment-analysis-a5d9013b523b/>

<https://towardsdatascience.com/neural-networks-for-real-time-audio-stateful-lstm-b534babeae5d/>

<https://github.com/Alec-Wright/CoreAudioML/blob/823a4727f4578aa434e715eae302e0e930576074/dataset.py>

<https://arxiv.org/abs/1711.00520>

<https://wokwi.com/projects/305569599398609473>

Presentation et motivation

Metaderoge et demache

Mise en avor et exprementation

Apors

Contrubtion

Syenethse

Mise en avre de la platforme de la gestion