YNOV informatique Toulouse

Titre RNCP 39586

Ingénieur en science des données spécialisé en infrastructure data ou en apprentissage automatique.

Rapport bloc 2

ANALYSER, ORGANISER ET VALORISER DES DONNEES

Exploration de données et analyse de traces électromagnétiques destinées à tester la vulnérabilité d'un processeur face aux attaques SCA (Side Channel attack)

Rédigé par	***Responsabilité***
Maxence colvray	Apprenti datascientist

Table des matières

1 Description synthétique du projet (C2.1.1) 4
1.1 Qu'est-ce qu'une attaque par canaux auxiliaires (SCA) ? 4
1.2 L'apport de l'intelligence artificielle dans les attaques par canaux auxiliaires (SCA) 4
2 Analyse du besoin (C2.1.1) 5
2.1 Enjeux et problématiques 5
2.2 Contexte 6
2.3 Environnement 6
2.4 Contraintes 7
3 Plan d'analyse (C2.1.2) 7
3.1 Les données à exploiter 7
3.2 Axes d'analyse exploratoire 7
3.2.1 Analyse temporelle (temps relatif) *7*
3.2.2 Analyse fréquentielle *8*
3.2.3 Analyse statistique *8*
3.2.4 Analyse comparative entre clés *8*
3.3 Extraction de caractéristiques (features) 8
3.4 Métriques retenues 8
3.5 Réduction et alignement 8
3.6 Organisation et restitution des résultats 9
3.7 Livrables 9
4 Présentation des outils et techniques d'analyse (C2.1.3) 10
4.1 Registre MongoDB et requête MQL 10
4.2 Présentation synthétique des données avec MQL depuis Mongo Charts 11
4.3 Analyse exploratoire avec jupyter notebook 12
4.3.1 Analyse temporelle comparative *12*
4.3.2 Analyse fréquentielle comparative *14*
5 Méthodologie de tests statistiques (C2.1.4) 15
5.1 Métriques statistiques (moyenne, écart-type, variance) 15
5.2 Distribution des amplitudes par coefficient 17
5.3 Analyse PCA 18
5.4 Calcul de distance DTW entre signaux (Dynamic Time Warping) 19
5.5 Synthèse des tests d'hypothèses statistiques 20
6 Visualisation des résultats (C2.2.1) 20
6.1 Plotly Dash 21
6.2 Présentation du tableau de bord interactifs 21
6.3 Prise en compte des personnes malvoyantes 24
7 Recommandations (C2.2.2) 25
8 Guide utilisateur (C2.3.1) 26

9 Documentation technique, Dashboard d'analyse des traces EM (C2.3.2) 28
10 Conclusion 30

Terminologie

SCA	Side Channel Attack ou attaque par canaux auxiliaire
EM	Electromagnétique
Radiofréquence	La radiofréquence est la gamme d'ondes électromagnétiques (3khz à 300Ghz) utilisée pour les transmissions sans fil
NTRU	(Nth degree Truncated polynomial Ring Units) algorithme post-quantique basé sur des polynômes à petits coefficients
ANSSI	Agence nationale de la sécurité des systèmes d'information
OLAP	Online Analytical Processing
PCA	Analyse en Composantes Principales
DTW	(Dynamic Time Warping) La distance DTW compare deux séries de données qui n'avancent pas au même rythme
FFT	(Transformée de Fourier Rapide) La FFT décompose un signal en ses différentes fréquences

1 Description synthétique du projet (C2.1.1)

1.1 Qu'est-ce qu'une attaque par canaux auxiliaires (SCA) ?

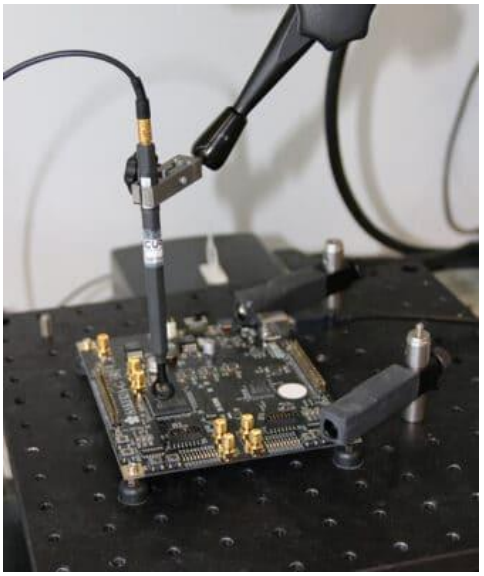
Une attaque par canaux auxiliaires (SCA, pour Side Channel Attack) est une méthode de piratage pour laquelle on ne rentre pas dans le système, mais où l'on exploite les fuites physiques volontaire ou involontaire (comme les ondes électromagnétiques qui sortent d'un système) pour en déduire des informations confidentielles. Ce projet concerne uniquement les SCA appliquées fuites d'ondes électromagnétiques, appelé aussi SCA EM. On retrouve 2 cas d'usage principaux :

Classification d'un trafic chiffré :

Analyser un signal radio chiffré, provenant par exemple d'une borne wifi, d'un satellite, d'une radio, pour déduire des informations sur l'activité en cours comme classifier certaines caractéristiques du signal sans avoir besoin de déchiffrer le contenu (ex : provenance, matériel utilisé, type de contenus etc.).

Cryptanalyse :

Déchiffrage de clé de chiffrement (ex : AES, RSA, NTRU) en classifiant des traces EM généré par le processeur pendant les opérations de déchiffrement.



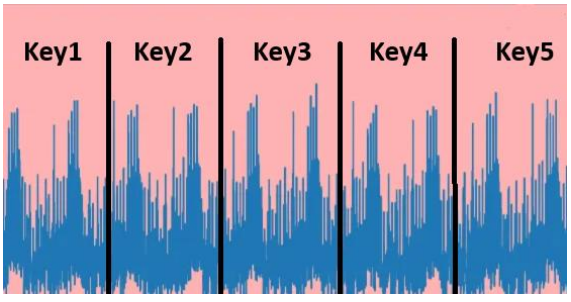
Plusieurs contre-mesures existent et ont besoin d'être testées pour palier à ce type d'attaques, blindage électromagnétique, capteurs de détection d'intrusion, filtrage et découplage ou contre-mesures logicielles comme le masquage ou le binding, mais avant cela il faut déjà tester la vulnérabilité du système en question aux attaques.

Pour mieux comprendre le principe des attaques par canaux auxiliaires assistées par intelligence artificielle, voici une vidéo explicative en français :
https://www.youtube.com/watch?v=fmnXhsZk61Y

Par exemple, à chaque fois que le processeur devra décrypter une requête entrante il utilisera une clé de chiffrement pour déchiffrer son contenu. En analysant l'onde générée par le processeur lors de l'utilisation de chaque élément de la clé on peut déterminer la valeur de l'élément de la clé (appelé aussi coefficient de la clé).

1.2 L'apport de l'intelligence artificielle dans les attaques par canaux auxiliaires (SCA)

Aujourd'hui il est possible de mener des attaques par canaux auxiliaires pour extraire des clés cryptographiques en analysant les traces électromagnétiques, et en appliquant des méthodes de classification basées sur le Deep Learning pour distinguer les valeurs de clé.



L'IA, associée à la collecte précise des traces électromagnétiques, transforme les attaques par canaux auxiliaires en rendant possible l'extraction rapide et précise d'informations complexes, tout en étant plus résiliente aux problèmes de synchronisation que les méthodes traditionnelles.

Ces approches permettent d'automatiser l'analyse de signaux complexes et de détecter des patterns subtils difficilement identifiables par des humains. En reproduisant ce type d'attaques en environnement contrôlé, l'intelligence artificielle devient un outil précieux pour évaluer la robustesse des implémentations matérielles face aux menaces SCA. Cela contribue à identifier les vulnérabilités dès la phase de conception et à mettre en place des contre-mesures efficaces pour sécuriser les processeurs et systèmes embarqués.

2 Analyse du besoin (C2.1.1)

2.1 Enjeux et problématiques

Les attaques par canaux auxiliaires (Side Channel Attacks ou encore SCA) représentent une menace croissante pour la sécurité des systèmes embarqués et des processeurs cryptographiques. Elles exploitent les fuites physiques, telles que les émissions électromagnétiques, pour extraire des informations sensibles comme des clés privées.

Les attaques SCA nécessitent des moyens matériels et logiciels importants pour être mises en œuvre, mais elles ciblent principalement des systèmes à forte valeur ajoutée, où les informations protégées peuvent justifient l'investissement requis. Dans ce contexte, l'intelligence artificielle, et notamment le Deep Learning, offre de nouvelles capacités d'analyse.

Ce projet vise à offrir aux ingénieurs en électronique et radiofréquence un outil pour visualiser, analyser et sélectionner des traces électromagnétiques, afin de tester, plus tard, la résistance de leurs processeurs face aux attaques par canaux auxiliaires (SCA).

L'objectif de ce rapport n'est donc pas de mener directement des attaques SCA basées sur le deep learning, mais de **se concentrer pour l'instant sur l'analyse, l'organisation et la valorisation des données entre la phase de collecte et la phase d'attaque**.

Concrètement les utilisateurs finaux des travaux présentés dans ce rapport seront les ingénieurs électroniques et radiofréquences.

Les enjeux de ce projet se déclinent en trois axes :

- **Vérifier l'intégrité** et la qualité des traces électromagnétiques afin de s'assurer que les mesures se sont correctement déroulées et que les données collectées sont exploitables.

- **Offrir une visualisation claire** permettant de comparer les signaux et de sélectionner les traces les plus pertinentes pour entraîner un modèle ou conduire des attaques.

- **Extraire des informations** et caractéristiques issues des signaux qu'elles soient fréquentielles (spectre, bande passante), temporelles (temps de montée, durée d'impulsion) ou statistiques (moyenne, variance, entropie) afin que les ingénieurs puissent travailler sur l'explicabilité des attaques par canaux auxiliaires.

Problématique technique : Comment s'assurer de la fiabilité et de l'extraction d'informations provenant des traces électromagnétiques représentant des millions de points, alors que les ingénieurs en électronique ou en radiofréquence, bien que spécialistes du signal, ne sont pas toujours familiers avec la manipulation et le traitement de données massives nécessitant des compétences avancées en data analyse ?

Problématique opérationnelle : Comment faciliter le travail des ingénieurs en rendant l'interprétation des traces plus rapide et intuitive, tout en permettant la reproduction aisée des attaques et en réduisant le temps nécessaire pour passer de la collecte des données à leur exploitation ?

2.2 Contexte

Le **dataset SCANTRU** a été publié par l'ANSSI dans le but de mettre à disposition un corpus représentatif de traces électromagnétiques collectées lors d'exécutions répétées de l'algorithme cryptographique NTRU sur un processeur dédié.

Ce jeu de données constitue une ressource quasiment unique pour l'étude des attaques par canaux auxiliaires (Side Channel Attacks, SCA), en offrant un ensemble de mesures réelles issues d'un environnement matériel en laboratoire.

NTRU (Nth-degree Truncated Polynomial Ring Units) est un algorithme de chiffrement à clé publique basé sur la cryptographie post-quantique. Il repose sur l'arithmétique des polynômes, ce qui le rend à la fois rapide et résistant aux attaques connues, y compris celles menées par des ordinateurs quantiques. A savoir qu'il n'y pas besoin d'utiliser d'ordinateur quantique pour utiliser des clés chiffrées avec l'algorithme NTRU.

Une clé NTRU est en fait un vecteur de coefficients entiers (souvent −1, 0, ou 2047) de longueur fixe. Dans la pratique, ça ressemble à une suite de nombres comme [1, 0, -1, 2047, ...] et ces éléments définissent la clé privée utilisée pour chiffrer/déchiffrer.

Les données sont organisées sous la forme de séries temporelles multidimensionnelles, segmentées par opérations cryptographiques, ce qui permet d'isoler et d'analyser chaque étape du calcul. Cette structure facilite l'examen du signal qui pourrait contenir des informations corrélées à des elements de la clé privée.

Dans un cadre industriel, l'exploitation de telles données présente plusieurs intérêts majeurs :

- Tester la résilience des implémentations matérielles et logicielles face à des attaques SCA, en mesurant leur capacité à masquer ou brouiller les fuites d'informations.
- Alimenter la recherche sur les contre-mesures, en comparant l'efficacité de différentes stratégies de protection et en identifiant leurs limites dans des scénarios réalistes.
- Former et sensibiliser les équipes de sécurité et de conception, en leur fournissant des exemples concrets de fuites exploitables et en illustrant les méthodes d'analyse qui peuvent être utilisées par un attaquant.

