

Université Paris Cité

Master 2 Machine Learning pour la Science des Données

Clustering appliqué aux séries temporelles

Rapport de projet

Année Universitaire: 2025/2026

Melissa MERABET
Ouarda BOUMANSOUR

Module : Clustering
Professeur responsable : Allou SAME

Table des matières

Table des figures	ii
Liste des tableaux	ii
1 Introduction	iii
Chapitre 1 Contexte	iv
1.1 Clustering de séries temporelles	iv
1.2 Problématique	iv
Chapitre 2 Données et prétraitement	v
2.1 Présentation des données	v
2.2 Analyse descriptive des données	v
2.2.1 Variabilité temporelle des signaux	vi
Chapitre 3 Approches étudiées	viii
3.1 K-means et DTW	viii
3.2 K-Shape	ix
3.3 Transformée de Fourier et K-Means	x
3.4 Transformée de Fourier avec HDBSCAN	xi
3.5 Résultats expérimentaux	xii
3.5.1 Qualité de clustering	xii
3.5.2 Visualisation des clusters	xiii
3.5.2.1 DBA K-Means	xiii
3.5.2.2 K-Shape	xiv
3.5.2.3 FFT + PCA + K-Means	xiv
3.5.2.4 FFT + UMAP + HDBSCAN	xv
3.5.3 Synthèse	xvi

Table des figures

2.1	Variabilité temporelle des neuf signaux inertIELS (moyenne \pm variabilité sur 128 échantillons.	vii
3.1	Clusters et matrices de confusion (DBA K-Means).	xiii
3.2	Clusters et matrices de confusion (K-shape).	xiv
3.3	Clusters et matrices de confusion (FFT+PCA+KMEANS).	xv
3.4	Clusters et matrices de confusion (FFT + UMAP + HDBSCAN).	xvi

Liste des tableaux

2.1	Répartition des 347 fenêtres temporelles selon les 6 activités.	vi
3.1	Résumé des performances de clustering pour les différentes approches.	xii

1 Introduction

La reconnaissance d'activité humaine à partir de capteurs inertIELS est aujourd'hui largement utilisée dans des domaines comme le sport, la santé ou encore l'analyse du mouvement. Ces capteurs enregistrent des signaux sous forme de séries temporelles multivariées, par exemple des accélérations ou des vitesses. L'objectif est alors de reconnaître automatiquement l'activité réalisée : marcher, s'asseoir, monter des escaliers, rester immobile, etc.

Cependant, regrouper ces signaux en différentes catégories d'activités n'est pas trivial. Les séries temporelles peuvent être bruitées, déphasées ou varier selon les individus, ce qui rend les méthodes de clustering classiques peu adaptées. Pour cette raison, plusieurs familles d'approches spécifiques aux données temporelles existent : des méthodes fondées sur la forme du signal, d'autres sur des distances plus robustes, et enfin des approches transformant les séries dans un autre espace avant d'appliquer un algorithme de clustering.

Dans ce projet, nous appliquons et comparons quatre méthodes différentes sur un jeu de données inertiel comportant neuf signaux synchronisés. L'objectif est d'évaluer leur capacité à regrouper automatiquement les différentes activités humaines, et d'analyser leurs performances, leurs avantages et leurs limites.

La suite du rapport présente les notions de base liées au clustering de séries temporelles, les algorithmes utilisés, les résultats expérimentaux obtenus ainsi qu'une discussion comparative des approches.

1

Contexte

1.1 Clustering de séries temporelles

Le clustering de séries temporelles vise à regrouper automatiquement des signaux en fonction de leur similarité, sans utiliser d'étiquettes. Contrairement à des données classiques, les séries temporelles sont des objets ordonnés dans le temps : leur dynamique, leur forme globale et les décalages éventuels entre signaux jouent un rôle important. Ainsi, l'utilisation directe de distances classiques comme la distance euclidienne est souvent insuffisante pour capturer la véritable similarité entre deux signaux.

Plusieurs difficultés rendent le clustering de séries temporelles particulièrement complexe :

- les signaux peuvent être **déphasés** : deux activités identiques peuvent être réalisées plus ou moins tôt dans la fenêtre observée ;
- les séries sont souvent **bruitées** et varient selon les individus ;
- les signaux sont **multivariés** (neuf variables dans notre cas) et peuvent présenter des amplitudes ou des fréquences différentes.

