
Hermès

Une solution pour analyser et prédire la pollution urbaine



Réalisé par :

Arthur Strimbu-Lee, Samuel Labignette

Supervisé par :

Jordan Barbier

"Science sans conscience n'est que ruine de l'âme." - Rabelais

Table des matières

1	Introduction	2
2	Le boîtier - <i>Pandore</i>	3
2.1	Méthodologie	3
2.1.1	3 gaz essentiels	3
2.1.2	3 gaz auxiliaires	3
2.1.3	Dangers pour la santé et l'environnement	3
2.2	Capteurs	3
2.3	Boîte	4
2.3.1	Conception et Prototypage	4
2.3.2	Analyse des données (Mission 1)	6
3	Le drone - <i>Hermès</i>	7
3.1	Méthodologie	7
3.2	Conception	7
3.3	Résultat	9
4	Analyse Sonore	10
4.1	Méthodologie	10
4.2	Traitement et nettoyage audio	10
4.2.1	Réduction du bruit avec Audacity - soustraction spectrale	10
4.2.2	Réduction du bruit avec Python - filtre de Wiener	15
5	Modèle de prédiction de concentration en polluants - <i>Pythie</i>	18
5.1	Méthodologie	18
5.2	Feature engineering	18
5.3	Choix de structure de modèle	19
6	Résultats et site internet	21
6.1	Bilan de Pandore, Hermès et Pythie	21
6.2	Notre site internet	21
A	Le système dBFS et le RMS	23

Remerciements

Nous tenons à remercier toutes les personnes qui ont rendu le Projet Hermès possible. Tout d'abord, nous remercions la Quintet Private Bank pour son soutien financier. Nous souhaitons exprimer notre gratitude tout particulièrement à M. Gianmarco Gioia, directeur au sein de la Quintet Private Bank, qui a cru en ce projet et en notre capacité à le concrétiser. Nous remercions également nos professeurs M. Devingt et M. Barbier, qui nous ont aidés à comprendre et à assimiler les aspects mathématiques.

1 Introduction

Les données de santé mondiales révèlent une réalité alarmante : 7,9 millions de décès prématurés par an sont attribués à la pollution atmosphérique, qui tue désormais plus de personnes que le tabac dans le monde. Cette urgence environnementale est la force motrice derrière la création d'Hermès, un drone autonome conçu pour surveiller la qualité de l'air grâce à un système économique mais très fiable. Le choix d'un drone comme plateforme de surveillance est stratégique, offrant plusieurs avantages uniques. L'analyse non-humaine permet aux drones de détecter et d'entrer dans des zones potentiellement dangereuses sans risquer la santé humaine. En tant que station d'analyse mobile, un drone transforme un capteur statique en laboratoire mobile, permettant d'évaluer facilement la qualité de l'air dans diverses zones, comme les zones résidentielles, les autoroutes et les parcs, en utilisant un protocole de mesure cohérent. Enfin, la détection en altitude permet une surveillance aérienne pour la collecte de données à différentes hauteurs, ce qui évite les perturbations au niveau du sol et fournit un profil 3D plus complet des polluants urbains.

2 Le boîtier - *Pandore*

2.1 Méthodologie

La qualité de l'air peut être évaluée en fonction de nombreux gaz, tous différents. Cependant, certains sont bien moins nécessaires à analyser que d'autres. Ainsi, nous nous sommes concentrés sur 6 gaz principaux :

2.1.1 3 gaz essentiels

- **Les PM10 et PM2.5** : désignant respectivement la taille des particules étudiées en micromètres. Les PM10 sont issues majoritairement du chauffage au bois, du trafic routier (pneus, freins) et des chantiers. Les PM2.5 sont essentiellement liées au chauffage au bois, à l'industrie, à l'agriculture et aux transports routiers. Ces deux tailles de particules en suspension constituent la base d'une étude sur la qualité de l'air.
- **Le CO2** (dioxyde de carbone) : issu principalement de la respiration humaine et des combustions, il est aussi le principal gaz à effet de serre responsable du réchauffement climatique.

2.1.2 3 gaz auxiliaires

- **Le CO** (monoxyde de carbone) : gaz extrêmement dangereux qui peut être mortel en milieu confiné. Il tire son origine d'une combustion incomplète.
- **Le NH3** (ammoniac) : agit principalement comme un précurseur dans la formation de particules fines secondaires. Il est principalement issu des activités agricoles.
- **Le NO2** (dioxyde d'azote) : gaz polluant brun-rougeâtre, principalement issu de la combustion (trafic routier diesel, industrie), fortement présent en milieu urbain.

2.1.3 Dangers pour la santé et l'environnement

Gaz	Dangers pour la santé et l'environnement
PM10	Irritent les voies respiratoires, aggravent l'asthme et les maladies pulmonaires.
PM2.5	Très dangereuses car elles pénètrent profondément dans le sang, augmentent les risques cardiaques.
CO2	À forte concentration : maux de tête, fatigue, perte de concentration et asphyxie.
CO	Très toxique : intoxication, perte de connaissance et mort (empêche le transport de l'oxygène).
NH3	Irrite fortement les yeux, la peau et les voies respiratoires.
NO2	Irrite les poumons, aggrave l'asthme et diminue la fonction respiratoire.

Table 1 – Récapitulatif des dangers des gaz étudiés

2.2 Capteurs

La boîte de Pandore est composée de **4 capteurs majeurs**, reliés à un **Raspberry Pi 4** qui joue le rôle de mini-ordinateur de contrôle :

- **Le capteur Nova PM (SDS011)** : fonctionne sur le principe de la diffusion laser pour mesurer la concentration de particules fines (PM2.5 et PM10).

- Le **MiCS-6814** : capteur compact mesurant le CO, le NO2 et le NH3. Il utilise trois éléments de détection à base d'oxyde métallique. La lecture nécessite un convertisseur analogique et une calibration complexe, ce qui peut générer certaines valeurs aberrantes.
- Le **SCD-41** : capteur de CO2, de température et d'humidité de haute précision.
- Le **Microphone** : connecté en Bluetooth, il permet le développement du système *Oracle*.

2.3 Boîte

2.3.1 Conception et Prototypage

Le premier prototype était divisé en deux parties distinctes s'assemblant l'une dans l'autre (parties inférieure et supérieure).