2.3 Environnement

Le dataset SCANTRU est fourni sous la forme de fichiers .H5 contenant les traces électromagnétiques et les informations associées aux exécutions de l'algorithme NTRU. Chaque trace est représentée par un tenseur de dimensions (100, 507, 950), 100 exécutions pour une même clé, 507 points de mesure par exécution (nombre d'éléments dans la clé et 950 échantillons (valeurs de l'onde EM enregistrée) pour chaque élément de la clé.

Ces données ont été collectées dans un environnement de laboratoire contrôlé, à l'aide d'équipements de mesure de haute précision, tels que des sondes électromagnétiques spécialisées et des oscilloscopes haut de gamme, capables d'acquérir des signaux à très haute fréquence. Cette configuration garantit la qualité et la fiabilité des mesures

Ces dimensions traduisent la richesse temporelle et structurelle des données, tout en offrant un format compatible avec les traitements analytiques ou l'apprentissage automatique. L'environnement de travail retenu repose sur Python, avec des bibliothèques spécialisées pour la manipulation de données et des outils de visualisation. Les analyses seront effectuées dans un Jupyter Notebook afin de combiner code, graphiques et interprétation dans un même document.

2.4 Contraintes

L'exploitation du jeu de données SCANTRU présente plusieurs contraintes techniques, réglementaires et organisationnelles :

Sur le plan technique, chaque fichier de traces contient environ **48 millions de points de mesure pour un seul fichier** (100 exécutions d'une seule clé cryptographique), et plusieurs dizaines de fichiers sont disponibles. La taille massive de ces données impose d'organiser et traiter les données de manière fiable et organisée.

Sur le plan réglementaire, les données proviennent d'expérimentations encadrées par l'ANSSI et sont mises à disposition à des fins de recherche et de formation uniquement. Il est donc impératif de respecter une règle stricte de **non-exploitation malveillante**, en excluant toute utilisation visant à compromettre un système réel.

D'un point de vue budgétaire, ce projet est mené avec des **ressources limitées**, il s'agirait d'un début de projet dans un contexte d'entreprise, sans investissement dans des infrastructures couteuses, et reposant principalement sur des outils open source.

Enfin, une contrainte majeure réside dans **l'accessibilité des résultats**. Les analyses et visualisations produites devront être compréhensibles et exploitables non seulement par des experts en sécurité informatique, mais aussi **par des ingénieurs non spécialisés en informatique**, afin de faciliter la communication interdisciplinaire et l'adoption des résultats.

3 Plan d'analyse (C2.1.2)

3.1 Les données à exploiter

Le dataset SCANTRU, contient des séries temporelles multidimensionnelles issues de mesures expérimentales. Chaque trace est représentée par un tenseur de forme (100, 507, 950), correspondant respectivement au nombre d'exécutions, au nombre de segments par exécution et au nombre d'échantillons par segment. A cela s'ajoute une clé de chiffrement de 507 éléments qui a été utiliser pour les 100 exécutions d'un même fichier.

Les données exploitables se décomposent en :

- **Signaux bruts** : valeurs d'intensité électromagnétique en fonction du temps, permettant l'observation directe de l'onde.
- **Métadonnées associées** : informations sur les **éléments de clé** (**appelé aussi coefficient**) (0, -1 ou 2047) ainsi que son emplacement dans la clé de chiffrement.

3.2 Axes d'analyse exploratoire

Pour structurer l'exploration et la visualisation du dataset SCANTRU, quatre axes principaux :

3.2.1 Analyse temporelle (temps relatif)

- Observation des signaux sur les 950 échantillons de chaque segment, considérés comme un temps relatif.
- Comparaison entre différentes exécutions pour mettre en évidence des variations d'amplitude et de forme d'ondes (entre un même coefficient à des emplacements similaires et différents dans la clé).
- Identification potentielle visuelle de patterns récurrents ou d'anomalies.

3.2.2 Analyse fréquentielle

- Application de la Transformée de Fourier pour observer la répartition d'énergie dans les fréquences.
- Mise en évidence de signatures fréquentielles caractéristiques (ex : pics, harmoniques, bruit).

3.2.3 Analyse statistique

- Calcul de métriques simples : moyenne, écart-type, variance par segment.
- Analyse des distributions (comme histogrammes) pour mesurer la dispersion et la stabilité des signaux.

3.2.4 Analyse comparative entre clés

- Étude des différences entre moyennes de traces de clés distinctes.
- Calcul de distances (euclidienne, corrélation croisée) pour mesurer le pouvoir discriminant.

3.3 Extraction de caractéristiques (features)

L'extraction de caractéristiques vise à transformer les signaux bruts en indicateurs exploitables pour l'analyse et la visualisation :

- Caractéristiques temporelles simples : valeur maximale, valeur minimale, moyenne, énergie totale du signal.
- Caractéristiques statistiques : variance, coefficient de variation.
- Caractéristiques fréquentielles : fréquence dominante, puissance moyenne dans des bandes spécifiques.

Ces caractéristiques, volontairement limitées et interprétables, permettront de créer des tableaux de bord et visualisations accessibles à des profils non-informaticiens.

3.4 Métriques retenues

Les métriques clés sont :

- Analyse exploratoire : moyenne, écart-type, énergie par segment.
- Comparaison inter-exécutions : distance euclidienne, corrélation croisée.
- Évaluation de stabilité : variance intra-segment, coefficient de variation.

3.5 Réduction et alignement

- Réduction de dimension (PCA, LDA) pour condenser l'information discriminante et faciliter la visualisation.
- Alignement temporel (Dynamic Time Warping, corrélation croisée) pour corriger les désynchronisations potentielles entre exécutions et améliorer la comparabilité des signaux.

D'après plusieurs informations contenues dans des papiers de recherche, l'étape de réduction et d'alignement vise à améliorer la qualité et l'exploitabilité des signaux analysés. La réduction de dimension (PCA, LDA) permet de condenser l'information discriminante dans un espace réduit, ce qui facilite la visualisation et diminue la complexité des calculs. L'alignement temporel (Dynamic Time Warping, corrélation croisée) compense les désynchronisations entre exécutions, garantissant que les signaux comparés sont bien alignés dans leur déroulement opérationnel. Cette étape est particulièrement utile pour rendre les analyses robustes face au bruit et aux variations de capture.

3.6 Organisation et restitution des résultats

Les volets **Analyse exploratoire et Réduction & Alignement** constitueront la partie la plus flexible et itérative du projet.

Ces étapes incluant **l'analyse temporelle, fréquentielle, statistique et comparative** entre clés, ainsi que les techniques de réduction de dimension et d'alignement temporel seront approfondies dans un **notebook** interactif afin de permettre des ajustements dynamiques selon la pertinence des signaux observés.

En effet, il est difficile de figer un schéma d'analyse et de résultats strict à ce stade, car l'utilité de chaque analyse dépendra des patterns et anomalies réellement détectés dans les données.

À l'inverse, les phases **extraction de caractéristiques** et **calcul de métriques retenus** seront systématiquement effectuées et restituées dans un **tableau de bord** interactif, car elles fournissent des informations toujours exploitables à faible coût computationnel et sont facilement interprétables par les utilisateurs finaux.

A noter qu'il y aura également un tableau de bord pour la **présentation synthétique des données** incluant les notamment les dimensions du dataset, le nombre d'exécutions, de fichiers, les processeurs utilisés.

Cette formulation ouvre la voie à des traitements automatisés et à la mise en place de tableaux de bord interactifs. Les analyses pourront être déployées dans un environnement Python ou exportées vers un outil BI, permettant à un utilisateur non-informaticien de naviguer facilement dans les données et de visualiser des comparaisons entre exécutions, clés ou segments.

3.7 Livrables

Les livrables issus de ce projet seront conçus pour répondre à la fois aux besoins des ingénieurs analystes métiers et donc des utilisateurs non-informaticiens. Ils combineront des supports interactifs, des exports exploitables et une documentation, afin de garantir la réutilisation et la compréhension des résultats par un large public.

Notebooks : pour l'analyse exploratoire et la réduction/alignement, permettant aux analystes de tester, filtrer et affiner les observations.

Tableaux de bord interactifs pour les métriques calculées, l'extraction de caractéristiques et la présentation synthétique des données afin d'offrir une visualisation claire.

Exports de données : au format CSV pour fournir l'extraction de caractéristiques à d'autres équipes.

Documentation utilisateur et technique : pour expliquer le fonctionnement des outils, les méthodes utilisées et la manière d'interpréter les résultats.

4 Présentation des outils et techniques d'analyse (C2.1.3)

4.1 Registre MongoDB et requête MQL

Comme montré ci-dessous j'ai conçu une base de données documentaire orientée OLAP (Online Analytical Processing) avec Mongodb pour répertorier l'ensemble des fichiers de traces au format H5, accompagnés de métadonnées essentielles telles que la date de collecte, la clé, les dimensions des données, le nombre d'exécutions et le type de processeur utilisé.
Cette base joue un rôle central dans l'architecture du projet, elle sert à la fois de point d'entrée pour les analyses et d'outil d'archivage et de comparaison entre différents fichiers d'exécutions. L'administration de cette base serait confiée à un DBA (Database Administrator) garantissant sa fiabilité et sa sécurité, tandis que son exploitation au quotidien serait assurée par des data analysts ou data scientists pour leurs travaux d'exploration et d'ananlyses.

```
_id: ObjectId('689c661288f58ee87c22619f'
file_name : "scantru_trace_002.h5"
date_collecte : "2025-08-02"
cle_coefficients : "[2047,1,0,...,2047]"
▸ dimensions : Array (3)
nombre_executions : 100
processeur : "ARM Cortex-M3"
taille_fichier_mo : 253.1
```

La base documentaire sert désormais de point central d'accès aux fichiers SCANTRU et à leurs métadonnées. Grâce à son indexation, il est possible d'effectuer des requêtes en MQL (MongoDB Query Language) depuis MongoDB Compass ou via un client Python.

```
_id: ObjectId('689c661288f58ee87c2261a0'
file_name : "scantru_trace_003.h5"
date_collecte : "2025-08-03"
cle_coefficients : "[1,1,2047,...,1]"
▾ dimensions : Array (3)
    0: 100
    1: 507
    2: 950
nombre_executions : 100
processeur : "STM32F407VG"
taille_fichier_mo : 260.7
```

Voici 2 exemples de requêtes MQL illustrant comment sélectionner les données des fichiers SCANTRU en fonction de critères spécifiques en utilisant le client pymongo :

```python
from pymongo import MongoClient
from datetime import datetime
import h5py

Client = MongoClient("mongodb://localhost:27017/")
db = client["scantru_db"]
collection = db["scantru_index"]

# 2. Récupérer les documents après une certaine date
documents = collection.find({
    "date_collecte": {"$gt": datetime(2024, 1, 1)}
})

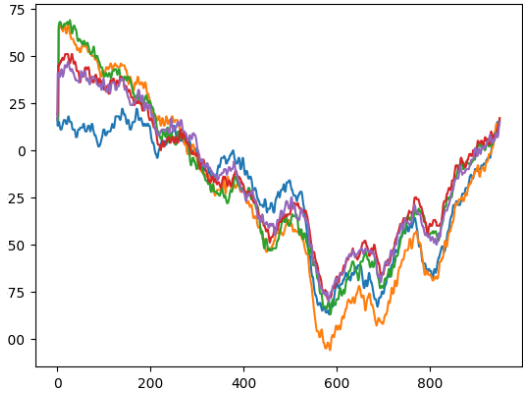
# 3. Charger les fichiers H5 et afficher quelques infos
for doc in documents:
    chemin_fichier = doc.get("/h5")
    # 4. Ouverture et lecture des traces
    with h5py.File(chemin_fichier, 'r') as h5f:
        print(f" {list(h5f.keys())}")
        traces = h5f["traces"][:]
```

Ci-contre, une requête MQL sélectionnant tous les documents de la collection *scantru_index* dont le champ *date_collecte* est postérieur au 1er janvier 2024, puis pour chaque document trouvé, on ouvre le fichier H5 associé pour accéder au groupe "*traces*".