1.2 Problématique

Dans ce projet, notre problématique principale est la suivante : *comment regrouper des séries temporelles multivariées issues de capteurs inertIELS en différentes activités humaines, en l'absence de labels, tout en tenant compte des décalages temporels, du bruit et de la variabilité du mouvement ?*

2

Données et prétraitement

2.1 Présentation des données

Les données utilisées dans ce projet sont issues des travaux de Anguita et al. (2013). Elles correspondent à plusieurs activités humaines courantes, telles que la marche, la station assise ou debout, ainsi que la montée et la descente d'escaliers. Au cours du protocole expérimental, neuf signaux inertIELS ont été enregistrés avec une fréquence d'échantillonnage de 50 Hz (une mesure toutes les 0.02 secondes). Les variables collectées sont les suivantes :

- trois accélérations réelles : `accm_x`, `accm_y`, `accm_z`;
- trois accélérations estimées (filtrées de la gravité) : `acce_x`, `acce_y`, `acce_z`;
- trois vitesses selon les trois axes : `vit_x`, `vit_y`, `vit_z`.

Pour l'analyse, les signaux ont été découpés en fenêtres de 128 échantillons (environ 2.5 secondes chacun). Chaque variable est donc représentée par une matrice de taille 347×128 , où chaque ligne correspond à une fenêtre temporelle. Les neuf signaux étant parfaitement synchronisés, l'ensemble forme un jeu de données multivarié composé de 347 fenêtres alignées. Un fichier de labels indique l'activité associée à chaque fenêtre ; il n'est utilisé que pour évaluer les performances du clustering.

2.2 Analyse descriptive des données

Avant d'appliquer les méthodes de clustering, il est important de comprendre la structure du jeu de données et la répartition des différentes activités. Le tableau 2.1 indique le nombre de fenêtres temporelles disponibles pour chaque activité.

Label	Nombre d'observations	Activité
1	95	Marche
2	53	Montée d'escaliers
3	49	Descente d'escaliers
4	47	Position assise
5	53	Position debout
6	50	Position allongée

TABLE 2.1 – Répartition des 347 fenêtres temporelles selon les 6 activités.

La répartition des fenêtres entre les différentes activités est relativement équilibrée : aucune classe n'est minoritaire, et la plus grande (la marche) ne dépasse pas 27% du jeu de données. Cela garantit que le clustering ne sera pas biaisé par une dominance excessive d'une activité par rapport aux autres.

2.2.1 Variabilité temporelle des signaux

La figure 2.1 illustre la variabilité des neuf signaux inertIELS pour l'ensemble des activités, en représentant pour chaque variable la moyenne temporelle accompagnée d'un intervalle de variation. On observe des différences claires entre activités statiques et dynamiques.

Les signaux d'accélération et de vitesse présentent des comportements très distincts selon le type d'activité :

- **Activités statiques (assis, debout, allongé).** Les signaux présentent une très faible variabilité, avec des amplitudes presque constantes. Cela traduit l'absence de mouvement, en particulier pour les variables `acce_x,y,z` et `vit_x,y,z`. Les accélérations mesurées (`accm`) conservent cependant une valeur moyenne non nulle, ce qui correspond à l'effet constant de la gravité dépendant de l'orientation du corps.
- **Activités dynamiques (marche, montée et descente d'escaliers).** Les signaux présentent des oscillations régulières, une variabilité beaucoup plus élevée et une énergie nettement supérieure. Les accélérations estimées et les vitesses capturent efficacement le rythme du mouvement humain.

Ces premières observations confirment que les séries temporelles contiennent une structure exploitable pour le clustering : les activités dynamiques et statiques génèrent des signatures temporelles bien distinctes.