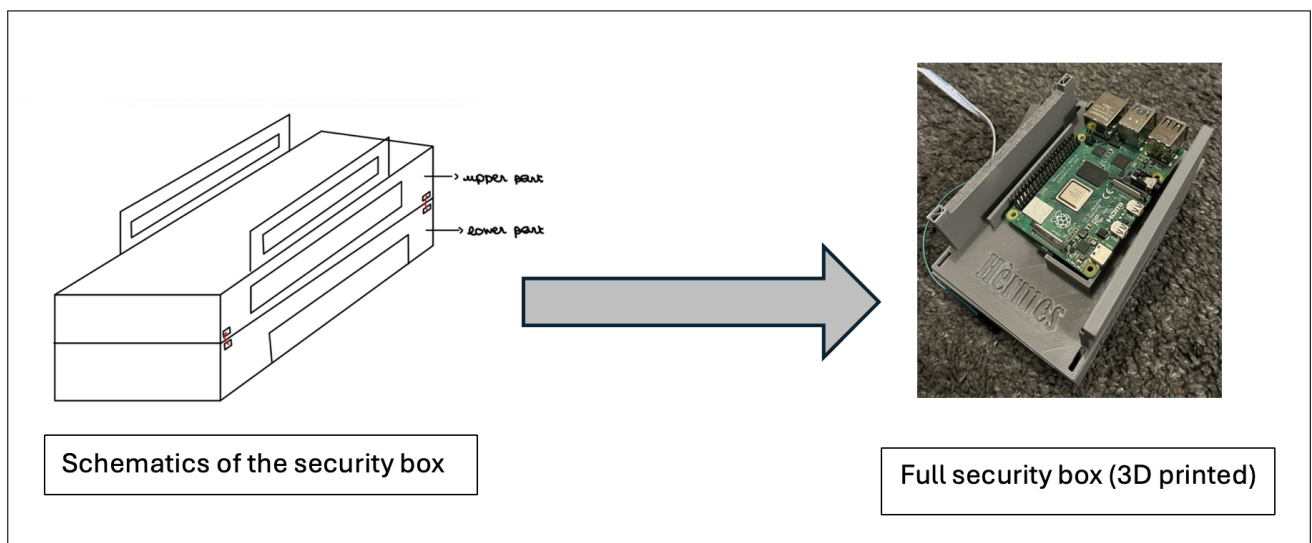


Figure 1 – La boîte de sécurité initiale

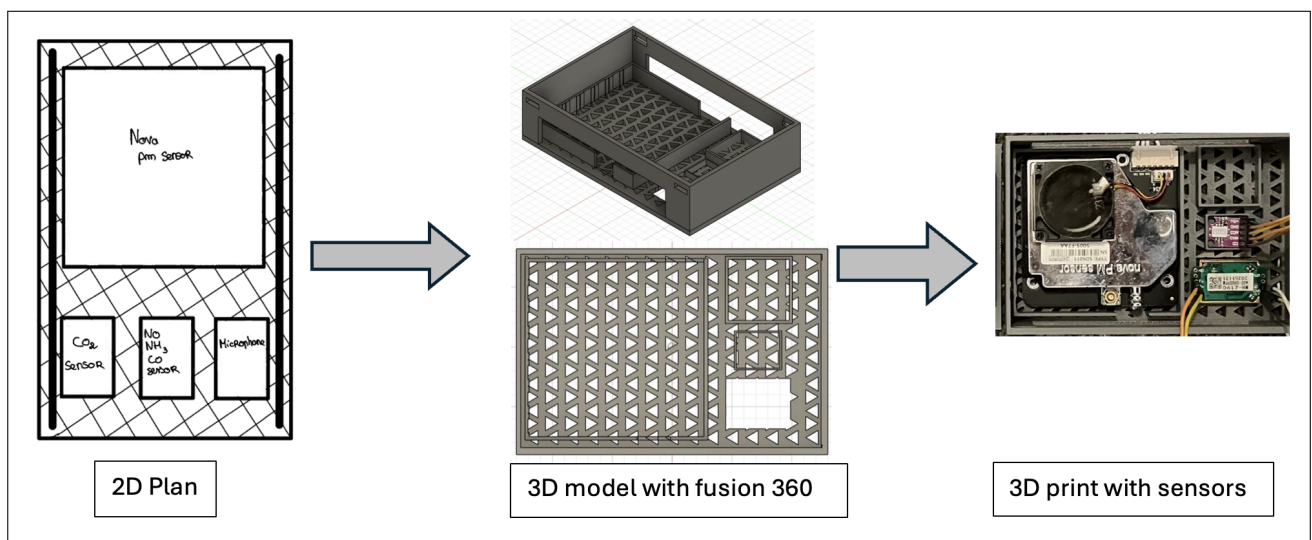


Figure 2 – La partie inférieure

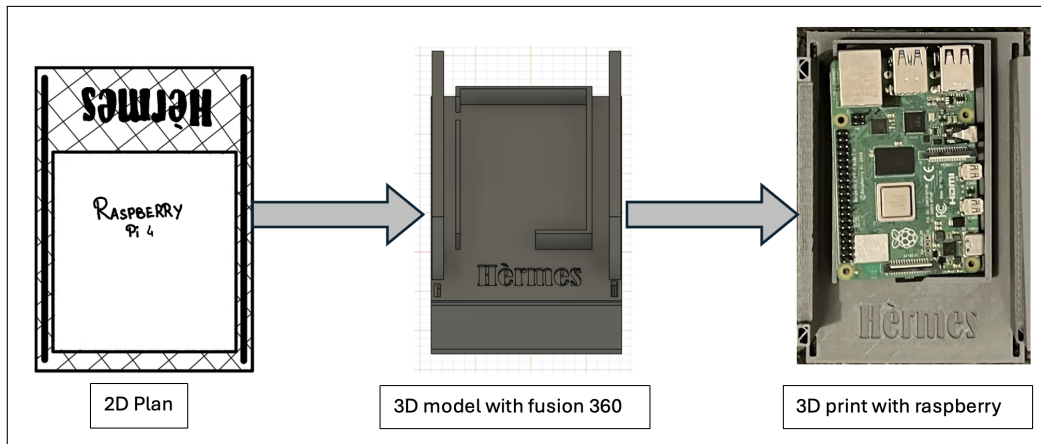


Figure 3 – *La partie supérieure*

Pour le produit final, nous avons opté pour un changement radical du design en assemblant ces parties en **une seule et même pièce**, avec un couvercle contenant le support batterie.

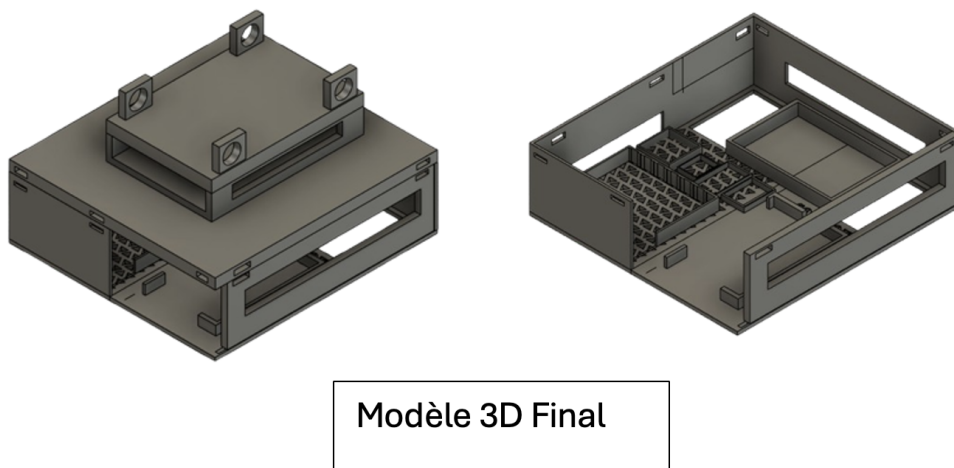
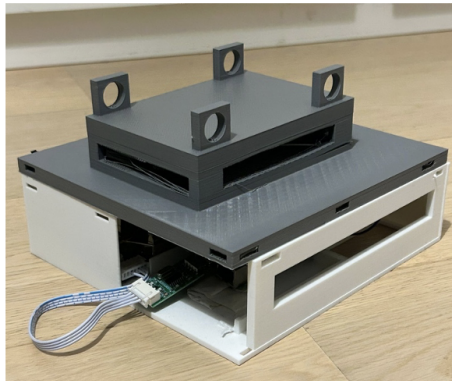


Figure 4 – *Modèle 3D final*



Boîte de Pandore finale imprimée en 3D

Figure 5 – Boîte de Pandore finale

Ce choix se justifie par la nécessité d'ajouter une *breadboard* pour faciliter les câblages. Ce design offre trois possibilités d'utilisation :

1. Analyse à **point fixe** sur support spécial pour des sessions de plusieurs heures.
2. Utilisation **nomade** portée à la main pour des analyses urbaines.
3. Fixation au **drone Hermes** pour des analyses en milieu extrême ou en altitude.