Voici ci-dessous un autre exemple de requête MQL permettant de sélectionner les documents dont la clé commence par 1 , à partir de cette sélection, il est ensuite possible de superposer facilement les traces avec *python* et *matplotlib* afin de comparer visuellement les signaux associés à ces clés spécifiques.

```python
documents = collection.find({
    "cle": {"$regex": "^1"}
})

# 3. Charger les fichiers H5 et afficher quelques infos
for doc in documents:
    chemin_fichier = doc.get("/h5")
    # 4. Ouverture et lecture des traces
    with h5py.File(chemin_fichier, 'r') as h5f:
        print(f" {list(h5f.keys())}")
        traces = h5f["traces"][:1]
```



4.2 Présentation synthétique des données avec MQL depuis Mongo Charts

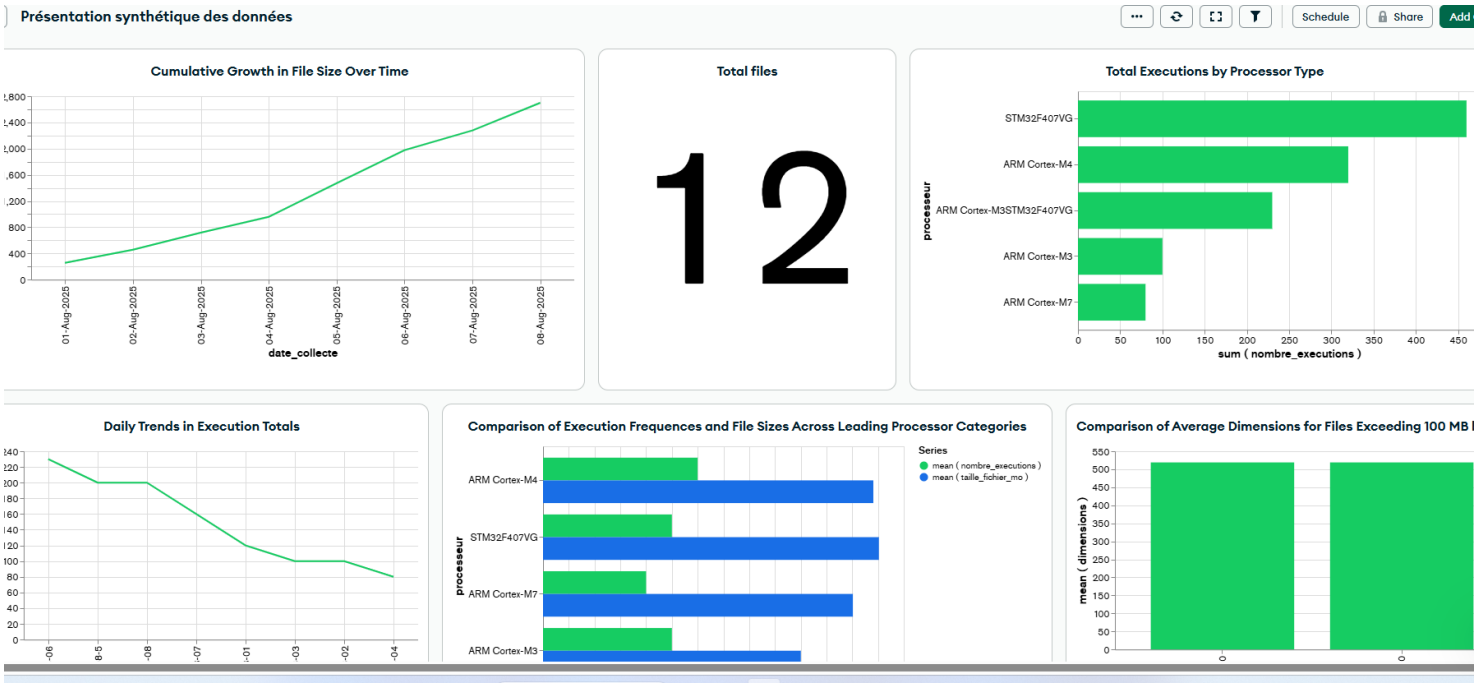
Un tableau de bord spécifique est mis en place pour la présentation synthétique des données, destiné aux ingénieurs métiers afin de leur offrir une vue d'ensemble claire et rapide sur le corpus SCANTRU.

Ce dashboard, conçu avec l'application MongoDB Charts, permet de visualiser via des requêtes MQL des informations clés telles que les dimensions du datasets, le nombre d'exécutions, le nombre total de fichiers et la répartition des processeurs utilisés. Il constitue ainsi un outil de Business Intelligence intégré directement à la base Mongo, facilitant l'exploration et la compréhension globale du jeu de données.

La requête MQL ci-dessous a été conçue pour alimenter le premier graphique du tableau de bord, en extrayant depuis la base MongoDB. Cette requête MQL convertit le champ *date_collecte* en format date, regroupe les documents par date, calcule la somme des tailles de fichiers pour chaque jour, puis prépare les résultats (x et y) pour générer un graphique.

```
[
  { $addFields: { date_collecte: { $toDate: "$date_collecte" } } },
  { $group: { _id: "$date_collecte", total_size_mo: { $sum: "$taille_fichier_mo" } } },
  { $project: { _id: 0, x: "$_id", y: "$total_size_mo" } },
  { $sort: { x: 1 } },
  { $limit: 5000 }
]
```

Ci-dessous sont présentés les six graphiques constituant le tableau de bord de synthèse du jeu de données SCANTRU. Disposés de haut en bas et de gauche à droite, ils permettent d'obtenir en un coup d'œil une vision globale et structurée du corpus. Le premier graphique illustre l'évolution du volume total du dataset dans le temps, suivi par un graphique indiquant le nombre total de fichiers collectés. Vient ensuite une visualisation présentant la répartition du nombre d'exécutions par processeur, puis un graphique montrant la variation quotidienne du nombre d'exécutions. Le cinquième graphique met en évidence la répartition conjointe du nombre d'exécutions et du volume de données par processeur, tandis que le dernier compare les dimensions moyennes des fichiers dépassant 100 Mo selon 2 plages de taille.
Ensemble, ces éléments offrent aux ingénieurs métiers et analystes une vision claire et interactive de la structure et de l'évolution du dataset.



4.3 Analyse exploratoire avec jupyter notebook

Ici, j'utilise Jupyter Notebook pour réaliser l'analyse exploratoire, car il permet de combiner code, graphiques et interprétation dans un même document. L'analyse se fera à partir des données brutes métier, tel que fournies par un ingénieur, sous la forme d'un fichier .h5 contenant les traces électromagnétiques préalablement découpées en segments.

Chaque segment correspond à la trace mesurée lors de l'utilisation d'un coefficient de la clé NTRU, ce qui permet d'étudier finement chaque étape du calcul. Le fichier utilisé ici contient 100 exécutions avec la même clé, et, à l'issue de l'analyse exploratoire.

On commence par exécuter une requête en PyMongo sur la base documentaire d'indexation afin de sélectionner les fichiers répondant à nos critères, puis on charge les fichiers .h5 correspondants pour en extraire et analyser les traces.

```
from pymongo import MongoClient
import h5py
import numpy as np

client=MongoClient("mongodb://localhost:27017/")
db=client["scantru_db"]
collection=db["scantru_index"]

resultats=collection.find({"date_collecte":{"$gt":"2025-08-01"}})

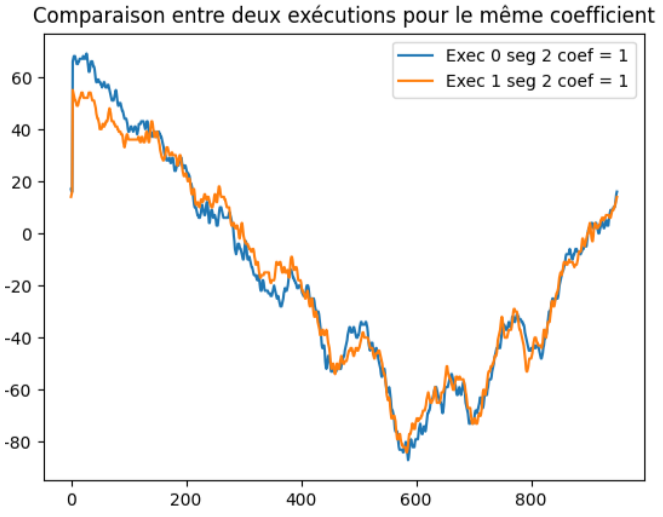
for doc in resultats:
    file_path=doc["file_name"]
    with h5py.File(file_path,'r') as f:
        traces=np.array(f["traces"])
        print(traces.shape)
```

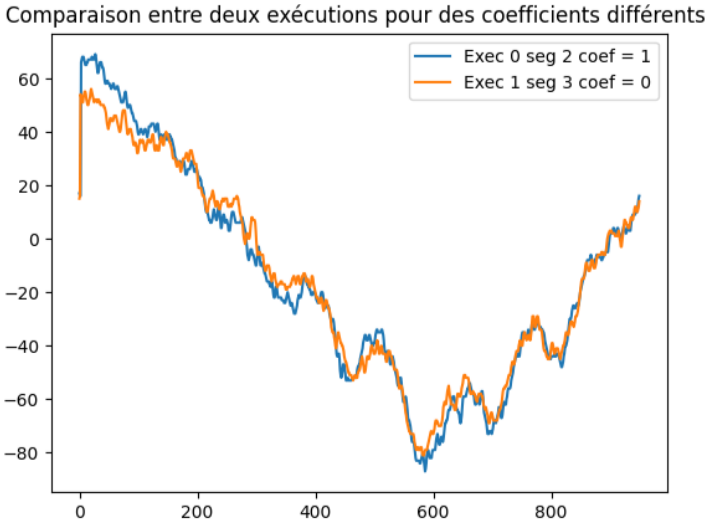
4.3.1 Analyse temporelle comparative

L'analyse temporelle consiste ici à observer visuellement la forme des signaux électromagnétiques sur un segment donné, en comparant plusieurs exécutions, pour des coefficients de clé similaires ou différents.

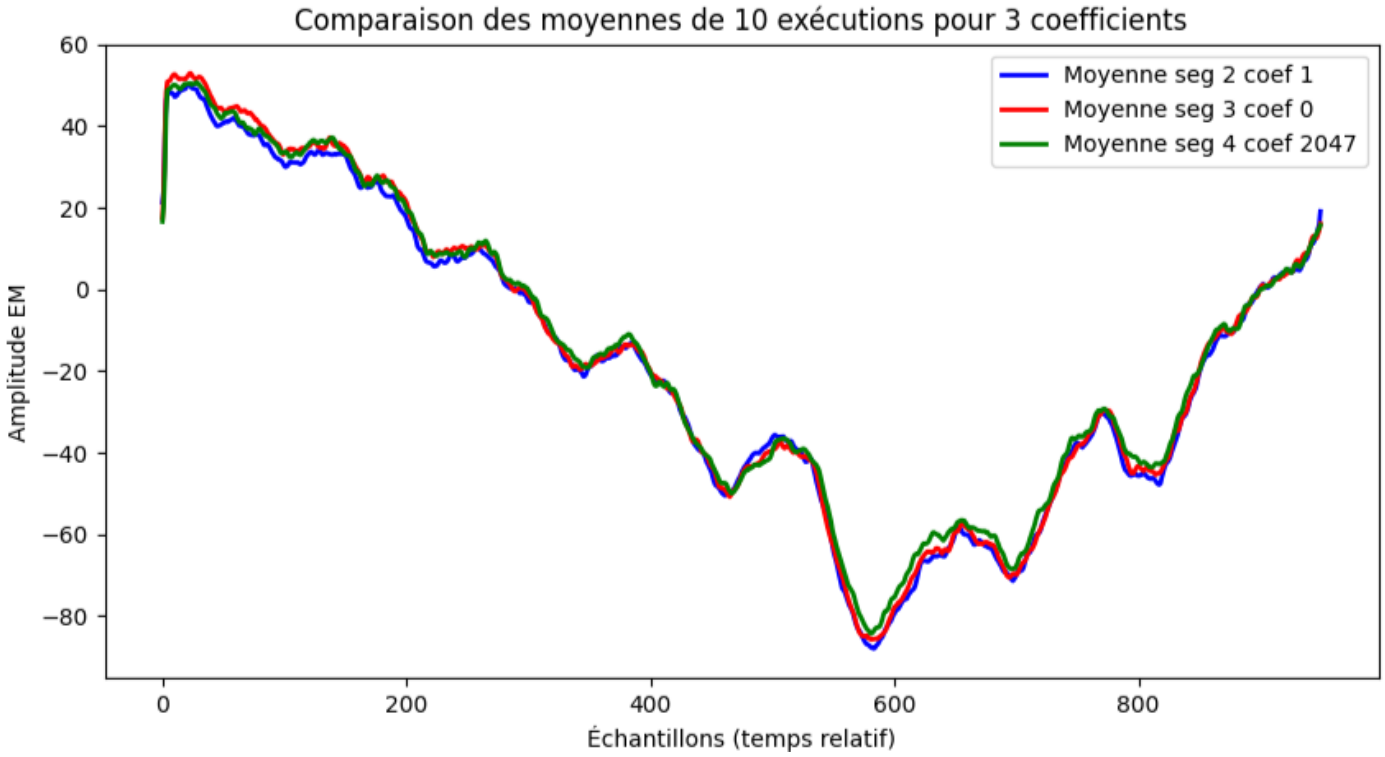
Cette approche permet de repérer rapidement des variations d'amplitude, des décalages ou des motifs récurrents qui pourraient indiquer des fuites d'informations exploitables.

Dans les graphiques, l'axe des abscisses représente le nombre d'échantillons (temps relatif au sein d'un segment, ici 950 points), tandis que l'axe des ordonnées correspond à l'amplitude du signal électromagnétique mesurée par les sondes lors de l'exécution en Dbm.