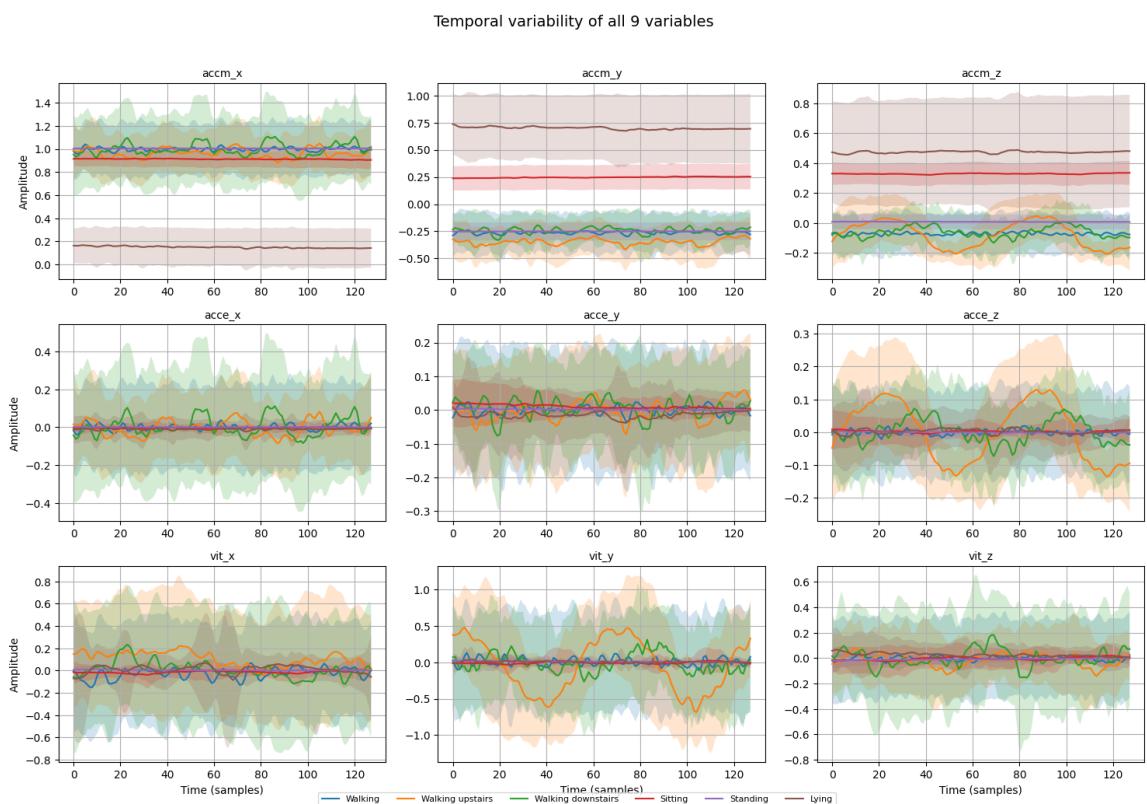


FIGURE 2.1 – Variabilité temporelle des neuf signaux inertIELS (moyenne \pm variabilité) sur 128 échantillons.

3

Approches étudiées

3.1 K-means et DTW

La première approche étudiée repose sur une variante du k-means adaptée aux séries temporelles, où la distance euclidienne est remplacée par la distance DTW (Dynamic Time Warping). Cette distance permet d'aligner deux signaux dans le temps, ce qui la rend robuste aux décalages temporels ou aux variations de vitesse.

Distance DTW : La distance DTW correspond au coût minimal parmi l'ensemble des chemins possibles alignant les deux séries, elle est définie par l'expression suivante :

$$d_{\text{DTW}}(\mathbf{x}, \mathbf{y}) = \min_{r \in M} \sum_{i=1}^m |x_{a_i} - y_{b_i}|^2$$

où M est l'ensemble des chemins d'alignement $r = ((a_1, b_1), \dots, (a_m, b_m))$ vérifiant :

$$a_i, b_i \in \{1, \dots, N\}, \quad a_1 = b_1 = 1, \quad a_m = b_m = N,$$

$$a_{i+1} \in \{a_i, a_i + 1\}, \quad b_{i+1} \in \{b_i, b_i + 1\}.$$

Principe de l'algorithme DBA k-means : Le K-Means classique repose sur une moyenne arithmétique des données, ce qui n'est pas adapté aux signaux alignés via DTW. L'algorithme DBA (*DTW Barycenter Averaging*) calcule un centroïde en tenant compte de l'alignement DTW entre toutes les séries du cluster. L'algorithme suit les étapes :

1. Initialisation de k centroïdes (méthode k-means++ dans notre implémentation) ;

-
2. Assignation des séries au cluster le plus proche selon la distance DTW ;
 3. Recalcul des centroïdes avec l'algorithme DBA ;
 4. Répétition jusqu'à convergence ou jusqu'au nombre maximal d'itérations.

Implémentation et optimisation des hyperparamètres : L'algorithme a été implémenté avec la classe `TimeSeriesKMeans` de `tslearn` en utilisant la distance DTW. Les séries sont d'abord normalisées, puis clusterisées. Les principaux hyperparamètres ont été optimisés à l'aide d'un grid search basé sur plusieurs métriques. Les meilleurs résultats ont été obtenus avec les valeurs suivantes : $n_clusters = 6$, $n_init = 25$, $max_iter = 50$, $max_iter_barycenter = 50$, $sakoe_chiba_radius = 20$. Ces réglages constituent un bon compromis entre qualité des regroupements et temps de calcul.