2.3.2 Analyse des données (Mission 1)

Nous avons testé les capacités du boîtier lors de l'analyse d'une zone verte urbaine. Les résultats sont satisfaisants :

- Concentrations en PM élevées près des axes routiers.
- Concentrations en CO₂ plus basses au niveau des espaces végétaux.
- *Note* : Les valeurs de CO sont restées incohérentes suite à un défaut de calibration.

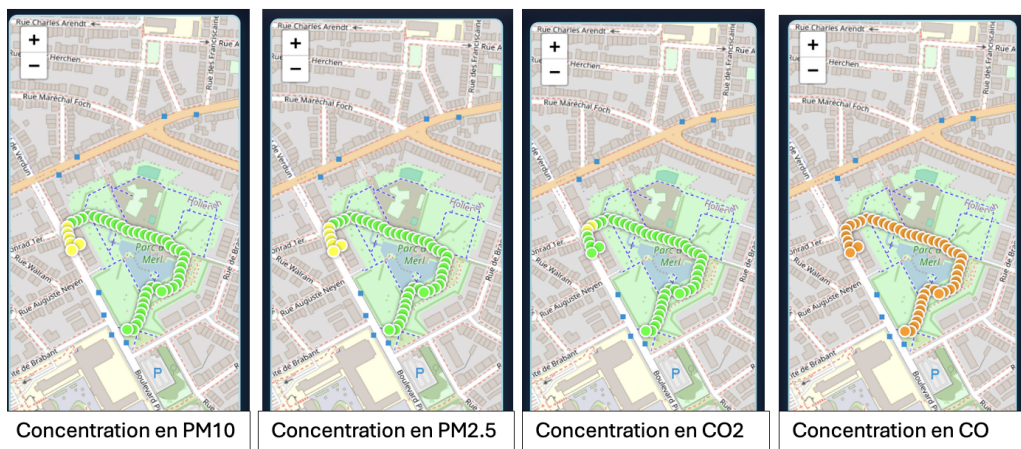


Figure 6 – Titre ou description du graphique

L'ensemble des résultats est disponible sur notre site web : https://vincenzosbrain.github.io/Website_projet_hermes/

3 Le drone - *Hermès*

3.1 Méthodologie

Imaginons qu'un incendie ait eu lieu dans un entrepôt. Du monoxyde de carbone (CO) se forme. S'il fallait s'aventurer dans cet entrepôt, savoir s'il est dangereux pour l'Homme est crucial. C'est pourquoi Pandore peut être utile. Cependant, ce n'est qu'une boîte. Il nous a donc fallu trouver une idée pour déplacer Pandore, sans intervention humaine directe. L'idée d'un drone nous est alors apparue. L'idée d'Hermès est née. Hermès doit satisfaire plusieurs critères que voici :

- Être capable de supporter une masse de 1,6 kg
- Avoir une autonomie de 20 minutes
- Être piloté à distance par une radiocommande
- Enregistrer de l'audio

Dans la sous-partie suivante, nous allons vous présenter notre conception d'Hermès.

3.2 Conception

Être capable de supporter une masse de 1,618 kg

Pour supporter la masse du drone, les moteurs doivent être choisis avec soin. Nous devons déterminer la valeur optimale du kV (la constante de vitesse du moteur) en considérant que notre batterie fournit 11,1V.

La force de poussée F nécessaire pour faire voler notre drone est calculée par :

$$F = m \cdot g \quad (1)$$

avec :

- m : la masse du drone de 1,618kg
- g : la gravité standard arrondie à $9,81m/s^2$

Nous avons alors $F \approx 15,9N$. La poussée totale pour faire décoller et voler notre drone est 2 fois supérieure à la force de poussée (poids). Comme notre drone possède 4 moteurs, chacun d'eux doit contribuer à hauteur de $7,95N$ comme le montre le calcul ci-dessous :

$$F = \frac{15,9 \cdot 2}{4} = 7,95N$$

La formule de la poussée statique suivante nous aidera à déterminer le nombre de rotations par minute (RPM) que notre drone doit effectuer :

$$F = C_T \cdot \rho \cdot n^2 \cdot D^4 \quad (2)$$

avec :

- F : Poussée statique
- C_T : Coefficient de poussée de 0,1 pour notre hélice
- ρ : Densité de l'air, environ $1,225 \text{ kg/m}^3$ au niveau de la mer
- n : Vitesse de rotation de l'hélice, mesurée en révolutions par seconde (tr/s)
- D : Diamètre de l'hélice de $0,254m$

Après manipulation de l'équation 2, nous obtenons :

$$n = \sqrt{\frac{F}{D^4 \cdot C_T \cdot \rho}} \quad (3)$$

En multipliant le résultat par 60 pour obtenir les RPM, nous arrivons à :

$$RPM \approx 7500$$

Le kV introduit précédemment est exprimé par :

$$kV = \frac{RPM}{V} \quad (4)$$

avec :

— $RPM = 7500$

— $V = 11,1$

Nous obtenons une valeur de $675kV$. Cependant, un moteur doit se situer à 70 % de sa puissance lors d'une poussée stationnaire (vol stationnaire). Le kV dont nous avons besoin est donc :

$$kV = \frac{675}{0,7} \approx 960$$

En conclusion, nous pourrions utiliser un moteur de $930kV$, car c'est la valeur de kV la plus proche que nous avons trouvée sur le marché.

Avoir une autonomie de 20 minutes

Hermès doit voler au moins 20 minutes pour couvrir une large zone de mesure en une seule mission. Pour y parvenir, nous utilisons une batterie Li-Po 3S de 4200 mAh. Cette batterie offre un bon compromis entre poids et puissance. En effet, de nombreuses batteries sont bien trop lourdes pour être utilisées sur ce type de drone. La trajectoire de vol est pré-programmée via le contrôleur de vol Pixhawk (Figure 7). Cette autonomie devrait réduire l'intervention humaine et accroître la précision de la mission.



Figure 7 – Interface de vol autonome Pixhawk

Être piloté à distance par une radiocommande

Bien qu'Hermès soit un drone autonome, il reste essentiel de disposer d'une option manuelle en cas de problème (perte de signal / urgences) ou lors des phases de test. La condition principale est la compatibilité avec le système Pixhawk. Le prix doit également être pris en compte, c'est pourquoi nous avons choisi une radiocommande simple répondant à nos besoins essentiels : la Flysky i6A.

Enregistrer de l'audio

En plus de la surveillance de la pollution atmosphérique, Hermès intègre un microphone (Figure 8) pour suivre la pollution sonore. Le micro enregistre les niveaux de bruit ambiant pendant le vol, et les données audio sont sauvegardées dans des fichiers pour une analyse ultérieure. Ces mesures complètent les données sur la qualité de l'air afin d'aider à identifier les sources potentielles de pollution.



Figure 8 – *Notre microphone DJI Mic mini*

3.3 Résultat

Après de nombreux essais et échecs, nous avons enfin un drone fonctionnel (Figure 9).



(a) *Vue de côté*



(b) *Vue du dessus*

Figure 9 – *Le drone Hermès finalisé*

4 Analyse Sonore

4.1 Méthodologie

Maintenant, nous avons un drone fonctionnel qui peut nous fournir des données pertinentes et utilisables. L'objectif de cette partie est de développer un modèle d'IA qui prédit les concentrations de polluants uniquement en fonction du son, capturé par notre capteur sonore. Cependant, avant d'entraîner un modèle d'IA, nous avons besoin de données, évidemment. Les fichiers sonores que nous avons sont des données brutes et ne conviennent pas pour entraîner un modèle d'IA. Par conséquent, nous devons faire de l'ingénierie de caractéristiques.