Ci-dessus, les différences visuelles entre traces, qu'elles proviennent d'un même coefficient ou de coefficients différents, restent faibles et ne permettent pas de conclusion directe. En revanche, la moyenne de 10 traces pour chacun des 3 coefficients, ci-dessous révèle de très légères variations perceptibles mais qui permettent difficilement de pouvoir tirer des conclusions car rien de probant visuellement pour l'instant.

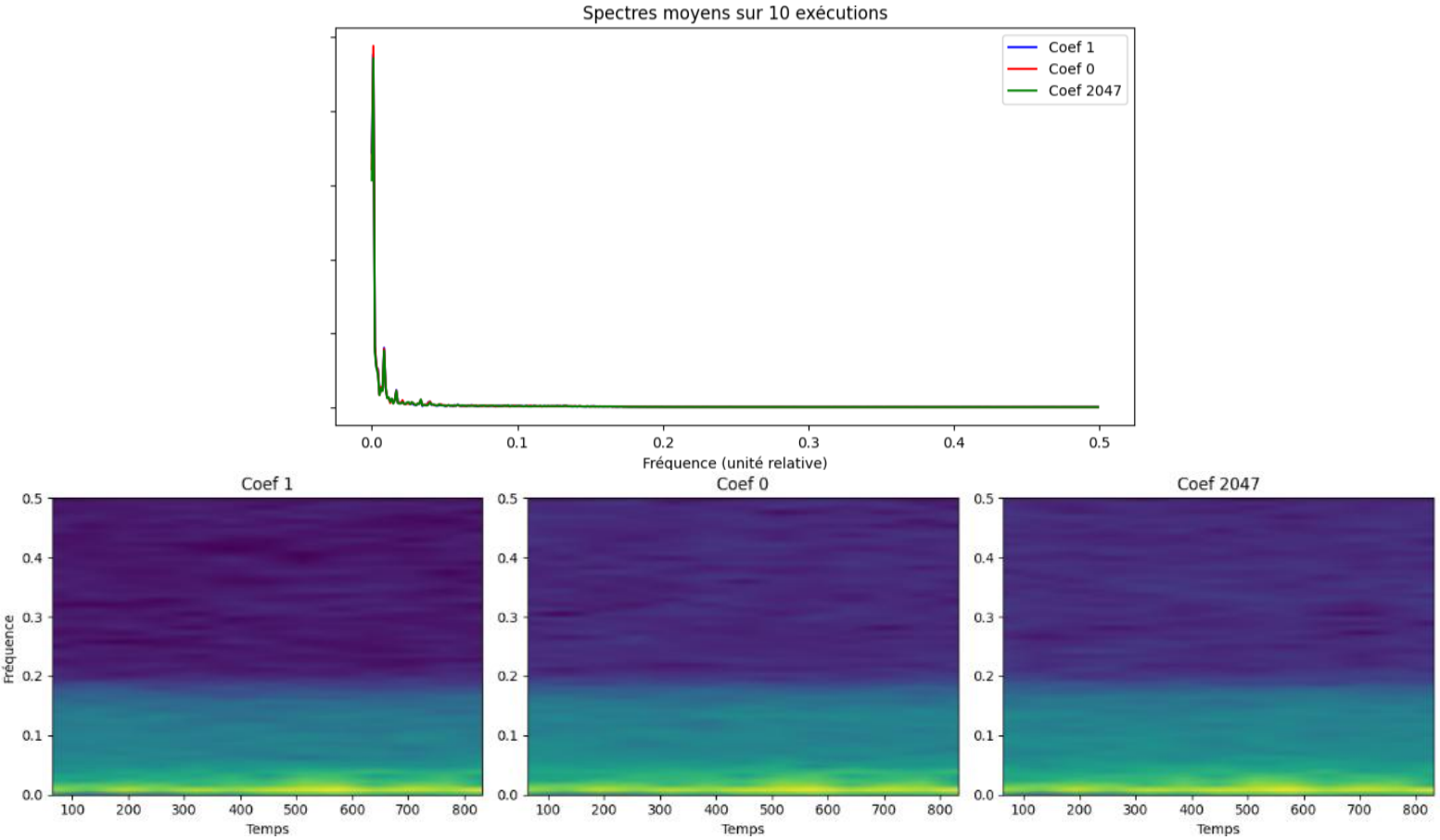


L'analyse temporelle permet tout de même de vérifier que les mesures électromagnétiques, obtenues à l'aide de sondes spécialisées et d'oscilloscopes haute précision, ont été correctement effectuées, sans anomalies évidentes ni pertes de données sur les segments observés.

4.3.2 Analyse fréquentielle comparative

On a vu précédemment que l'analyse en série temporelle brute ne permettait pas d'identifier clairement des différences significatives entre les signaux, qu'il s'agisse d'un même coefficient ou de coefficients différents. Nous passons donc maintenant à l'analyse fréquentielle, qui étudie la répartition de l'énergie du signal selon les fréquences. Techniquement, cette analyse est réalisée en Python en appliquant la Transformée de Fourier discrète (FFT) via *numpy.fft.fft* sur les signaux moyens. Cette opération décompose le signal en une somme de sinusoïdes, permettant d'identifier les composantes fréquentielles dominantes.

Le premier graphique présente ainsi la FFT moyenne calculée sur 10 exécutions pour chacun des trois coefficients, offrant une vue synthétique de l'amplitude des fréquences présentes.
Le second graphique est obtenu à l'aide de *matplotlib.pyplot.specgram*, qui calcule et affiche un spectrogramme, c'est-à-dire la FFT sur des fenêtres temporelles glissantes. Cela permet de visualiser l'évolution des fréquences au fil du temps relatif du segment et de repérer visuellement d'éventuelles signatures ou variations spécifiques à certaines plages temporelles.



On constate que l'analyse par FFT met en évidence de très légères variations dans la zone des basses fréquences entre les différents coefficients, bien que ces écarts restent faibles et peu exploitables directement à l'œil nu.

En revanche, les spectrogrammes offrent une vue plus complète de l'évolution fréquentielle dans le temps et permettent d'apercevoir, notamment dans les hautes fréquences, de subtiles différences de distribution d'énergie (zones bleu foncé).

Ces variations, bien que discrètes, confirment qu'il existe des nuances spectrales entre coefficients, mais leur exploitation nécessiterait probablement un traitement ou un filtrage plus poussé pour devenir pleinement discriminantes.

5 Méthodologie de tests statistiques (C2.1.4)

Dans le cadre de l'analyse statistique du dataset SCANTRU, j'ai réalisé une série de tests statistiques pour caractériser et comparer les traces électromagnétiques selon différents coefficients de la clé NTRU.

Notre méthodologie s'appuie sur quatre approches complémentaires : les métriques statistiques de base (moyenne, écart-type, variance) pour quantifier les caractéristiques centrales des signaux, l'analyse des distributions d'amplitudes via des histogrammes pour identifier les signatures distinctives, l'Analyse en Composantes Principales (PCA) pour réduire la dimensionnalité des 950 échantillons temporels et faciliter la visualisation des patterns discriminants, et le Dynamic Time Warping (DTW) pour mesurer la similarité temporelle entre signaux et quantifier les variations entre exécutions.

Chaque analyse est réalisée sur 50 exécutions pour 3 types de coefficients distincts (coefficients -1, 0, et 2047 de l'algorithme NTRU), soit un total de 150 comparaisons par méthode. Cette approche systématique permet d'évaluer la robustesse et la reproductibilité des patterns observés, tout en validant ou réfutant nos hypothèses statistiques et en identifiant les caractéristiques électromagnétiques spécifiques à chaque type de coefficient cryptographique.

Ces quatre méthodes implémentent directement les axes du plan d'analyse initial, à l'exception de l'analyse exploratoire déjà présentée dans le chapitre précédent, en couvrant l'analyse statistique (métriques et distributions), l'extraction de caractéristiques (PCA) et l'alignement temporel (DTW).

Hypothèse de variabilité inter-coefficients : Les différents types de coefficients NTRU (1, 0, 2047) génèrent des signatures électromagnétiques statistiquement distinctes en termes de moyenne, écart-type et variance, observables via la comparaison des métriques statistiques de base.

Hypothèse de normalité des distributions : Les amplitudes des signaux électromagnétiques suivent une distribution normale pour chaque type de coefficient, vérifiable par l'analyse des histogrammes comparatifs des amplitudes EM et permettant l'application de tests paramétriques.

Hypothèse de séparabilité dimensionnelle : La réduction de dimension par PCA des 950 échantillons temporels préserve l'information discriminante entre les coefficients, permettant une séparation efficace dans l'espace des composantes principales réduites.

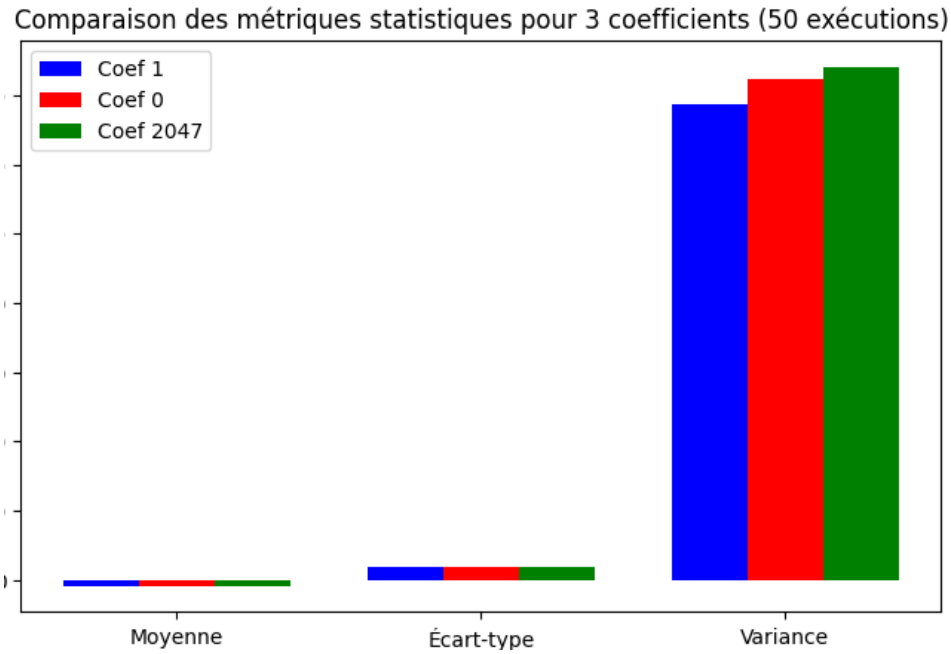
Hypothèse de stabilité temporelle : Les signaux présentent des patterns reproductibles entre les 50 exécutions d'un même coefficient, avec des variations limitées dues au bruit instrumental, quantifiables par l'analyse DTW des distances temporelles.

5.1 Métriques statistiques (moyenne, écart-type, variance)

Cette analyse calcule et compare les statistiques descriptives (moyenne, écart-type, variance) des traces électromagnétiques pour trois types de coefficients NTRU distincts. L'objectif technique est de quantifier numériquement les différences entre les signatures électromagnétiques en agrégeant, pour chaque coefficient, 50 exécutions × 950 échantillons temporels (soit 47 500 points de données) en trois indicateurs synthétiques, la moyenne (niveau d'amplitude électromagnétique moyen), l'écart-type (dispersion des amplitudes) et la variance (variabilité globale du signal). Le code extrait les données des segments 2, 3 et 4 correspondent aux coefficients 1, 0 et 2047, calcule les métriques et les présente dans un graphique en barres comparatif permettant d'identifier directement les différences statistiques entre les types de coefficients et de valider l'hypothèse de variabilité inter-coefficients.