3.2 K-Shape

La seconde approche étudiée est l'algorithme **K-Shape**, une méthode de clustering dédiée aux séries temporelles univariées. Contrairement à DBA k-means, cette méthode repose sur une mesure de similarité fondée sur la corrélation, appelée *Shape-Based Distance* (SBD). Elle cherche à regrouper les séries en fonction de leur forme plutôt que de leur amplitude.

Distance SBD : La distance SBD (*Shape-Based Distance*) est une mesure de similarité invariante aux déphasages temporels. Elle s'appuie sur la corrélation croisée normalisée et est définie par :

$$d_{SB}(x, y) = 1 - \frac{\max_{\alpha} CC_{\alpha}(x, y)}{\|x\|_2 \|y\|_2}$$

où $CC_{\alpha}(x, y)$ désigne la corrélation croisée au décalage α , définie par :

$$CC_{\alpha}(x, y) = \sum_t x_{t+\alpha} y_t.$$

Cette distance mesure la similarité de forme entre deux séries, indépendamment de l'amplitude ou d'un éventuel décalage temporel, ce qui en fait un outil particulièrement adapté au clustering basé sur la structure des signaux.

Principe de l'algorithme K-Shape : L'algorithme k-Shape fonctionne de manière similaire au K-Means, mais utilise la SBD et un recalcul de centroïdes basé sur une normalisation et une optimisation par corrélation :

1. Transformation des séries multivariées en un signal univarié (via moyenne, PCA ou sélection d'une variable) ;
2. Normalisation des séries (centrage-réduction) ;
3. Initialisation des centroïdes ;
4. Assignation des séries au cluster le plus proche selon la distance SBD ;
5. Recalcul des centroïdes par *shape extraction* (méthode reposant sur SVD) ;
6. Répétition jusqu'à convergence ou jusqu'au nombre maximal d'itérations.

Implémentation et optimisation des hyperparamètres : L'algorithme K-Shape est implémenté à l'aide de la classe `KShape` du module `tslearn.clustering`. Nous avons ensuite réalisé un grid search sur les principaux hyperparamètres en évaluant les performances à l'aide d'indices externes et internes. Les meilleurs résultats ont été obtenus avec : $n_clusters = 6$, $n_init = 10$, $max_iter = 50$, $reduce_mode = "mean"$. Ce réglage montre que la simple moyenne des 9 variables donne un signal univarié suffisamment représentatif pour capturer la forme globale du mouvement.

3.3 Transformée de Fourier et K-Means

La troisième approche repose non plus sur la forme temporelle des signaux, mais sur leur contenu fréquentiel. L'idée est d'extraire des caractéristiques issues de la **Transformée de Fourier (FFT)**, puis de réduire leur dimension à l'aide d'une Analyse en Composantes Principales (PCA) avant d'appliquer un algorithme de K-Means classique dans cet espace compact et informatif.

Transformée de Fourier (FFT) : La FFT permet de représenter chaque fenêtre temporelle sous la forme d'un spectre d'amplitudes. Elle capture les fréquences dominantes du mouvement, ce qui est particulièrement pertinent pour les activités cycliquesn la montée d'escaliers. Pour chaque série nous utilisons :

- L'ajout de dérivées temporelles pour enrichir les signaux ;
- Le calcul d'un spectre d'amplitudes moyen par canal ;

- L'extraction des n premiers coefficients de Fourier (features fréquentielles).

En notant \hat{x}_k les coefficients spectraux, le vecteur de caractéristiques fréquentielles est :

$$X_{\text{freq}} = (|\hat{x}_1|, |\hat{x}_2|, \dots, |\hat{x}_n|).$$

Extraction des descripteurs temporels : En complément du spectre fréquentiel, des mesures temporelles simples sont extraites : moyennes, variances, énergies, mesures de dynamique locale. Ces descripteurs capturent l'intensité et la variabilité des signaux.

Réduction de dimension par PCA : Les caractéristiques concaténées sont ensuite normalisées, puis projetées dans un espace de faible dimension à l'aide d'une PCA. Nous utilisons une PCA conservant 98% de la variance, avec un solveur SVD adapté. Cette étape permet de :

- Réduire le bruit,
- Supprimer les corrélations redondantes,
- Accélérer considérablement l'étape de clustering.