4.2 Traitement et nettoyage audio

4.2.1 Réduction du bruit avec Audacity - soustraction spectrale

Nous voulons seulement étudier le son de l'environnement comme les voitures, les conversations humaines, le vent, etc. Nous ne voulons pas que les sons parasites produits par notre drone fassent partie de nos données pour le modèle d'IA. Le son constant que notre drone produit est appelé *bruit*, et le son de la ville que nous voulons capturer est appelé *signal utile*.

Dans cette partie, nous allons nous amuser à tester l'outil de *Réduction du bruit* d'Audacity sur un fichier audio où la 5ème Symphonie de Beethoven est le signal utile et un climatiseur (A.C.) constant est le bruit.

D'abord, nous devons importer le fichier audio que nous voulons modifier et nous l'appellerons le signal source. Ensuite, nous devons importer un échantillon du signal du climatiseur seul. L'objectif est d'obtenir à la fin le signal utile. Le principe de l'outil de réduction du bruit dans Audacity (Figure 10) est d'extraire le *Profil de bruit* de l'échantillon et de le supprimer du signal source. Le *Profil de bruit* qu'Audacity possède caractérise numériquement le signal constant du climatiseur seul. Concrètement, Audacity effectue une soustraction spectrale du *Profil de bruit* sur le signal source. Selon Steven F. Boll [1], nous pouvons estimer le signal utile comme suit :

$$|\hat{S}(f)| = \max(|Y(f)| - \alpha \cdot |N(f)|, 0) \quad (5)$$

où

- $|\hat{S}(f)|$ est le spectre de magnitude estimé du signal utile
- $|Y(f)|$ est le spectre de magnitude du signal source
- α est le facteur de soustraction exprimé par $10^{\frac{dB}{20}}$
- $E[|N(f)|]$ est la magnitude moyenne du bruit (le *Profil de bruit* d'Audacity)

Si le coefficient α est trop grand, la valeur de $\hat{S}(f)$ peut être négative. Cependant, Boll a considéré que pour toute fréquence f , si $|Y(f)| - \alpha \cdot |N(f)| < 0$, alors $|\hat{S}(f)|$ serait fixée à 0. C'est

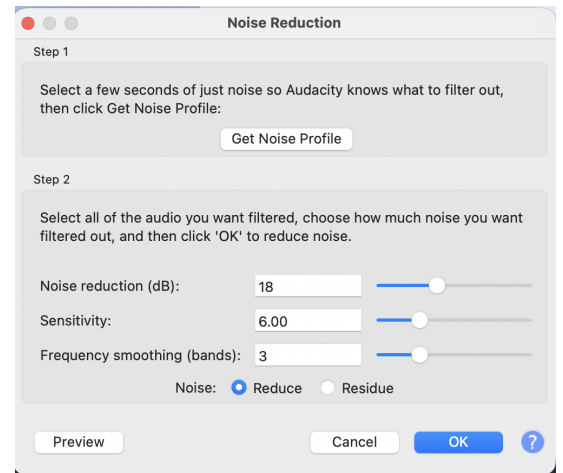


Figure 10 – Fenêtre de l'outil de réduction du bruit dans Audacity

pourquoi la fonction $\max(|Y(f)| - \alpha \cdot |N(f)|, 0)$ renvoie soit la valeur de $|Y(f)| - \alpha \cdot |N(f)|$ soit 0.

Il y a 3 paramètres que nous pouvons modifier pour rendre la réduction du bruit plus efficace :

1. Réduction du bruit (en dB)
2. Sensibilité
3. Lissage fréquentiel (nombre de bandes)

Réduction du Bruit

Le niveau de réduction du bruit est le paramètre principal, car il détermine la quantité de magnitude de bruit qui sera soustraite du spectre de puissance du signal original. Idéalement, nous voulons nous débarrasser de tout le bruit produit par le climatiseur. C'est pourquoi, intuitivement, nous avons pensé que régler la réduction du bruit à la valeur la plus élevée possible était une bonne idée. Nous avons testé notre hypothèse et comparé deux réductions de bruit avec deux valeurs différentes :

- 8 dB, une valeur courante utilisée dans la réduction du bruit
- 48 dB, le maximum autorisé par Audacity.

Avant d'appliquer tout changement sur le signal source, nous devons savoir à quoi ressemblent les spectres de fréquence du bruit et de l'ensemble du signal source. Nous pouvons voir sur la Figure 11 que le bruit est concentré autour de 120 Hz et son RMS est de -44,1 dBFS. La Figure 12 est évidemment plus riche en fréquences et le RMS du signal source est de -38,3 dBFS. (Annexe A)

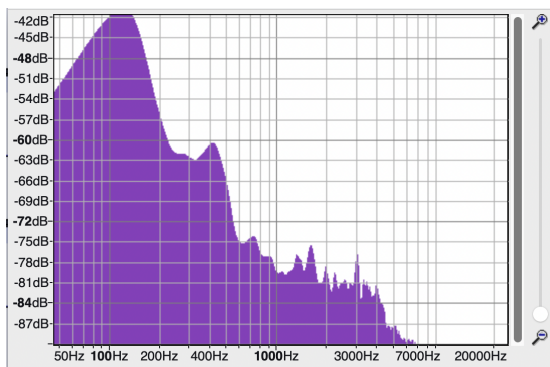


Figure 11 – *Spectre de fréquence du bruit seul*

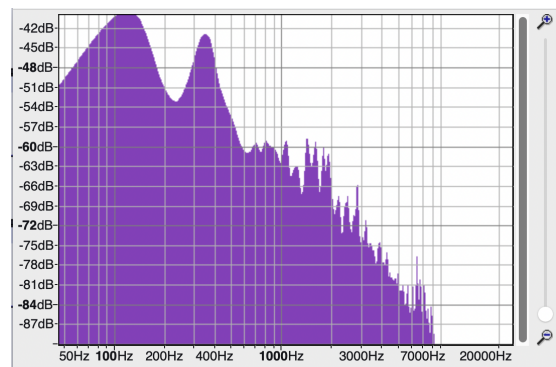


Figure 12 – *Spectre de fréquence du signal source*

Les deux graphiques suivants sont les spectres de fréquence de l'audio avec les deux intensités de réduction du bruit différentes appliquées. Sur la Figure 14, nous pouvons voir des lacunes dans le spectre de fréquence. Cela signifie que la réduction du bruit était trop agressive et a complètement supprimé certaines fréquences du signal source, conformément à (5). Supprimer complètement certaines fréquences signifie perdre beaucoup d'informations, même sur le son utile, et nous voulons éviter cela pour notre modèle d'IA plus tard. Sur la Figure 13, nous pouvons voir que la réduction du bruit avec 8 dB est beaucoup moins agressive que celle avec 48 dB. Pourtant, elle est encore assez destructrice lorsque la fréquence est supérieure à 5000 Hz.