Cette analyse calcule les statistiques descriptives (moyenne, écart-type, variance) des traces électromagnétiques pour trois coefficients NTRU distincts via les fonctions NumPy, *np.mean*, *np.std*, np.var. L'objectif est de quantifier les différences entre signatures électromagnétiques. Le graphique en barres comparatif valide l'hypothèse de variabilité inter-coefficients.

```python
# Fonction pour calculer les stats d'un coefficient
def stats_segment(segment_id):
    data = traces[exec_ids, segment_id, :]
    moyenne = np.mean(data)
    ecart_type = np.std(data)
    variance = np.var(data)
    return moyenne, ecart_type, variance
```

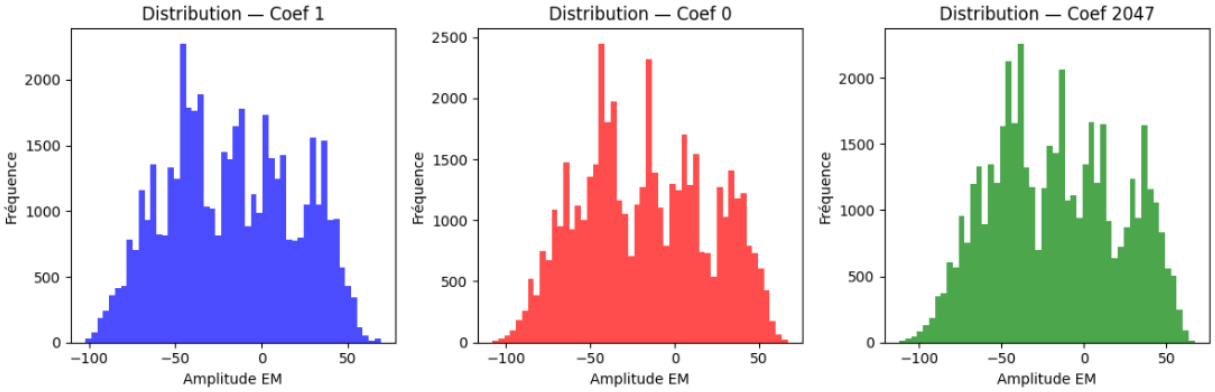



```
Variance Coefficient 1: 1372.3194
Variance Coefficient 0: 1445.2337
Variance Coefficient 2047: 1481.8991
```

Le graphique en barres comparatif révèle que les différences significatives entre les trois coefficients se manifestent uniquement au niveau des variances (l'unité de la variance est le carré de l'unité de base c'est-à-dire ici l'amplitude électromagnétique), tandis que les moyennes et écarts-types restent relativement similaires. Cette observation indique que les coefficients génèrent des signatures électromagnétiques de niveau d'amplitude comparable, mais avec des degrés de variabilité distincts, validant partiellement l'hypothèse de variabilité inter-coefficients.

5.2 Distribution des amplitudes par coefficient

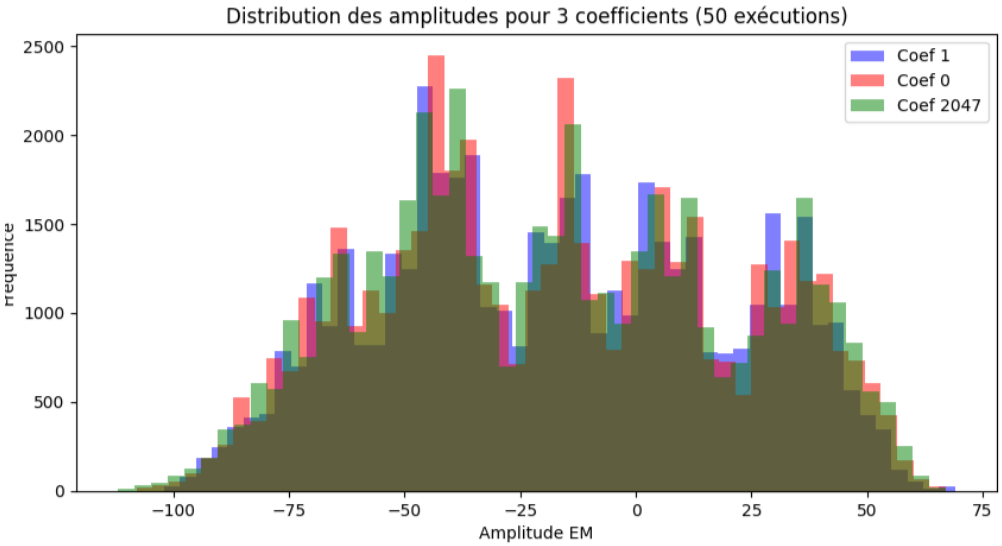
Suite au calcul des variances respectives des trois coefficients d'intérêt, nous procédons à présent à l'analyse de la distribution statistique des amplitudes pour chacun de ces coefficients. Cette étude vise à caractériser la répartition des valeurs d'amplitude observées sur l'ensemble des 50 exécutions pour chacun des 3 coefficients, permettant ainsi d'identifier d'éventuelles signatures en fonction des coefficients analysés. L'examen de ces distributions pourrait mettre en évidence des propriétés telles que des moments prépondérants, des asymétries ou des écarts à la normalité susceptibles de révéler des corrélations entre les patterns d'amplitude observés et les opérations cryptographiques sous-jacentes du processus de chiffrement NTRU pour chaque coefficient.



Les histogrammes présentés placent l'amplitude en abscisse et la fréquence d'occurrence en ordonnée. L'axe des abscisses représente les différentes valeurs d'amplitude observées dans les traces, tandis que l'axe des ordonnées indique le nombre de fois que chaque valeur d'amplitude apparaît sur l'ensemble des 50 exécutions et des points temporels analysés autrement dit, combien de fois on mesure une amplitude donnée dans toutes nos données pour chaque coefficient.

L'analyse des résultats révèle que les trois histogrammes présentent des profils de distribution subtilement distincts, confirmant que les coefficients 0, 1 et 2047 exhibent des comportements statistiques différenciés. Ces disparités dans les distributions d'amplitude à travers la position des pics principaux de la distribution qui témoignent de signatures spécifiques à chaque coefficient. Cette hétérogénéité de « pics » statistiques suggère que les opérations effectuées sur ces différents coefficients durant le processus de chiffrement NTRU induisent des consommations énergétiques caractéristiques, potentiellement exploitables dans le cadre d'une analyse par canaux auxiliaires.

La superposition des trois histogrammes ci-contre permet une comparaison directe des distributions. On observe clairement les zones où les 4 pics diffèrent en amplitude entre les coefficients. Certaines valeurs d'amplitude présentent des fréquences d'occurrence nettement distinctes selon le coefficient considéré.



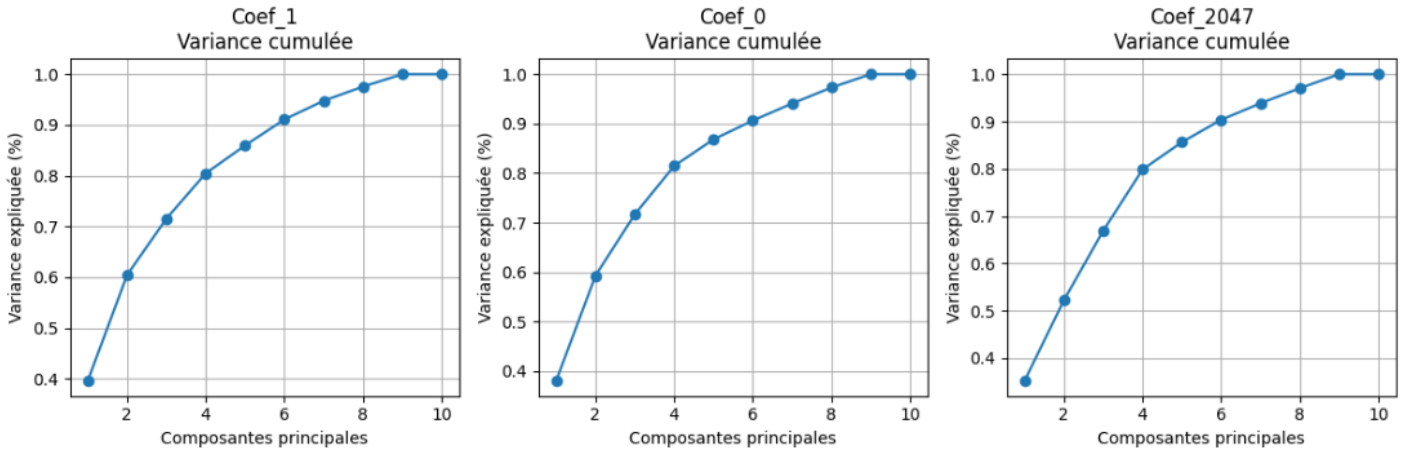
5.3 Analyse PCA

Dans cette étape, nous appliquons une Analyse en Composantes Principales (PCA) pour réduire la complexité de nos données de traces tout en gardant l'information importante. La PCA permet d'identifier les patterns principaux dans les signaux de consommation électrique pour chaque coefficient.

Concrètement, nous partons de 50 exécutions avec chacune 950 points de mesure temporels. C'est beaucoup trop de dimensions pour une analyse efficace. La première étape standardise les données, on centre autour de zéro et on normalise les échelles. Cela évite qu'une partie du signal avec de fortes amplitudes masque les variations subtiles ailleurs dans la trace.

La PCA trouve ensuite les directions où les données varient le plus. Au lieu d'analyser 950 points temporels, on se retrouve avec 10 composantes principales qui capturent l'essentiel de l'information. Chaque composante est une combinaison des points originaux qui représente un pattern caractéristique dans les traces.

Les graphiques montrent le pourcentage de variance expliquée cumulée pour chaque coefficient. Si une courbe monte rapidement (par exemple 90% de variance avec seulement 3 composantes), cela signifie que les traces de ce coefficient sont très redondantes et suivent des patterns simples. Si la courbe monte lentement, les traces sont plus complexes.



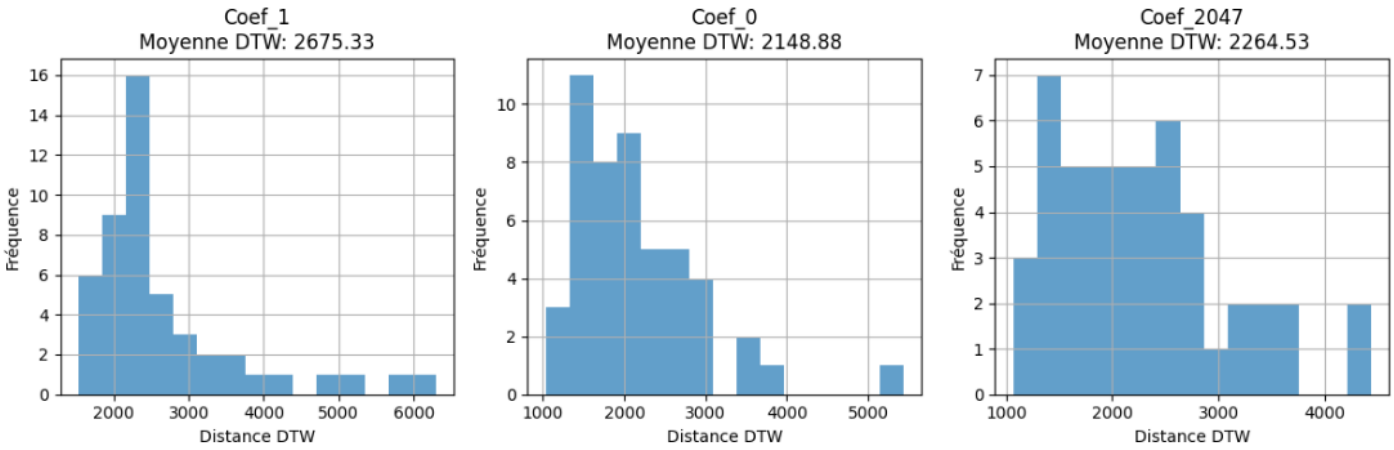
L'analyse des courbes de variance cumulée révèle des profils similaires pour les trois coefficients étudiés. Les courbes suivent toutes une progression quasi-identique formant un arc de cercle presque parfait, indiquant une distribution homogène de la variance sur l'ensemble des composantes principales.

Cette uniformité de distribution signifie qu'aucune composante ne domine significativement les autres, la variance est répartie de manière égale plutôt que concentrée sur les premières composantes. En pratique, cela traduit l'absence de patterns dominants ou de structures privilégiées dans les traces de consommation. Chaque composante capture une part équivalente d'information, suggérant que les signaux mesurés présentent une complexité uniformément distribuée dans le temps.

Du point de vue de l'analyse par canaux auxiliaires, ces résultats sont peu encourageants. L'absence de différenciation entre les profils PCA des différents coefficients indique que cette approche ne permet pas d'isoler des signatures caractéristiques exploitables. Les opérations effectuées sur les coefficients 0, 1 et 2047 ne génèrent pas de patterns de consommation suffisamment distincts pour être discriminés via une réduction dimensionnelle classique.

5.4 Calcul de distance DTW entre signaux (Dynamic Time Warping)

Nous appliquons ici l'algorithme Dynamic Time Warping (DTW) pour mesurer la similarité entre les traces de consommation d'un même coefficient à travers les 50 exécutions. Le DTW est crucial en analyse par canaux auxiliaires car il compare des signaux en tenant compte des décalages temporels potentiels entre exécutions.
L'algorithme aligne de manière optimale deux signaux en autorisant des distorsions temporelles. Contrairement à une distance euclidienne point à point, le DTW peut étirer ou comprimer l'axe temporel pour trouver la meilleure correspondance entre les patterns. Il construit une matrice de coûts et trouve le chemin minimal via programmation dynamique.
Pour chaque coefficient, la première exécution sert de référence et nous calculons la distance DTW avec les 49 autres. Une faible distance indique des traces similaires et reproductibles malgré les désynchronisations. Une distance élevée suggère une forte variabilité entre exécutions, possiblement due à du bruit ou des contre-mesures.
Les histogrammes montrent la distribution de ces distances pour chaque coefficient. Une distribution étroite autour de valeurs faibles révèle des traces très reproductibles, facilitant une attaque potentielle. Une distribution large indique plus de variabilité, rendant l'exploitation plus difficile. La comparaison entre coefficients permet d'identifier lesquels ont les comportements temporels les plus stables.