Clustering dans l'espace réduit. Une fois les données projetées dans l'espace fréquentiel réduit, un algorithme K-Means standard est appliqué. À ce stade, la distance euclidienne est cohérente, car les descripteurs sont normalisés et orthogonalisés par la PCA.

Implémentation et optimisation des hyperparamètres : L'ensemble du pipeline (FFT + time-features + scaling + PCA + clustering) a été optimisé par un grid search portant à la fois sur les caractéristiques fréquentielles et sur les paramètres du K-Means. Les meilleurs résultats ont été obtenus avec les valeurs suivantes : $n_{\text{clusters}} = 6$, $n_{\text{init}} = 15$, $\text{max_iter} = 50$, $n_{\text{fourier_coeffs}} = 16$, $\text{PCA}_{\text{variance}} = 0.98$, $\text{svd_solver} = \text{"full"}$.

3.4 Transformée de Fourier avec HDBSCAN

La dernière approche repose sur un pipeline en trois étapes : extraction de caractéristiques dans le domaine fréquentiel, projection non linéaire dans un espace latent de faible dimension, puis clustering par densité. L'idée est de combiner des descripteurs riches avec une méthode de réduction de dimension adaptée aux structures non linéaires, avant d'appliquer un algorithme de clustering.

Principe de l'approche : Les séries normalisées sont d'abord transformées dans le domaine fréquentiel grâce à la FFT, puis enrichies par des descripteurs temporels simples. L'ensemble forme un vecteur de caractéristiques standardisé. Ces vecteurs sont ensuite projetés dans un espace latent de faible dimension à l'aide d'UMAP, qui préserve la structure locale des données. Enfin, le clustering est réalisé avec HDBSCAN, un algorithme basé sur la densité capable d'identifier automatiquement des groupes cohérents et de détecter les points considérés comme bruit.

Implémentation et optimisation des hyperparamètres : Le pipeline complet (*FFT + features temporelles + normalisation + UMAP + HDBSCAN*) a été implémenté puis optimisé à l'aide d'un grid search sur les hyperparamètres clés de l'extraction fréquentielle, de la réduction de dimension et du clustering par densité. Les meilleurs résultats ont été obtenus avec : *fourier_coeffs24*, *umap_n_components = 2*, *umap_n_neighbors = 15*, *umap_min_dist = 0.0*, *umap_metric = "manhattan"*, *hdb_min_cluster_size = 20*, *hdb_min_samples = 5*, *hdb_metric = "euclidean"*.

3.5 Résultats expérimentaux

3.5.1 Qualité de clustering

Le tableau 3.1 synthétise les performances obtenues pour chacune des approches étudiées à l'aide de plusieurs indicateurs standard du clustering :

TABLE 3.1 – Résumé des performances de clustering pour les différentes approches.

Métrique	ARI	NMI	Purity	Silhouette
DBA K-Means	0.52	0.69	0.62	0.11
K-Shape	0.62	0.58	0.72	0.22
FFT + PCA + K-Means	0.55	0.73	0.65	0.12
FFT + UMAP + HDBSCAN	0.76	0.79	0.79	0.64

Analyse : Les résultats montrent des différences nettes entre les approches. Les méthodes basées uniquement sur la forme (DBA k-means et K-Shape) offrent des performances correctes, K-Shape se distinguant légèrement grâce à une meilleure Pureté et un Silhouette supérieur. L'approche FFT + PCA + K-Means améliore la NMI mais reste globalement

comparable aux méthodes précédentes. En revanche, l'approche FFT + UMAP + HDBSCAN surpassé nettement toutes les autres selon l'ensemble des métriques, avec des gains particulièrement importants en Silhouette et en Pureté. Cette supériorité suggère que la représentation fréquentielle combinée à une réduction de dimension non linéaire produit un espace bien structuré pour un clustering basé sur la densité.

3.5.2 Visualisation des clusters

Dans cette section, nous présentons les représentations visuelles des clusters obtenus pour les différentes approches, ainsi que les matrice de confusion associées.

3.5.2.1 DBA K-Means

Les projections t-SNE basées sur les distances DTW montrent dans la figure 3.1 des clusters perceptibles mais encore fortement imbriqués. Les activités dynamiques présentent une certaine cohérence, mais les groupes restent diffus, avec des frontières floues. La matrice de confusion confirme ces observations :

- Les activités dynamiques comme Stairs Up/down sont correctement isolées ;
- Des confusions importantes apparaissent entre Sitting, Standing et Laying ;
- La marche est également mal séparée, se dispersant sur 2 clusters.