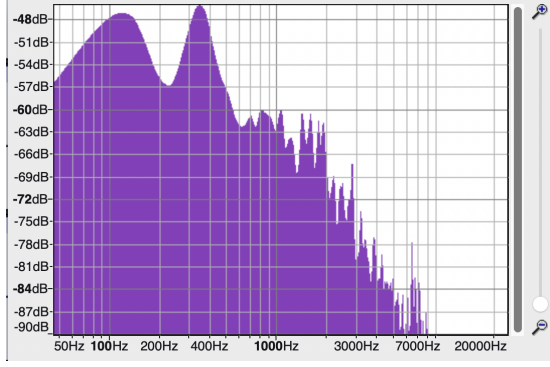


Figure 13 – Spectre de fréquence de l'audio avec une réduction du bruit de 8dB appliquée

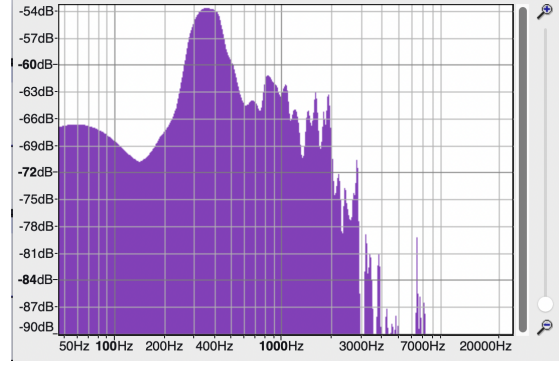


Figure 14 – Spectre de fréquence de l'audio avec une réduction du bruit de 48dB appliquée

Cette agressivité est due au coefficient α qui dépend directement de la valeur en dB que nous mettons dans le paramètre de réduction du bruit d'Audacity (Figure 10). Par conséquent, nous devons trouver une valeur de dB qui ne fera jamais passer aucune fréquence du signal utile en dessous de 0. Cela signifie que le spectre de magnitude doit toujours être plus grand que le spectre de magnitude du bruit que nous supprimons. Mathématiquement, cela signifie :

$$|Y(f)| > \alpha \cdot E[|N(f)|] \quad (6)$$

Un concept clé à connaître, avant de manipuler (6), est le Rapport Signal sur Bruit (SNR). Le SNR mesure la clarté d'un signal. Plus la valeur est grande, plus le signal est propre. À l'inverse, plus le SNR est faible, plus le signal est bruité. Le SNR peut être représenté dans deux échelles : l'échelle linéaire ($SNR_{linéaire}$) qui quantifie le rapport de magnitude comme une valeur sans dimension, et l'échelle logarithmique ($SNR_{logarithmique}$) qui quantifie le rapport de puissance exprimé en décibels. Après une première manipulation de (6), nous avons :

$$\alpha < \frac{|Y(f)|}{E[|N(f)|]}, \quad \text{avec } \frac{|Y(f)|}{E[|N(f)|]} = SNR_{linéaire}$$

Nous savons d'après (5) que $\alpha = 10^{\frac{dB}{20}}$ avec dB la valeur que nous mettons sur Audacity (Figure 10). Après isolation de dB dans l'équation, nous trouvons :

$$dB < 20 \cdot \log_{10}(SNR_{linéaire})$$

Comme indiqué précédemment, le SNR est défini sur l'échelle linéaire et sur l'échelle logarithmique. La relation entre les deux est la suivante :

$$SNR_{logarithmique} = 20 \cdot \log_{10}(SNR_{linéaire}) \quad (7)$$

Par conséquent, nous pouvons affirmer que la valeur dB mise dans le paramètre de réduction du bruit d'Audacity ne doit pas dépasser le $SNR_{logarithmique}$. Cependant, comment pouvons-nous le mesurer ? La formule pour calculer le $SNR_{logarithmique}$ est beaucoup plus simple et seule une petite manipulation dans Audacity nous permet de le déterminer. La relation a été établie par l'Institute of Electrical and Electronics Engineers (IEEE).

$$SNR_{logarithmic} = P_{signal,dBFS} - P_{noise,dBFS} \quad (8)$$

avec :

- $P_{signal,dBFS}$ la puissance du signal en dBFS
- $P_{noise,dBFS}$ la puissance du bruit en dBFS

Les valeurs de puissance dont nous avons besoin sont les valeurs RMS que nous avons mesurées précédemment. Nous pouvons maintenant associer $P_{signal,dBFS}$ avec -38,3 dBFS et $P_{noise,dBFS}$ avec -44,1 dBFS. En utilisant (8), la valeur $SNR_{logarithmic}$ vient naturellement :

$$SNR_{logarithmic} = -38,3 - (-44,1)$$

$$SNR_{logarithmic} = 5,8\text{dB}$$

Maintenant, nous savons que la valeur maximale que nous pouvons mettre dans le paramètre de réduction du bruit est de 5,8 dB. Audacity n'accepte que des valeurs entières, donc la valeur réelle prise en compte par Audacity était $[5,8] = 5\text{dB}$. Essayons maintenant la réduction du bruit lorsque nous mettons 5 dB dans ce paramètre. Les résultats sont très similaires à la réduction du bruit de 8 dB (Figure 13). Lorsque nous comparons le spectre de la réduction du bruit de 5 dB (Figure 15) à celui de la réduction du bruit de 8 dB (Figure 13), la forme est à peu près la même et nous avons toujours une perte notable de puissance dans les fréquences proches de 7000 Hz alors que nous ne devrions pas.

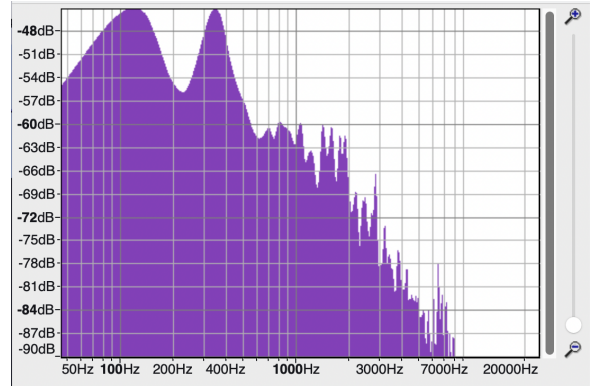
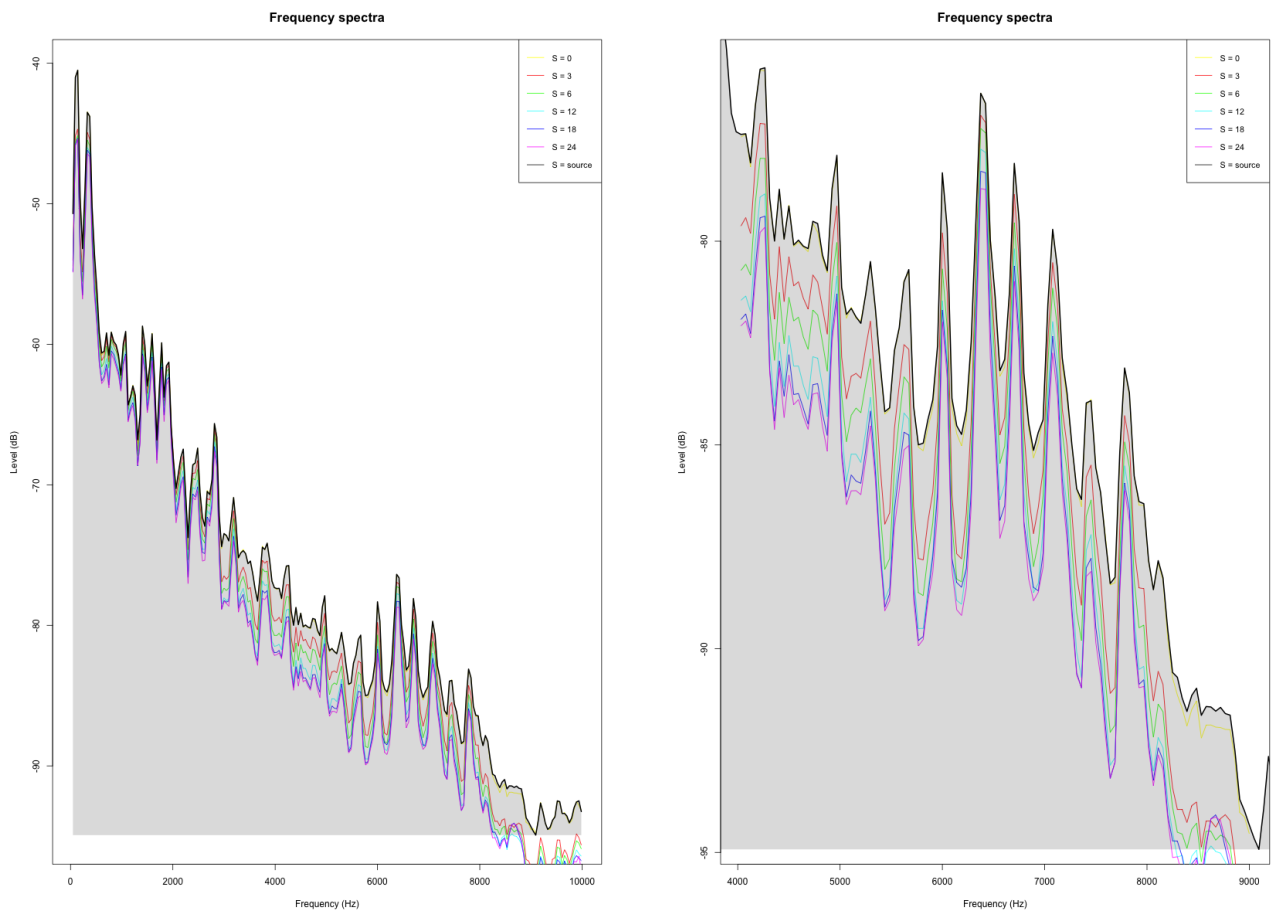