Les histogrammes obtenus montrent des distributions visuellement distinctes entre les trois coefficients. On observe que les coefficients 0 et 1 présentent des distributions plutôt centrées vers la gauche, avec des distances DTW concentrées sur des valeurs faibles. Cela indique des traces hautement reproductibles avec peu de variations temporelles entre exécutions. Le coefficient 2047, en revanche, affiche une distribution plus étalée et centrée au milieu du spectre, révélant une variabilité temporelle nettement plus importante.

Ces signatures DTW distinctes constituent une vulnérabilité potentielle, un attaquant pourrait exploiter ces différences pour identifier quel coefficient est traité à un instant donné, simplement en analysant les caractéristiques temporelles des traces. Les coefficients 0 et 1, avec leur comportement temporel stable et prévisible, seraient plus facilement reconnaissables et potentiellement plus vulnérables à une exploitation. Le coefficient 2047, avec sa plus grande variabilité, pourrait bénéficier d'une forme de protection naturelle due à cette dispersion temporelle.
Cette capacité de discrimination basée sur le DTW suggère que l'implémentation NTRU étudiée présente des fuites d'information exploitables via l'analyse des patterns temporels

5.5 Synthèse des tests d'hypothèses statistiques

Hypothèse de variabilité inter-coefficients : PARTIELLEMENT VALIDÉE
L'analyse statistique révèle des différences principalement concentrées sur les variances, les moyennes et écarts-types demeurant comparables entre coefficients. Cette observation suggère que les signatures électromagnétiques partagent des niveaux d'amplitude similaires tout en présentant des degrés de dispersion caractéristiques pour chaque coefficient.

Hypothèse de normalité des distributions : VALIDÉE
Bien que les distributions ne suivent pas une loi normale (profils multi-modaux avec quatre pics distincts), ce résultat n'est pas problématique car l'objectif principal n'est pas d'obtenir des distributions gaussiennes mais d'identifier des différences exploitables entre coefficients. Les profils de distribution observés présentent effectivement des signatures distinctes selon le coefficient, avec des positions de pics et des formes caractéristiques permettant une discrimination entre les coefficients 0, 1 et 2047.

Hypothèse de séparabilité dimensionnelle : NON VALIDÉE
Les profils PCA des trois coefficients présentent des courbes de variance cumulée remarquablement similaires, décrivant un arc plutôt régulier sans composante dominante. Cette répartition homogène de l'information sur l'ensemble des composantes principales empêche toute discrimination efficace via cette méthode de réduction dimensionnelle.

Hypothèse de stabilité temporelle : VALIDÉE
Le DTW met en évidence une reproductibilité variable selon les coefficients, haute stabilité temporelle pour les coefficients 0 et 1 (distributions DTW concentrées à gauche), variabilité accrue pour le coefficient 2047 (distribution étalée centralement). Cette hétérogénéité de stabilité temporelle constitue paradoxalement un vecteur de discrimination exploitable entre coefficients.

6 Visualisation des résultats (C2.2.1)

Ce tableau de bord présente les résultats d'analyse de traces électromagnétiques à destination d'ingénieurs élec/RF et s'articule autour de trois objectifs : contrôle de qualité (vérification de l'intégrité des mesures pour garantir des données exploitables), visualisation comparative (comparaison rapide des signaux et sélection des traces pertinentes pour l'entraînement de modèles ou la conduite d'attaques), et extraction d'indicateurs (caractéristiques fréquentielles, temporelles et statistiques pour nourrir l'explicabilité des attaques par canaux auxiliaires). Conçu pour des jeux de données massifs, le dashboard accélère et simplifie l'analyse, facilite la reproductibilité des expériences et réduit le délai entre collecte et exploitation, tout en privilégiant lisibilité et facilité d'utilisation grâce à une mise en forme soignée (échelles pertinentes, légendes claires, codes couleur et graphiques adaptés).

Dans la suite, je montrerai des exemples du tableau de bord qui reprennent les éléments utiles à l'analyse exploratoire : vues de contrôle qualité, comparaisons de signaux et extraction d'indicateurs. Chaque exemple s'appuie sur les hypothèses formulées puis validées précédemment, et met en avant les métriques clés (moyenne, écart-type, variance) ainsi que l'alignement DTW pour quantifier la similarité temporelle des traces.

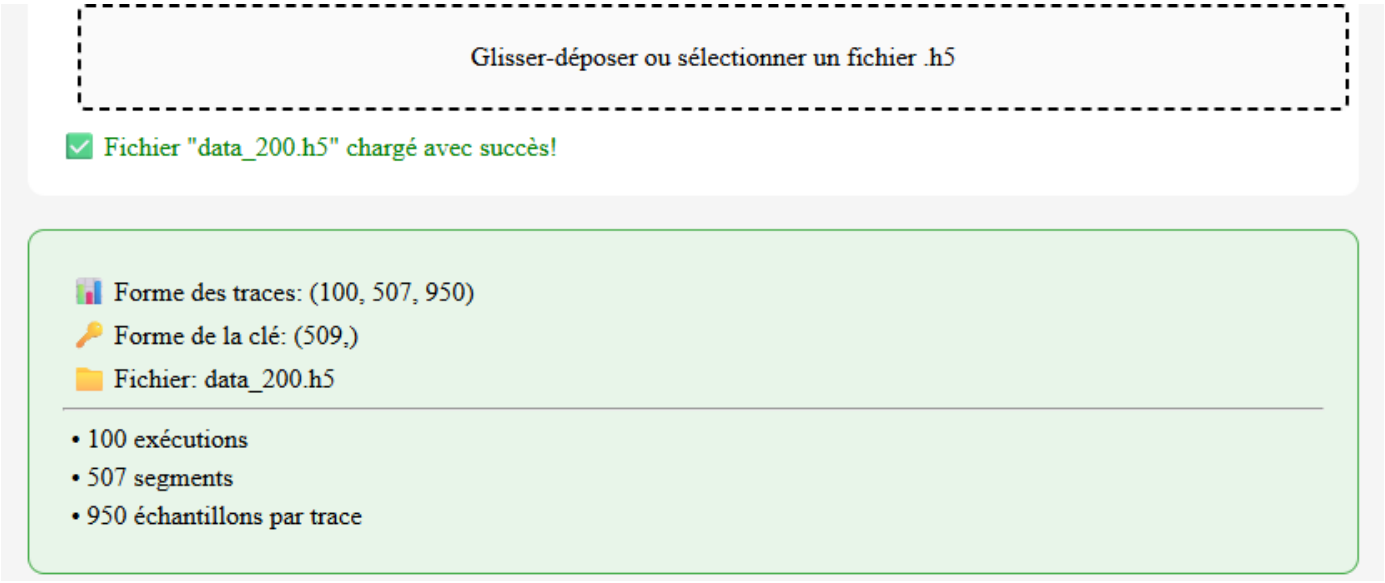
6.1 Plotly Dash

Plotly Dash me permet de transformer directement mon code Python (NumPy/SciPy, h5py, scikit-learn) en un tableau de bord web interactif, sans écrire de JavaScript. C'est idéal car **je réutilise mes traitements python existants** et je les expose via des contrôles (sliders, sélecteurs, fenêtres d'analyse). Les graphiques Plotly offrent nativement zoom, pan, survol d'indices/valeurs, sélection lasso et export d'images, ce qui accélère l'analyse exploratoire et la comparaison de signaux ce qui est très adapté à l'**exploration/analyse de données orienté ingénierie** même s'il existe des outils serait plus customisable pour la mise en forme.

Côté performance, Dash s'appuie sur le navigateur et WebGL pour rendre de longues séries, je peux décimer/échantillonner, tout en manipulant des millions de points grâce à des conversions et à un fenêtrage interactif. L'architecture Flask de Dash facilite le déploiement local, le packaging Docker et l'intégration d'authentification si nécessaire. Pour les utilisateurs finaux, **un simple navigateur en labo suffit**. Enfin, Dash favorise la traçabilité et la reproductibilité, les paramètres d'analyse sont centralisés, l'état peut être sauvegardé (**export CSV/JSON**), et chaque vue est "source of truth" pour discuter des résultats entre ingénieurs élec/RF et data. Dans le cadre des attaques par canaux auxiliaires et de l'explicabilité, cette combinaison d'**interactivité, de calcul Python natif et de partage simple** fait de Dash l'outil le plus efficace **pour passer rapidement de la collecte à l'exploitation des données**, avec une lisibilité et une facilité d'utilisation adaptées au public cible.

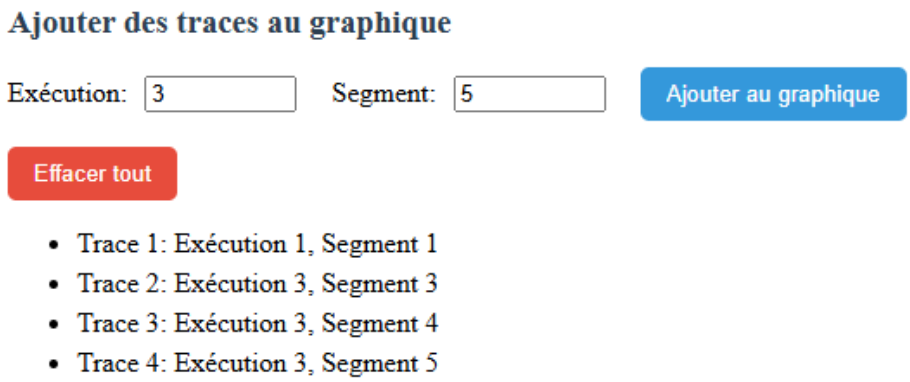
6.2 Présentation du tableau de bord interactifs

Pour commencer, le tableau de bord propose un champ d'import qui permet de sélectionner le fichier H5 contenant les données, une fois chargé, on réalise une vérification de base (présence d'un fichier au format .H5) puis on affiche clairement les dimensions du dataset afin de valider l'exactitude du chargement. Pour lever toute ambiguïté, ces dimensions sont immédiatement annotées avec le nom et la signification de chaque axe, par exemple : exécutions × segments × échantillons, où exécutions désigne les répétitions de mesure, segments les découpes/événements dans une exécution, et échantillons les points temporels du signal. Cette légende contextuelle rappelle également les conventions utilisées (p. ex. 100 correspond au nombre d'exécutions réalisées avec la même clé), ce qui facilite la lecture et l'alignement entre experts élec/RF et data lors des premières étapes d'analyse exploratoire.

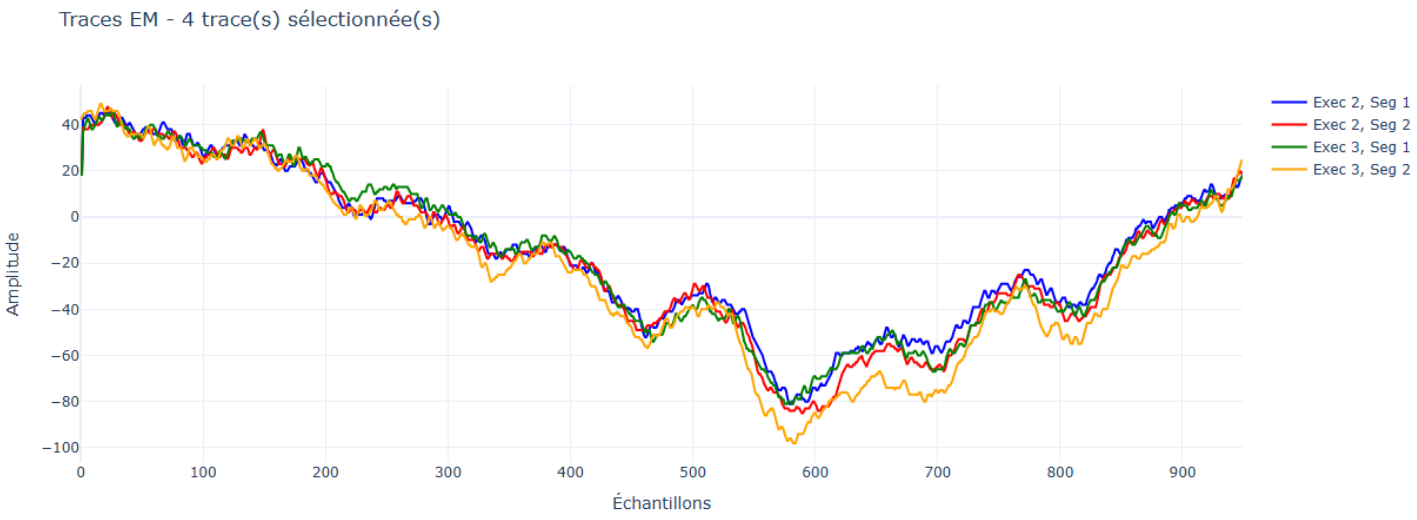


Lorsque la validation est réussie, la couleur de fond devient verte et un logo de validation s'affiche, offrant un retour visuel clair, agréable et rassurant pour l'utilisateur.