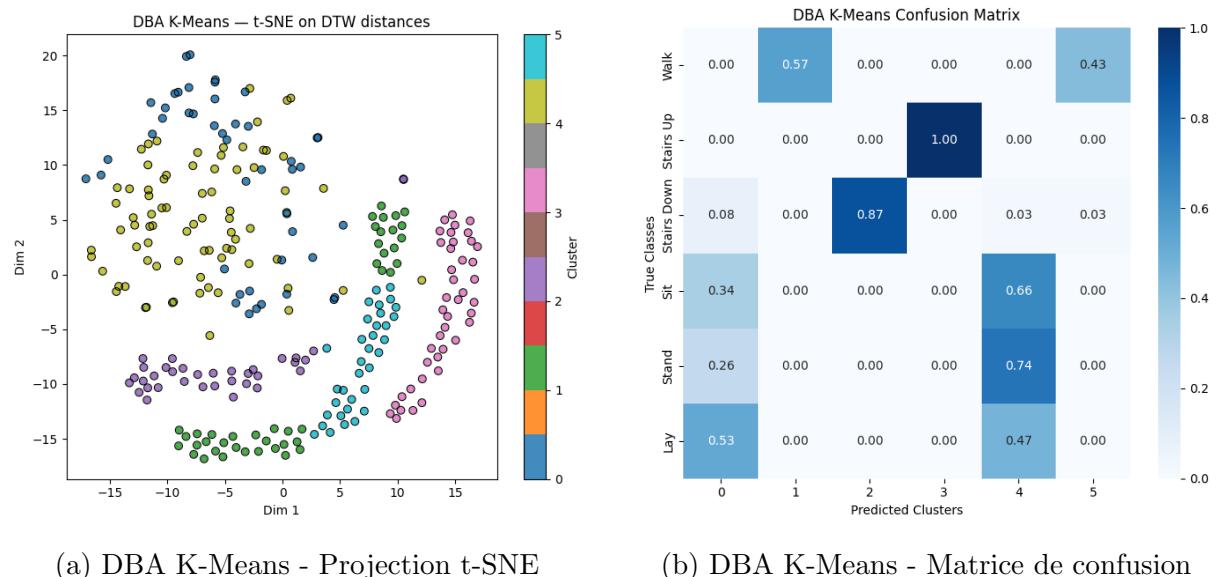


FIGURE 3.1 – Clusters et matrices de confusion (DBA K-Means).

3.5.2.2 K-Shape

Dans la figure, la projection t-SNE construite à partir des distances SBD révèle des clusters relativement structurés, notamment pour les activités dynamiques liées aux escaliers. Toutefois, une partie importante des points reste mélangée, en particulier pour les activités statiques ou faiblement différenciées (Sitting, Standing, Lying). Cela traduit une séparation imparfaite lorsque la forme du signal ne présente pas de motif clair. La matrice de confusion confirme cette observation :

- Stairs Up / Down et Walk sont bien identifiés, avec des clusters dominants nettement visibles.
- Les activités Sitting, Lying et Standing sont distribuées sur plusieurs clusters, traduisant une forte ambiguïté de forme.