Figure 15 – Spectre de fréquence de l'audio avec une réduction du bruit de 5 dB appliquée

Sensibilité

Le paramètre de sensibilité est utilisé par Audacity pour déterminer si le signal qu'il traite est du bruit ou du signal utile. Plus la sensibilité est élevée, plus Audacity percevra le signal comme du bruit et donc plus il réduira chaque fréquence. Ce que nous voulons faire dans cette partie est de régler la sensibilité pour qu'elle ne réduise pas le signal utile. Nous avons déterminé empiriquement la valeur de sensibilité la plus pratique qui rendait l'audio ni trop bruité ni trop détruit dans les fréquences supérieures du signal source. Pour les 5 dB constants de réduction du bruit, nous avons réduit notre audio avec différentes valeurs de sensibilité S. Ce que nous voulons est un spectre où les fréquences supérieures sont préservées dans l'audio.



(a) Spectres de l'audio avec différentes sensibilités S appliquées

(b) Spectres agrandis affichant les fréquences entre 4000 Hz et 9000 Hz

Figure 16 – Spectres de notre audio lorsque nous modifions le paramètre de Sensibilité

La Figure 16 affiche les différents spectres de notre audio après avoir utilisé la réduction du bruit en variant le paramètre de sensibilité. La zone grise représente le spectre du signal source sans aucune modification appliquée. Nous avons dit que nous voulions préserver les fréquences supérieures autant que possible. Sur la version agrandie des spectres (Figure 16b), nous pouvons voir que pour une sensibilité supérieure à 3, le signal est particulièrement réduit. Le spectre que nous voyons avec une sensibilité de 3 (la courbe rouge) est la limite acceptable pour supposer que le signal utile n'est pas trop détruit. Mais nous observons toujours que pour les fréquences supérieures à 7000 Hz, le signal est drastiquement réduit alors qu'il ne devrait pas l'être. D'autre part, avec une sensibilité de 0¹, le signal n'est pas perceptiblement réduit du tout et nous ne voulons évidemment pas cela. Cependant, lorsque nous écoutons l'audio réduit avec une sensibilité de 3, le son du climatiseur est présent et n'est pas beaucoup réduit.

Lissage fréquentiel

Le paramètre de lissage fréquentiel ne pourrait pas être mieux nommé. Il lisse simplement

1. Ici nous disons 0, mais c'est un abus de langage car la valeur minimale de sensibilité d'Audacity est de 0.001.

les fréquences pour éviter le bruit musical et rend l'audio agréable à écouter. Bien sûr, dans notre cas avec Beethoven, nous voulons un audio agréable et clair, mais dans notre perspective d'IA, nous ne l'écouterons pas donc nous n'en avons pas besoin. L'appliquer ne ferait que perturber davantage le signal source.

Nous avons conclu que l'outil de réduction du bruit dans Audacity n'est pas adapté pour nous car il nous place dans l'un des deux cas indésirables : soit la réduction du bruit ne réduit pas le bruit suffisamment dans l'audio, soit elle réduit convenablement le bruit mais détruit significativement le signal utile que nous voulons préserver.

4.2.2 Réduction du bruit avec Python - filtre de Wiener

Nous venons de voir que nous ne pouvons pas nous fier à l'outil de réduction du bruit d'Audacity. Le problème en soi n'est pas Audacity, mais la méthode qu'il utilise qui est jugée trop agressive pour notre signal source. C'est un outil très intéressant pour une réduction du bruit de base, surtout pour les particuliers, mais avec lui, nous perdons soit trop de signal utile, soit la réduction du bruit n'est pas suffisamment efficace.

Dans cette partie, nous explorerons une autre méthode pour réduire le bruit dans notre signal source tout en préservant un maximum de signal utile. Au lieu de réduire le bruit de manière homogène dans le signal source, la méthode que nous présenterons utilise une méthode statistique pour faire varier la réduction du bruit. C'est l'utilisation d'un filtre de Wiener.

Avant d'appliquer un filtre de Wiener, le signal doit être converti en une représentation appropriée. Un fichier sonore est typiquement représenté comme une amplitude au fil du temps, ce qui n'est pas idéal pour la réduction du bruit. Au lieu de cela, nous opérons dans le domaine fréquentiel en estimant la Densité Spectrale de Puissance (PSD). L'outil utilisé pour passer du domaine temporel au domaine fréquentiel est la Transformée de Fourier.

$$\hat{f}(\xi) = \int_{-\infty}^{+\infty} f(t)e^{-2\pi i t \xi} dt \quad (9)$$

où :

- ξ représente la fréquence (en Hz)
- $\hat{f}(\xi)$ est le spectre de fréquence du signal
- $f(t)$ est le signal original défini dans le domaine temporel.

Cependant, numériquement, un signal n'est par définition plus continu. Il a été discrétisé par le microphone lorsqu'il a enregistré le son. Cela signifie que nous avons un nombre fini de points que nous appelons échantillons (Figure 17) à analyser et non une infinité que l'ordinateur ne pourrait pas traiter. La Transformée de Fourier classique n'est pas applicable dans notre cas. Au lieu de cela, nous utiliserons la Transformée de Fourier Discrète (DFT) définie par la formule suivante :

$$S[k] = \sum_{n=0}^{N-1} s[n] \cdot e^{-j \frac{2\pi kn}{N}} \quad (10)$$

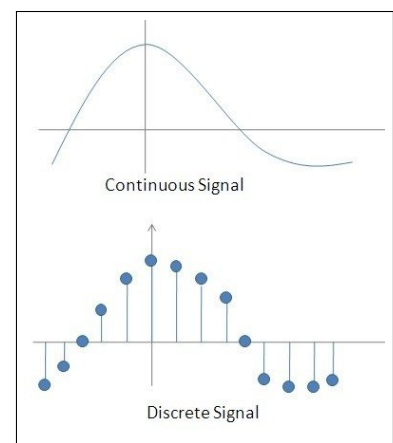


Figure 17 – *Différence entre un signal continu et un signal discret*

où :

- k représente l'indice de fréquence discret
- $S[k]$ est le spectre de fréquence discret du signal
- $s[n]$ est le signal d'entrée discret défini dans le domaine temporel
- N est le nombre total d'échantillons.