Une fois le fichier H5 chargé et validé, le tableau de bord déverrouille des sélecteurs d'Exécution et de Segment qui permettent de cibler précisément les traces à analyser. On peut ajouter les traces une par une via le bouton bleu "Ajouter au graphique" : à chaque ajout, la courbe correspondante s'affiche et une liste récapitulative se met à jour pour garder une vue claire de la sélection en cours (par ex. : Trace 1 : Exécution 1, Segment 1 ; Trace 2 : Exécution 3, Segment 3). Ce récap joue le rôle de pense-bête visuel et évite de perdre le fil lorsqu'on compare plusieurs signaux. Pour fluidifier l'exploration ou repartir d'une page blanche, un bouton rouge "Effacer tout" supprime instantanément les traces sélectionnées ainsi que le graphique associé. Les champs numériques sont bornés par les dimensions détectées (exécutions, segments) afin de prévenir les erreurs de saisie ; l'interface réagit en temps réel, ce qui rend la manipulation rapide et intuitive. Le flux "charger → sélectionner → récapituler → comparer → réinitialiser" améliore la lisibilité, réduit les erreurs et accélère l'itération, tout en gardant l'écran épuré et focalisé sur les traces réellement pertinentes pour l'analyse.



Une fois les traces sélectionnées, ci-dessous le graphique principal qui superpose toutes les courbes en pleine largeur pour offrir une forte granularité d'observation. Les signaux apparaissent sur fond blanc avec des couleurs vives et bien distinctes comme bleu jaune rouge ce qui évite la confusion visuelle. En haut du graphique figure le nombre de traces affichées et à gauche une légende indique pour chaque courbe le numéro d'exécution et de segment par exemple « exec 3 seg 1. Les axes sont indiqués avec abscisse en échantillons et ordonnée en amplitude (dBm). Avec ce type de schéma Plotly on bénéficie de la barre d'outils du graphique aussi appelée modebar qui permet de zoomer de dézoomer de se déplacer et d'enregistrer une capture ainsi que d'exporter les données en CSV ou en JSON sans ajouter de bouton spécifique de notre côté.



Ci-dessous figure un récapitulatif des traces sélectionnées avec pour chacune l'exécution, le segment et la valeur associée de la clé (ou coefficient) choisi. Cette liste facilite la vérification de la sélection, le repérage rapide d'une trace précise et la traçabilité entre signaux et paramètres retenus.

Valeurs des clés

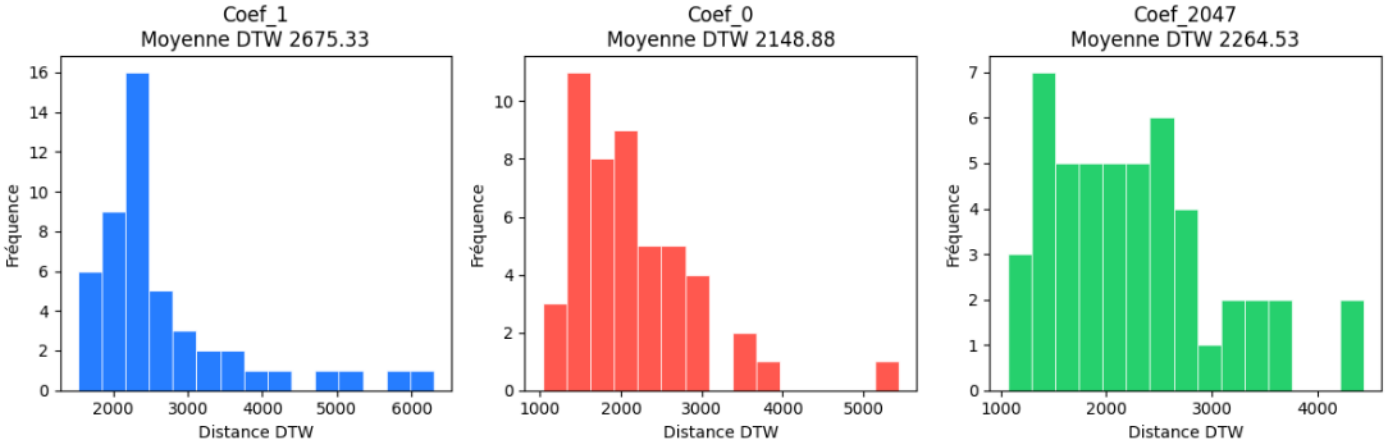
Trace	Position	Valeur de la clé
Trace 1	Exec 1, Seg 2	**1**
Trace 2	Exec 1, Seg 3	**1**
Trace 3	Exec 2, Seg 4	**0**
Trace 4	Exec 2, Seg 5	**2047**
Trace 5	Exec 3, Seg 5	**2047**
Trace 6	Exec 4, Seg 4	**0**

Ici on récapitule les métriques clés pour chaque trace, la moyenne, l'écart-type, le minimum, le maximum et la médiane. Ce panneau permet de vérifier rapidement la cohérence des signaux, de repérer les Outliers et d'appuyer les comparaisons entre exécutions et segments. Les valeurs se mettent à jour en fonction de la sélection courante, avec les mêmes unités que le graphique pour rester cohérent et facile à lire.

Statistiques des traces

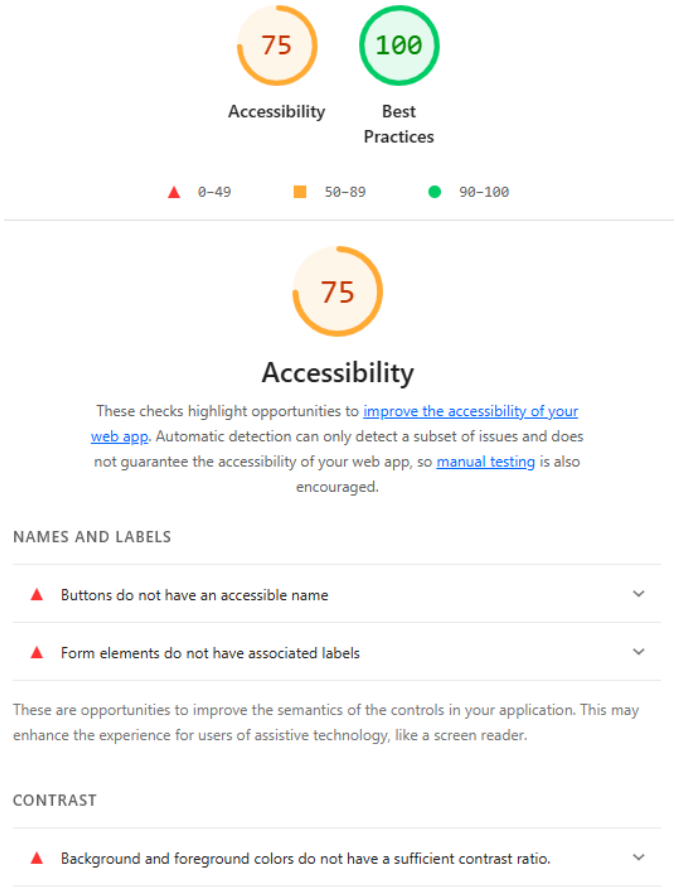
Trace	Moyenne	Écart-type	Min	Max	Médiane
Trace 1	-16.12	38.01	-84.00	55.00	-16.00
Trace 2	-15.70	37.68	-81.00	56.00	-17.00
Trace 3	-16.63	38.55	-91.00	50.00	-13.00
Trace 4	-15.31	35.32	-83.00	53.00	-18.00
Trace 5	-15.10	34.56	-80.00	51.00	-18.00
Trace 6	-15.15	32.47	-78.00	45.00	-15.00

Pour terminer, ci-dessous se trouvent les graphiques DTW, pour Dynamic Time Warping, une mesure d'alignement temporel qui compare des séries même lorsqu'elles sont décalées ou dilatées dans le temps, afin d'évaluer la similarité des signaux. Ils comparent les trois clés sélectionnées et reprennent sensiblement la même présentation que dans le notebook, mais avec des couleurs vives et bien distinctes pour faciliter la lecture et la comparaison d'un coup d'œil entre les trois panneaux. Chaque graphique affiche la distribution des distances DTW et juste au-dessus sont récapitulées la principale métrique DTW c'est-à-dire la moyenne, ce qui permet d'apprécier rapidement la dispersion et la robustesse des signaux pour chaque clé.



6.3 Prise en compte des personnes malvoyantes

Pour évaluer l'accessibilité de mon dashboard, j'ai utilisé Google Lighthouse et j'obtiens une note d'accessibilité de 70/100, insuffisante au regard des bonnes pratiques. L'outil met en évidence des points d'amélioration prioritaires sur la sémantique et l'accessible *name* des composants, ajouter un nom explicite aux boutons (aria-label) et remplacer autant que possible les div génériques par des balises sémantiques (header, nav, main, section, footer, button, label). Il recommande aussi d'améliorer les couleurs de fond et les contrastes pour atteindre les ratios requis, en particulier pour les textes et éléments interactifs. Je vais donc corriger ces aspects en priorité pour augmenter la lisibilité, la navigation au clavier et la compatibilité lecteurs d'écran, avec un objectif d'atteindre un score supérieur à 90.



J'ai déjà appliqué les recommandations Lighthouse et fait passer l'accessibilité de mon dashboard à 85/100, mais sans l'intervention d'un développeur web je n'arrive pas à aller plus loin, il reste des chantiers pointus de sémantique HTML, d'ARIA (rôles, états, noms accessibles), et de contrastes qui demandent un affinage. Faire intervenir un spécialiste permettra de se rapprocher de 100/100 et c'est crucial pour les personnes malvoyantes car les lecteurs d'écran s'appuient sur une structure sémantique propre pour annoncer correctement les sections, boutons et états, des noms accessibles clairs évitent les "bouton sans étiquette", des contrastes suffisants et des focus visibles rendent la navigation clavier fiable surtout pour les malvoyants, une palette sûre pour le daltonisme et la non-dépendance à la couleur assurent que l'information reste compréhensible. Enfin, le zoom et les raccourcis de navigation réduisent l'effort cognitif. Investir ce dernier objectif technique améliore l'autonomie, la vitesse d'usage et le confort des utilisateurs malvoyants, tout en renforçant la conformité aux bonnes pratiques d'accessibilité.

7 Recommandations (C2.2.2)

J'ai développé un tableau de bord Plotly Dash pour analyser des traces électromagnétiques à grande échelle. L'outil contrôle la qualité à l'import, superpose et compare les signaux, et calcule des indicateurs clés ainsi que DTW pour mesurer la similarité malgré des décalages temporels. À partir de ces éléments, voici des recommandations structurées, synthétiques et argumentées pour éclairer la décision.

Recommandations :

Adopter le Dashboard comme point d'entrée unique

- Action : officialiser le tableau de bord pour la vérification, l'exploration et la comparaison des signaux.
- Pourquoi : aligner les équipes sur un même référentiel d'indicateurs, réduire les allers-retours et sécuriser l'interprétation.
- Bénéfice : temps d'analyse réduit, décisions plus homogènes.

Standardiser le format H5 et la collecte

- Action : publier un schéma H5 versionné, documenté, avec contrôles d'intégrité obligatoires à l'import.
- Pourquoi : limiter l'hétérogénéité des acquisitions et les erreurs de parsing.
- Bénéfice : intégration plus fluide de nouveaux jeux de données et comparaisons plus fiables.

Renforcer l'accessibilité pour un usage inclusif

- Action : confier à un développeur frontend la sémantique HTML, le contraste, l'ordre de tabulation et les alternatives textuelles, afin de dépasser 90/100d'accesibilité sur Lighthouse.
- Pourquoi : l'accessibilité améliore l'expérience de tous, garantit l'usage par les personnes malvoyantes et réduit les risques de non-conformité.
- Bénéfice : adoption élargie, navigation clavier fiable, lecture par lecteur d'écran, meilleure lisibilité.

Documenter la traçabilité et la reproductibilité

- Action : associer à chaque dataset une "carte de jeu de données" (provenance, paramètres de capture, hypothèses, limites) et permettre l'export CSV/JSON des vues.
- Pourquoi : garantir la reprise d'analyse et l'auditabilité des décisions.
- Bénéfice : échanges plus efficaces entre équipes élec/RF et data, capitalisation durable.