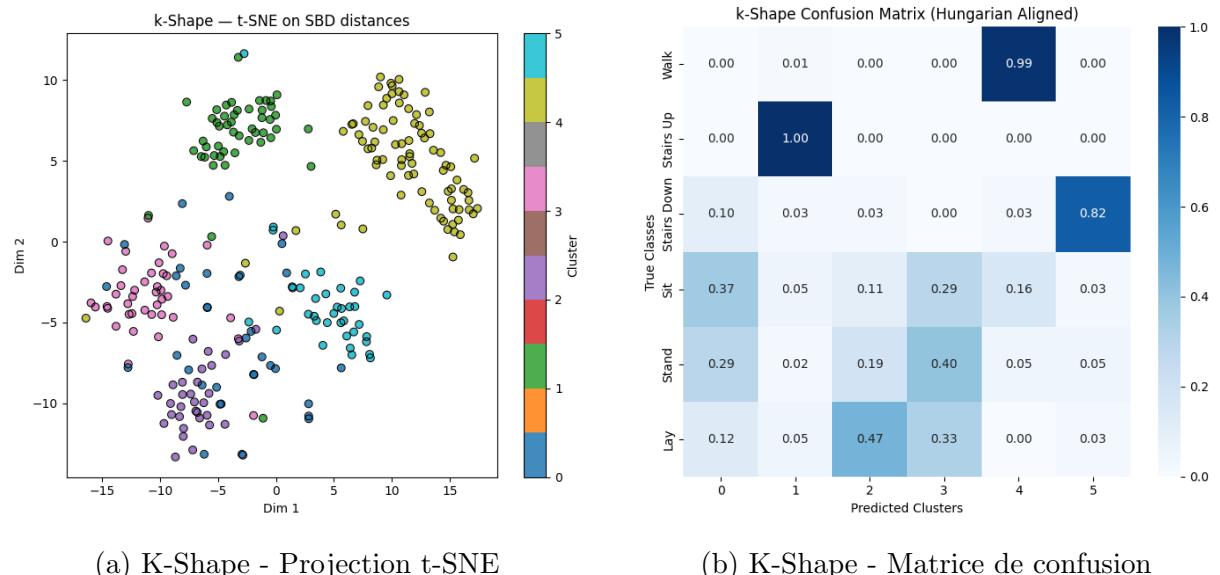
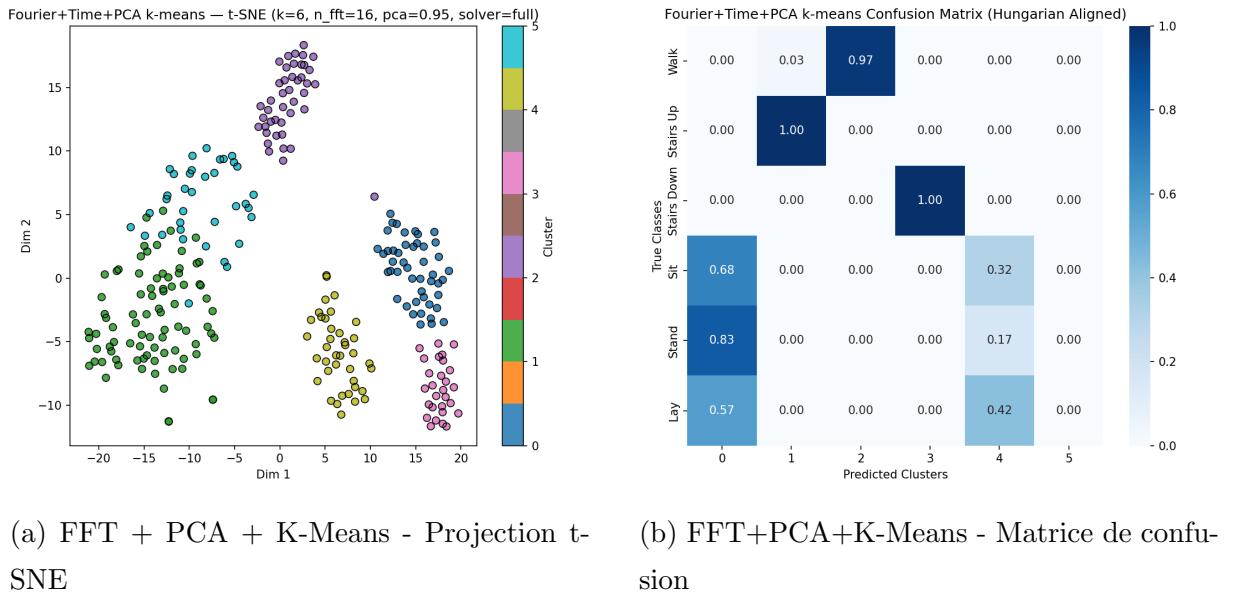


FIGURE 3.2 – Clusters et matrices de confusion (K-shape).

3.5.2.3 FFT + PCA + K-Means

La projection t-SNE des données transformées en domaine fréquentiel montre des clusters plus compacts et plus nets que ceux obtenus avec DTW ou SBD. Les caractéristiques spectrales séparent efficacement les activités rythmées telles que Walking, Stairs Up et Stairs Down, qui forment des groupes clairement délimités comme illustré dans la figure 3.3. La matrice de confusion confirme cette bonne séparation pour les activités dynamiques. En revanche, les activités statiques présentent encore des chevauchements,

leurs spectres étant naturellement très proches.