Calculer la DFT de chaque point pour chaque bin de fréquence a une complexité algorithmique de $O(N^2)$ ce qui est énorme. La Transformée de Fourier Rapide (FFT) est un algorithme qui calcule les mêmes résultats mais réduit la complexité à $O(N \log_2 N)$. Enfin, la Transformée de Fourier à Court Terme (STFT) consiste à appliquer la FFT aux trames successives du signal audio. Dans notre cas, l'audio est divisé en trames de 20ms, et une FFT est appliquée à chacune. Ce processus cartographie le signal en bins de fréquence discrets pour chaque intervalle de temps, nous permettant d'observer comment le spectre de fréquence évolue au fil du temps, résultant en un spectrogramme.

Le signal source est découpé en petites trames de 20ms afin que nous puissions supposer que le signal est stationnaire. Il est très important de supposer que le signal est stationnaire pour chaque trame car un filtre de Wiener nécessite des propriétés statistiques constantes pour distinguer efficacement le signal du bruit. La Figure 18 montre ce qu'est un filtre de Wiener.

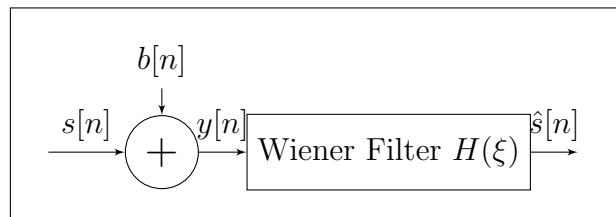


Figure 18 – *Diagramme bloc du système de filtre de Wiener.*

où :

- $s[n]$: est le signal utile
- $b[n]$: est le signal de bruit introduit dans le système
- Nœud de sommation (+) : est le processus où le signal propre et le bruit sont combinés pour former le signal observé
- $y[n]$: est le signal observé bruité
- Filtre de Wiener $H(\xi)$: est le filtre linéaire optimal qui traite le signal bruité dans le domaine fréquentiel
- $\hat{s}[n]$: est le signal propre estimé produit après le processus de réduction du bruit

L'avantage d'un filtre de Wiener est qu'il est adaptatif. Le filtre est souvent appelé gain et calculé avec la formule suivante pour chaque bin de fréquence de chaque trame.

$$G(f) = \frac{S_{xx}(f)}{S_{xx}(f) + S_{nn}(f)} \quad (11)$$

où :

- $G(f)$: Gain du filtre de Wiener.
- $S_{xx}(f)$: La densité spectrale de puissance du signal source
- $S_{nn}(f)$: La densité spectrale de puissance du bruit

Après l'implémentation d'un filtre de Wiener, nous n'avons pas de fichier audio. Rappelons que nous sommes toujours dans le domaine temps-fréquence. Pour avoir un fichier audio que nous pouvons écouter, nous devons juste appliquer la Transformée de Fourier à Court Terme Inverse (ISTFT).

Lorsque nous avons écouté le nouvel audio, le bruit du climatiseur était vraiment réduit. Et en regardant son spectre de fréquence, nous étions très satisfaits. En effet, pour la qualité de la réduction du bruit, le spectre était agréablement préservé, surtout les fréquences supérieures comme nous pouvons le voir sur la Figure 19. Dorénavant, notre implémentation Python d'un filtre de Wiener est indispensable avant de faire toute ingénierie de caractéristiques pour notre modèle d'IA.

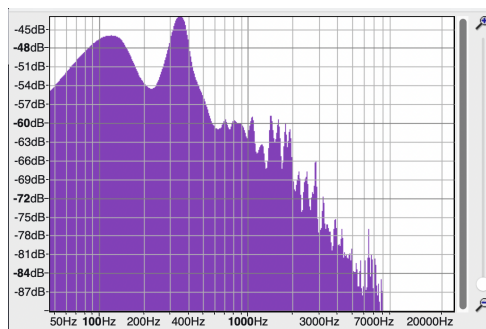


Figure 19 – *Spectre de fréquence de l'audio avec un filtre de Wiener appliqué*

5 Modèle de prédiction de concentration en polluants - *Pythie*

5.1 Méthodologie

Intuitivement, plus il y a de voitures, plus la pollution est élevée. De même, plus le trafic routier est important, plus le niveau sonore augmente. Partant de cette intuition, nous avons exploré la possibilité que le son ambiant d'une zone soit directement corrélé à sa pollution. Dans cette partie, nous souhaitons tester cette hypothèse. Nous allons donc dans un premier temps faire du *feature engineering*, nécessaire pour entraîner un modèle d'IA. Puis nous testerons différentes structures d'IA pour avoir un modèle optimal.

Nous allons entraîner des modèles de machine learning qui vont calculer la concentration en polluants pour chaque fenêtre de 20ms et qui a la fin, fera une moyenne de toutes les valeurs obtenues pour avoir un résultat global de pollution pour un fichier audio donné.

5.2 Feature engineering

Intuitivement, plus il y a de voitures, plus la pollution est élevée. De même, plus le trafic est dense, plus le niveau sonore augmente. Partant de cette intuition, nous avons exploré la possibilité que le son ambiant d'une zone soit directement corrélé à sa pollution. Dans cette partie, nous souhaitons tester cette hypothèse.

La STFT est une excellente représentation d'un signal. Cependant, elle traite chaque fréquence de manière linéaire, ce que le système auditif humain ne fait pas et ce qui n'est pas idéal pour capturer la complexité d'un environnement urbain. Dans une ville, les sons les plus utiles, comme le vrombissement d'un moteur ou le bruit de pas, sont concentrés dans des plages de fréquences spécifiques. En utilisant un spectrogramme Mel, nous pouvons concentrer la résolution sur ces zones d'intérêt. C'est une approche bien plus efficace pour l'entraînement d'une IA car elle réduit la dimensionnalité tout en préservant l'information pertinente.

Le diagramme suivant détaille les étapes nécessaires pour convertir notre audio en un spectrogramme Mel (Figure 20).

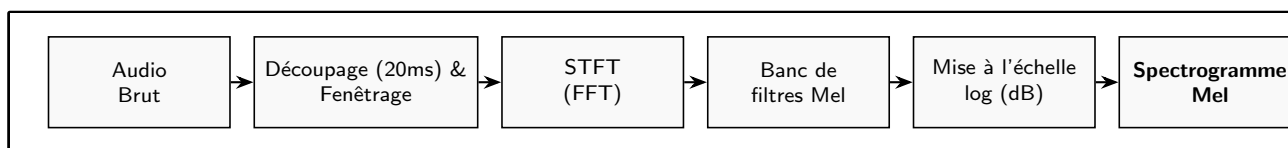


Figure 20 – *Pipeline de conversion en spectrogramme Mel*

Nous avons découpé notre audio en fenêtres (frames) de 20 ms. Grâce au spectrogramme Mel, chacune d'elles est décrite par un vecteur de 128 valeurs. Ces valeurs caractérisent le son sur cette période de 20 ms. Nous disposons désormais de nos features.