Optimiser la performance à grande échelle

- Action : Afficher seulement la portion utile du signal (fenêtrage), réduire le nombre de points sans perdre l'essentiel (décimation), réutiliser les résultats déjà calculés (cache) et lancer plusieurs calculs DTW en même temps si les jeux de données deviennent très gros.
- Pourquoi : garder une interface fluide sur des millions de points.
- Bénéfice : meilleures performances sans sacrifier la précision.

Valider l'industrialisation du dashboard comme outil officiel, lancer la normalisation H5 et l'automatisation des rapports DTW, et allouer un budget ciblé pour finaliser l'accessibilité. Ces mesures traduisent les données en recommandations claires et actionnables, et aident concrètement les décideurs à établir leurs stratégies.

8 Guide utilisateur (C2.3.1)

1) Enjeux et objectifs pédagogiques

Ce dashboard Plotly Dash a été conçu pour accélérer et fiabiliser l'analyse de traces électromagnétiques (EM) par des ingénieurs élec/RF et data. Il permet de contrôler la qualité des données à l'import, de comparer rapidement des signaux et d'extraire des indicateurs utiles à la décision. Les objectifs de la formation sont de savoir charger un fichier H5 structuré et vérifier son intégrité, sélectionner et superposer des traces pour comparer exécutions et segments, interpréter les métriques clés (moyenne, écart-type, min, max, médiane) et la similarité temporelle via DTW et savoir documenter et exporter les résultats pour les partager.

2) Public cible et prérequis

Public : ingénieurs élec/RF, data analysts, test & validation, R&D sécurité.
Prérequis : notions de base en traitement du signal (échantillonnage, amplitude), connaissance du contexte d'acquisition.

3) Vue d'ensemble de l'outil

- Import H5 : sélection du fichier contenant traces et, si disponible, key.
- Validation automatique : affichage des dimensions détectées exécutions × segments × échantillons. Un rappel sémantique précise chaque axe. Par ex. 100 correspond au nombre d'exécutions avec la même clé.
- Retour visuel : si tout est correct, la zone passe en vert et un logo validé s'affiche.
- Sélecteurs : champs Exécution et Segment pour ajouter des courbes une par une au graphique.
- Récapitulatif : liste des traces sélectionnées, ex. Trace 1 Exec 1 Seg 1 ; Trace 2 Exec 3 Seg 3.
- Actions rapides : bouton bleu pour Ajouter au graphique ; bouton rouge pour Effacer tout.
- Graphique principal : pleine largeur, fond blanc, couleurs vives distinctes ; légende claire avec Exec/Seg.
- Modebar Plotly : zoom, pan, capture PNG et export des données (CSV/JSON) sans code.
- Panneau "Clés / Coefficients" : table récapitulant la clé ou le coefficient associé à chaque trace.
- Panneau "Statistiques" : moyenne, écart-type, min, max, médiane par trace.
- Analyse DTW : sélection d'une exécution de référence, choix des segments et exécutions à comparer, histogrammes des distances DTW et stats par segment.

4) Prise en main en 10 minutes (pas à pas)

- Charger le fichier via Importer un .h5. Vérifier les dimensions et la présence de traces (et key si disponible). Retour vert attendu.
- Ajouter des courbes : saisir Exécution et Segment, cliquer Ajouter au graphique. Répéter pour plusieurs traces.
- Lire la légende et les axes : vérifier Exec/Seg et unités dBm. Utiliser la modebar pour zoomer, se déplacer, capturer l'image, exporter CSV/JSON.
- Consulter les statistiques : contrôler moyenne/écart-type/min/max/médiane pour chaque trace.
- Comparer avec DTW : choisir l'exécution de référence, sélectionner un ou plusieurs segments, lancer le calcul et interpréter les histogrammes et les statistiques DTW.
- Nettoyer si besoin avec Effacer tout et recommencer une nouvelle sélection.

5) Cas d'usage guidés

- Contrôle de qualité : après import, confirmer que la forme des traces est (n_exec, n_seg, n_samples) et que les unités d'amplitude sont cohérentes.

- Comparaison inter-exécutions : superposer 2–3 traces d'un même segment pour détecter dérive ou bruit.
- Sélection pour entraînement : utiliser stats + DTW pour retenir les traces stables et similaires à la référence.
- Rapport rapide : exporter l'image et les données brutes en CSV/JSON pour les annexer à un CR.

6) Interpréter les métriques

- Moyenne / Médiane : niveau central du signal ; la médiane est robuste aux outliers.
- Écart-type / Min / Max : dispersion et bornes, utiles pour repérer du bruit ou des saturations.
- DTW (Dynamic Time Warping) : mesure de similarité temporelle robustes aux décalages et dilatations.
    - Faible DTW → signaux proches de la référence.
    - DTW élevé → différences de forme ou de synchronisation.

- Interpréter l'histogramme par segment pour juger de la variabilité inter-exécutions.

7) Bonnes pratiques et erreurs fréquentes

Bonnes pratiques

- La forme des données à l'import et l'unité en ordonnée.
- Utiliser la légende Exec/Seg et nommer explicitement les segments dans la documentation projet.
- De très longues séries, privilégier des fenêtres pertinentes avant d'interpréter les statistiques.
- Conserver un journal de sélection des traces retenues et des raisons du tri.

À éviter

- Confondre Échantillons et temps absolu.
- Conclure uniquement sur min/max sans regarder la dispersion ou la forme du signal.
- Oublier d'exporter et d'archiver les paramètres de sélection et les états d'analyse.

8) Accessibilité et inclusion

- L'interface offre des couleurs contrastées, une légende textuelle explicite et ne dépend pas uniquement de la couleur.
- Navigation clavier possible sur les sélecteurs et boutons ; retour visuel à la validation.
- Pour les malvoyants, privilégier la modebar pour zoomer, et le tableau de données (équivalent textuel) qui se lit facilement avec un lecteur d'écran.
- Un renforcement web (structure HTML, contrastes) est en cours pour viser un score Lighthouse d'accessibilité supérieur à 90.

9) FAQ rapide

- Mon fichier ne charge pas → vérifier qu'il contient traces au format 3D et que key est facultatif.
- Je ne vois pas de courbe → ajouter au moins une trace via les sélecteurs puis "Ajouter au graphique".
- les données → utiliser la modebar pour la capture PNG et l'export CSV/JSON.
- Comprendre DTW → choisir une exécution de référence, sélectionner des segments, lancer le calcul et lire histogramme + stats.

10) Plan de formation recommandé

- Session 1 — 45 min : prise en main, import H5, sélection de traces, lecture du graphique, export.
- Session 2 — 45 min : interprétation des statistiques, DTW, cas d'usage comparatifs, bonnes pratiques et pièges.

9 Documentation technique, Dashboard d'analyse des traces EM (C2.3.2)

1) Public concerné

- Ingénieurs élec/RF pour la validation des acquisitions et la comparaison de signaux.
- Data analysts / sécurité pour la sélection de traces pertinentes.
- Test & validation / R&D pour un support traçable et partageable.

2) Sources de données

Origine : fichiers HDF5 (.h5) issus de campagnes de mesures EM.
Jeux requis :

- traces — tenseur 3D de forme (n_exec, n_seg, n_samples)
  - n_exec : nombre d'exécutions d'une même opération
  - n_seg : segments/événements dans une exécution
  - n_samples : échantillons temporels du signal
  - Types supportés : int8, int16, float32. Unités affichées en dBm
- key (optionnel) — vecteur 1D associé aux segments, p. ex. clé/coef choisi

Périmètre : upload local, comparaisons intra-fichier.
Contrôles à l'import : présence de traces, dimension 3D, cohérence n_seg vs key ; retour visuel vert + icône validé si OK.

3) Fonctionnement général

1. Chargement du .h5 puis vérification de la forme et affichage des dimensions.
2. Sélection interactive d'Exécution et Segment pour ajouter des courbes.
3. Visualisation pleine largeur, couleurs vives distinctes, légende Exec/Seg, axes normalisés (abscisse = Échantillons, ordonnée = Amplitude dBm).
4. Récapitulatif des traces choisies avec clé/coef associé.
5. Statistiques par trace et analyse DTW par segment avec histogrammes + stats.
6. Export via la modebar Plotly (zoom, pan, capture PNG, export CSV/JSON).

4) Méthodes de calcul

Statistiques : moyenne, écart-type, min, max, médiane.
DTW (Dynamic Time Warping) : mesure la similarité temporelle robuste aux décalages/dilatations ; complexité O(N*M). Dans le dashboard : référence d'exécution, distances par segment, histogrammes et stats.
Bonnes pratiques : fenêtrer les signaux, décimer prudemment, interpréter DTW avec les stats.

5) Indicateurs — description technique et fonctionnelle

- Qualité des données : validation de forme, présence des champs, unités explicites — Go/No-Go d'exploitation.
- Statistiques par trace : moyenne, écart-type, min, max, médiane — repérage d'outliers et stabilité.
- DTW par segment : distribution des distances vs référence — similarité de forme, robustesse inter-exécutions.

6) Traçabilité et reproductibilité

Export PNG, CSV/JSON ; versionnage des dépendances ; conteneur Docker reproductible ; journalisation légère des paramètres d'analyse ; dataset card recommandée.

7) Déploiement en conteneur (Docker) mode d'emploi

Build : docker build -t em-dashboard:latest .
Run : docker run --rm -p 8050:8050 em-dashboard:latest
Variables : PORT, WORKERS, THREADS, TIMEOUT (ex. -e PORT=8052 -p 8052:8052).

Annexes — fichiers de déploiement

Dockerfile

```
FROM python:3.11-slim
ENV PYTHONDONTWRITEBYTECODE=1 PYTHONUNBUFFERED=1 PIP_NO_CACHE_DIR=1 PORT=8050
WORKERS=2 THREADS=4 TIMEOUT=120
RUN apt-get update && apt-get install -y --no-install-recommends build-essential libhdf5-dev curl && rm -rf
/var/lib/apt/lists/*
RUN useradd -ms /bin/bash appuser
WORKDIR /app
COPY requirements.txt /app/requirements.txt
RUN pip install --upgrade pip && pip install -r /app/requirements.txt
COPY app_with_dtw.py /app/app_with_dtw.py
USER appuser
EXPOSE ${PORT}
CMD gunicorn --bind 0.0.0.0:${PORT} --workers ${WORKERS} --threads ${THREADS} --timeout ${TIMEOUT}
app_with_dtw:app.server
```

.dockerignore

```
# Keep only what's needed
!app_with_dtw.py
!requirements.txt
# Common ignores
__pycache__/
*.py[cod]
*.ipynb
*.h5
*.csv
*.json
.env
.git/
.DS_Store
```

requirements.txt

```
dash>=2.15
plotly>=5.22
h5py>=3.11
numpy>=1.24
gunicorn>=21.2
```

10 Conclusion

Ce jeu de données est exigeant car on étudie des traces électromagnétiques longues, variabilité inter-exécutions, segments hétérogènes, et un lien très subtil entre signaux et clés/coefficients.

Pourtant, le travail mené avec le contrôle d'intégrité à l'import, visualisation comparative en pleine largeur, indicateurs simples mais robustes (moyenne, écart-type, min/max, médiane) et mesure de similarité via DTW pose les fondations indispensables de l'analyse exploratoire dans notre contexte élec/RF.

Le tableau de bord facilite le passage du brut au compréhensible, on charge le .h5, on lit immédiatement les dimensions exécutions × segments × échantillons, on superpose des courbes bien identifiées Exec/Seg, on exporte les images et données, et l'on dispose d'histogrammes DTW par segment pour objectiver la stabilité des signaux. Surtout, l'étape compréhension des traces + choix de caractéristiques (fréquentielles, temporelles, statistiques) devient un pont critique entre la collecte et l'application d'attaques SCA, elle oriente la sélection des exécutions pertinentes, guide le fenêtrage, la normalisation et l'alignement, et réduit le risque de conclusions biaisées. Cette analyse nous permet d'identifier les paramètres clés sur lesquels agir pour lancer des attaques SCA en faisant les bons choix.

À court terme, ce socle améliore la reproductibilité et la vitesse d'itération, à moyen terme, il prépare l'industrialisation des métriques (dont DTW) et une gouvernance de données claire.

À long terme, l'importance des attaques par canaux auxiliaires ne va que croître, généralisation d'objets embarqués et IoT, déploiement de schémas post-quantiques à évaluer sous contraintes réelles, exigences accrues de certification et de conformité.

Dans ce contexte, disposer d'une chaîne outillée de la qualité des données jusqu'à l'évaluation SCA n'est plus un luxe, c'est une condition pour tester la vulnérabilité avec rigueur, éclairer les décisions d'ingénierie et renforcer durablement la sécurité des systèmes.