(a) FFT + PCA + K-Means - Projection t-SNE

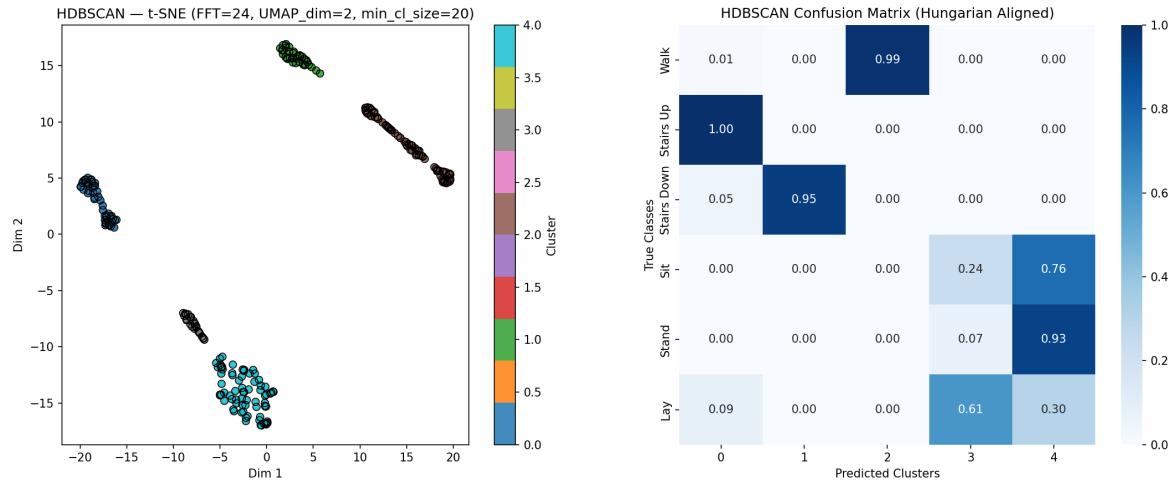
(b) FFT+PCA+K-Means - Matrice de confusion

FIGURE 3.3 – Clusters et matrices de confusion (FFT+PCA+KMEANS).

3.5.2.4 FFT + UMAP + HDBSCAN

Les projections obtenues après réduction de dimension par UMAP montrent des clusters extrêmement bien séparés, beaucoup plus nets que toutes les autres méthodes. Chaque activité forme un groupe dense et compact, sans chevauchement visible comme montré sur la figure 3.4. La matrice de confusion alignée renforce cette observation :

- Les activités dynamiques sont presque parfaitement reconnues ;
- Les activités statiques sont mieux séparées qu'avec les autres approches ;
- HDBSCAN identifie également des points de bruit, améliorant encore la structure globale.



(a) FFT + UMAP + HDBSCAN - Projection UMAP

(b) FFT+UMAP+HDBSCAN - Matrice de confusion

FIGURE 3.4 – Clusters et matrices de confusion (FFT + UMAP + HDBSCAN).

3.5.3 Synthèse

Les méthodes basées directement sur les signaux temporels (DBA K-Means, K-Shape) capturent correctement les activités dynamiques, mais montrent rapidement leurs limites dès que les signaux deviennent peu distinctifs, notamment pour les postures statiques qui présentent une très faible variabilité.

À l'inverse, les approches hybrides combinant des caractéristiques temporelles et fréquentielles, puis une réduction de dimension non linéaire, structurent bien mieux l'espace des données. Elles révèlent des clusters plus nets et obtiennent les meilleures performances, ce qui montre l'importance de transformer les séries temporelles dans un espace plus discriminant avant d'appliquer un algorithme de clustering.

Conclusion

L’objectif du projet était de répondre à la problématique suivante : *comment regrouper des séries temporelles multivariées issues de capteurs inertIELS en différentes activités humaines, en l’absence de labels, tout en tenant compte des décalages temporels, du bruit et de la variabilité du mouvement ?* Les différentes expérimentations ont permis de montrer que :

- Les méthodes fondées directement sur les séries temporelles (DTW, SBD) sont efficaces pour des activités dynamiques mais restent limitées pour des postures plus similaires ;
- La combinaison de caractéristiques temporelles et fréquentielles, suivie d’une réduction de dimension et d’un algorithme de clustering classique, permet de structurer un espace latent beaucoup plus discriminant ;
- Cette approche hybride offre les meilleurs résultats et répond de manière satisfaisante à la problématique initiale.

Ainsi, le rapport met en évidence que l’efficacité du clustering de séries temporelles repose autant sur la représentation des données que sur le choix de l’algorithme, et que des techniques avancées de transformation et de réduction dimensionnelle sont indispensables pour obtenir des groupes cohérents dans des contextes réels et bruités.