Concernant les targets, nous avons précisé précédemment que toutes les données des polluants sont relevées chaque seconde. Pour correspondre à nos fenêtres de 20 ms, chaque donnée est multipliée 50 fois afin d'aligner les échelles temporelles. Pour le CO_2 qui nous donne des données toutes les 5 secondes, nous multiplions les données par 250 pour toujours correspondre

à nos fenêtres de 20 ms

En résumé, voici les données dont nous disposons pour entraîner notre modèle d'IA :

- **Features** Un vecteur de 128 valeurs pour chaque fenêtre de 20 ms.
- **Targets** Les valeurs de $PM_{2.5}$, PM_{10} , CO , CO_2 , NH_3 et NO_2 pour chaque fenêtre de 20 ms.

5.3 Choix de structure de modèle

Nos features et nos targets sont des données numériques. Nous essayons, à partir de données numériques de prédire des données numériques également. Ce type de problème en machine learning s'appelle une regression en apprentissage supervisé. Les structures de modèles les plus utilisées pour ce type de problème sont les Random Forest, Gradient Boosting et les réseaux de neurones MLP (Multi-Layer Perceptron).

Nous avons donc entraîné ces trois types de modèles sur le même jeu de données et en leur faisant prédire la concentration en polluants moyenne pour un même fichier audio. Les résultats sont montrés sur la Figure 21.

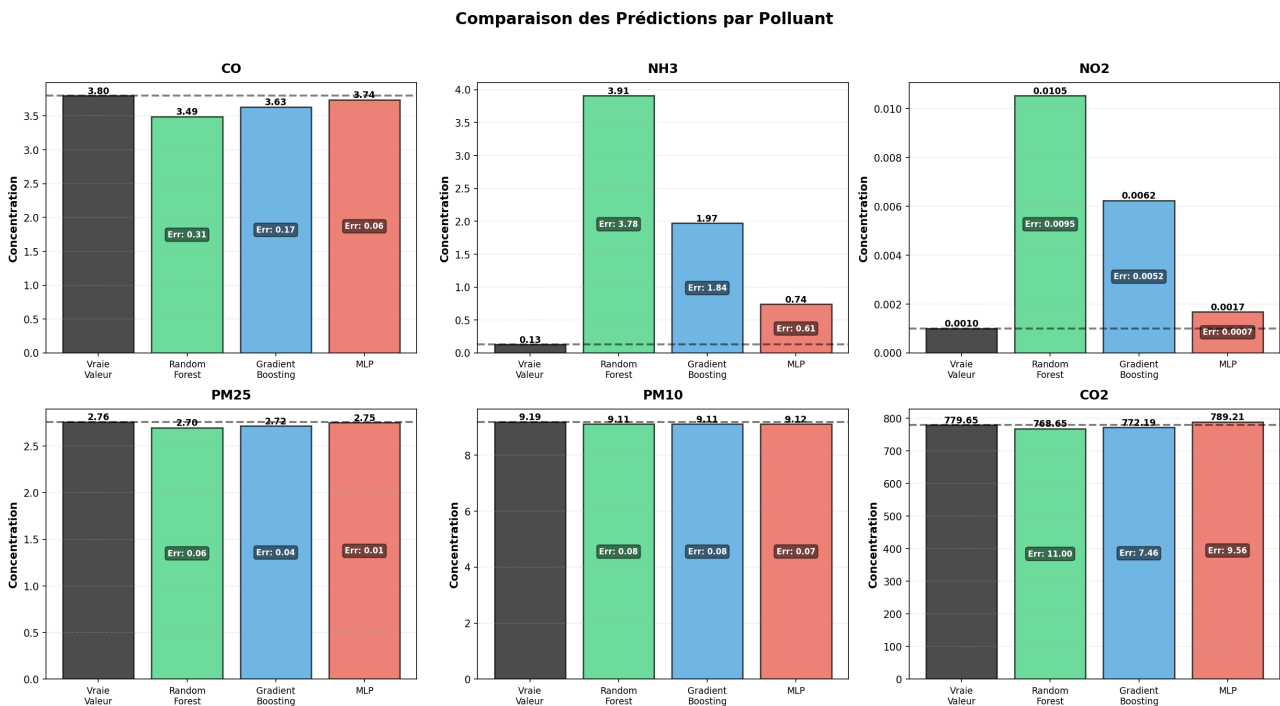


Figure 21 – Comparaisons du Random Forest, Gradient Boosting et du MLP

Les résultats sont très intéressants, sauf pour l'ammoniac et le dioxyde d'azote. Cela peut être expliqué par le fait que les voitures ne sont pas responsables directement des concentrations en ammoniac et en dioxyde d'azote. Les modèles ayant appris en écoutant un audio, majoritairement de trafic routier, ils ne peuvent pas bien faire de lien entre les features et ces targets NH_3 et NO_2 .

Pour évaluer chaque modèle, nous avons calculé leur erreur moyenne sur tous les polluants sauf le NH_3 et NO_2 avec cette formule :

$$\text{Erreur} = \frac{1}{N} \sum_{i=1}^N \frac{|C_{\text{réelle},i} - C_{\text{prédite},i}|}{C_{\text{réelle},i}}$$

Nous avons alors les erreurs suivantes :

- MLP : 0.97%
- Gradient Boosting : 1.96%
- Random Forest : 3.18%

Le meilleur modèle est le MLP. Nous avons alors notre Pythie qui est fonctionnelle.

6 Résultats et site internet

6.1 Bilan de Pandore, Hermès et Pythie

Nous avons à présent une boîte à capteurs, Pandore, qui fonctionne bien et qui peut être utilisée dans de nombreux cas de figure. Hermès vole, sauf lorsqu'il y a des crash puisque cela nous prend du temps à le réparer. Enfin Pythie, notre oracle est très précise lorsqu'il s'agit de prédire les données de pollution en fonction d'un audio.

6.2 Notre site internet

Nous pensons qu'avoir des résultats est important, mais l'accès à ces résultats l'est encore plus. C'est pourquoi nous avons créé un site internet qui affiche les missions que nous avons effectuées jusqu'à présent. Ces missions comportent une cartographie de notre itinéraire et la concentration en polluants sont classés en 4 catégories :

- Faible
- Modéré
- Élevé
- Très élevé

Nous vous invitons à le visiter à l'adresse suivante : https://vincenzosbrain.github.io/Website_projet_hermes/

Références

- [1] S. F. Boll, “Suppression of acoustic noise in speech using spectral subtraction,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-27, pp. 113–120, April 1979.
- [2] C. E. Shannon, “Communication in the presence of noise,” *Proceedings of the IRE*, vol. 37, no. 1, pp. 10–21, 1949.
- [3] S. W. Smith, *The Scientist and Engineer’s Guide to Digital Signal Processing*. San Diego, California : California Technical Publishing, 1997.
- [4] S. Davis and P. Mermelstein, “Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 4, pp. 357–366, 1980.
- [5] C. Gasquet and P. Witomski, *Transformation de Fourier*. Paris : Dunod, 2000. Introduction à l’analyse de Fourier et à ses applications.

Annexes

A Le système dBFS et le RMS

Le système dBFS

Le décibel pleine échelle (dBFS, pour *Decibels relative to Full Scale*) est une unité de mesure des niveaux d'amplitude dans les systèmes numériques. Il définit le niveau du signal par rapport au maximum possible, où 0 dBFS représente le plafond numérique absolu. Comme un signal ne peut pas dépasser la résolution binaire disponible, toutes les valeurs sont exprimées par des nombres négatifs.

Le RMS

La valeur efficace ou RMS (*Root Mean Square*) est la mesure mathématique de la puissance moyenne d'un signal. Elle est calculée en élevant les valeurs d'amplitude au carré, en en faisant la moyenne, puis en extrayant la racine carrée du résultat. Contrairement aux niveaux de crête, le RMS représente l'énergie effective du son, ce qui en fait la mesure la plus précise pour la perception du volume sonore.