



## Rapport explicatif du projet Data Intelligence

### **Phase 3 :**

Dans cette phase 3 du projet Data Intelligence, notre objectif est d'évaluer la performance de l'algorithme précédemment développé pour détecter le type de mouvement effectué par une personne dès la première minute, en utilisant les données de test. Pour atteindre cet objectif, nous avons mis en place une série de fonctions, chacune jouant un rôle crucial dans le processus d'évaluation. Ces fonctions traitent les données, calculent des mesures de performance et affichent les statistiques des tests. Voici un aperçu des choix que nous avons faits dans cette phase et des fonctions implémentées :

#### `convertFiletoTablePattern`

Cette fonction lit un fichier CSV contenant des modèles de déplacement et les convertit en une table de structures de modèle. Chaque ligne du fichier correspond à un modèle de déplacement, avec le premier élément étant le numéro du mouvement et les éléments suivants étant les valeurs d'accélération.

Elle utilise la fonction `strtok` pour séparer les éléments de chaque ligne à l'aide de virgules comme délimiteurs.

Les valeurs numériques sont extraites à l'aide de la fonction `sscanf` et stockées dans les champs appropriés de la structure de modèle.

#### `convertFiletoTableTestSet`

Cette fonction lit un fichier CSV contenant un ensemble de tests et les convertit en une table de structures de modèle. Chaque ligne du fichier correspond à un test, avec le premier élément étant le numéro du mouvement, le deuxième élément étant le numéro de sous-mouvement, et les éléments suivants étant les valeurs d'accélération.

Comme dans la fonction précédente, elle utilise `strtok` et `sscanf` pour extraire les données du fichier et les stocker dans les structures de modèle.

#### `euclideanDistance`

Cette fonction calcule la distance euclidienne entre deux vecteurs de doubles de même longueur.

Elle itère sur chaque élément des vecteurs, calcule la différence carrée entre les éléments correspondants, et ajoute les carrés des différences.

La racine carrée de la somme des carrés des différences est ensuite retournée comme la distance euclidienne.

#### `findClosestModel`

Cette fonction trouve le modèle le plus proche dans une table de modèles, en utilisant la distance euclidienne pondérée par les tolérances de chaque dimension.

Elle parcourt tous les modèles, calcule la distance entre chaque modèle et les valeurs de test, et garde trace de la distance minimale.

Le modèle avec la distance minimale est retourné comme le modèle le plus proche.

### rateSuccess

Cette fonction évalue le taux de succès d'une série de tests par rapport à des modèles de référence.

Elle utilise la fonction `findClosestModel` pour attribuer à chaque test un modèle de référence le plus proche, puis compare le mouvement réel du test avec celui estimé.

Les résultats sont stockés dans une structure de résultats, contenant le nombre total de succès et d'échecs pour chaque mouvement.

### toleranceRate

Cette fonction calcule le taux d'erreur en considérant les mouvements réels et estimés.

Elle compte le nombre de mouvements incorrects et le divise par le nombre total de tests pour obtenir le taux d'erreur, exprimé en pourcentage.

### filter

Cette fonction cherche à trouver les meilleures tolérances pour filtrer les modèles de test en fonction des modèles de référence.

Elle explore un espace de tolérance défini par des bornes et un pas, en essayant différentes combinaisons de tolérances pour maximiser le taux de succès.

Les meilleures tolérances sont ensuite stockées dans la variable globale `bestTolerance`.

### displayStats

Cette fonction affiche les statistiques des tests, y compris le nombre de succès et d'échecs pour chaque mouvement, le pourcentage de succès, le nombre d'attractions incorrectes pour chaque mouvement, et le taux de succès global.

Chaque fonction joue un rôle spécifique et essentiel dans le processus d'évaluation de notre algorithme de détection de mouvement. De la lecture des données à partir de fichiers CSV à l'affichage des statistiques des tests, ces fonctions contribuent à une évaluation approfondie de la performance de notre algorithme. En utilisant ces fonctions, nous sommes en mesure d'analyser les résultats de manière efficace et de prendre des décisions éclairées pour améliorer notre système de détection de mouvement.