Etude personnelle de données temporelles d'un acccéléromètre

Pierre Berjon

February 4, 2021

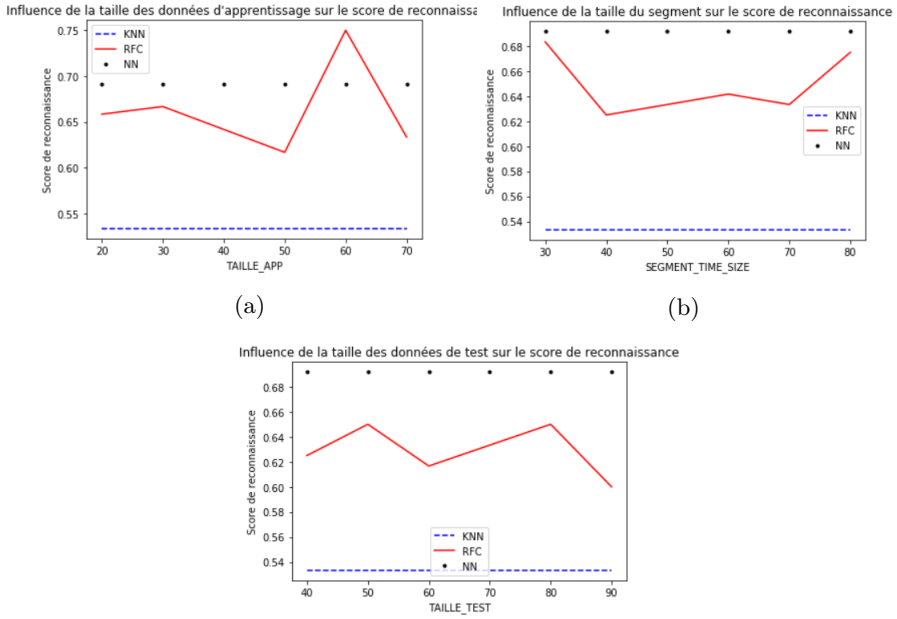
1 Influence des dimensionnements des données sur le score de reconnaissance

Tout d'abord, nous souhaitons étudier l'évolution du score de reconnaissance en fonction des paramètres de dimensionnement des données. Ici, nous allons faire varier trois paramètres :

- **la taille du segment**
- **la taille du jeu de données d'apprentissage**
- **la taille du jeu de test**

Dans un premier temps, nous devons pré-traiter les données reçues en les séparant en données d'apprentissage et en données de test. Ensuite, nous déterminerons quelles sont les valeurs optimales pour ces paramètres, que nous utiliserons ensuite. Nous avons également pris le parti d'effectuer une ACP pour cette étude. Les modèles utilisés seront des KNN, RFC et MLP.

Pour cela, nous avons défini des fonctions de 'score' permettant de déterminer le score de reconnaissance associé à un modèle particulier. Ainsi, nous n'avons ensuite qu'à faire varier les paramètres cités plus haut et déterminer les score de reconnaissance de chaque modèle en fonction :



(a) (b)

(c)

Figure 1: Evolution du score de reconnaissance en fonction des différents paramètres de dimensionnement

Ici, nous en arrivons au stade où nous pouvons observer l'influence des paramètres sur l'efficacité de nos modèles.

On voit bien ici que tous ces paramètres n'ont aucun impact sur le score de reconnaissance pour KNN et le réseau de neurones utilisé. Cependant, ils en tous une sur le classifieur Random Forest. Dans ces conditions, on constate que nous avons un intérêt certain à augmenter chacun de ces paramètres.

Cependant, on ne peut pas représenter tenir compte graphiquement des valeurs optimales pour les trois paramètres, car nous les obtiendrions en faisant varier les 3 paramètres en même temps.

Les valeurs optimales obtenues sont alors :

- **taille du segment = 30**
- **taille du jeu d'apprentissage = 80**
- **taille du jeu de test = 40**

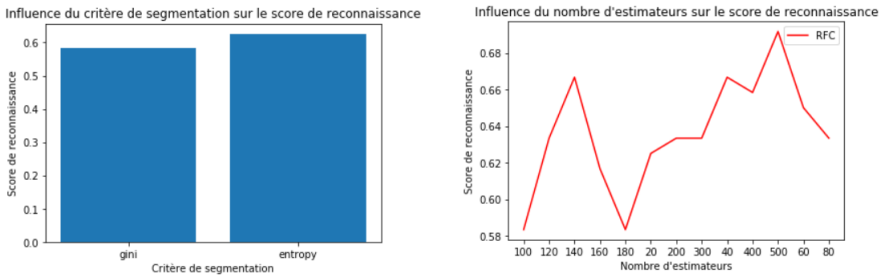
Pour ces valeurs, le score de reconnaissance atteint 0.816.

2 Analyse détaillée des différents modèles utilisés

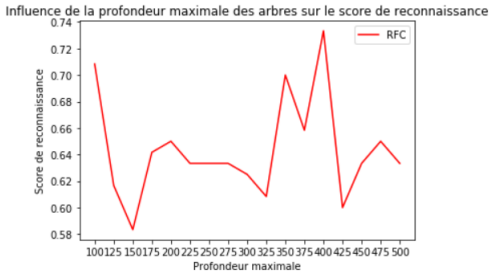
Maintenant, nous allons tenter d'optimiser les différents modèles utilisés et de distinguer le plus efficace (avec les valeurs précédemment déterminées).

2.1 Random Forest Classifiers

Nous limiterons ici notre étude sur le **nombre d'estimateurs**, le **critère de segmentation** à utiliser, et la **profondeur maximale** de l'arbre de décision.



(a) (b)



(c)

Figure 2: Evolution du score de reconnaissance en fonction des paramètres du RFC

De la même façon que pour la section précédente, on peut déterminer des valeurs optimales pour un paramètre précis (en gardant les autres paramètres par défaut) qui optimiseraient le score de reconnaissance de notre modèle. Ici, ce serait :

- **nombre d'estimateurs = 140**
- **critère de segmentation = Gini**
- **profondeur maximale = 125**

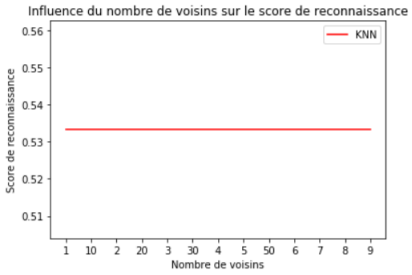
Cependant, on effectue une analyse conjointe et on obtient les valeurs suivantes :

- **nombre d'estimateurs = 140**
- **critère de segmentation = Entropie**
- **profondeur maximale = 325**

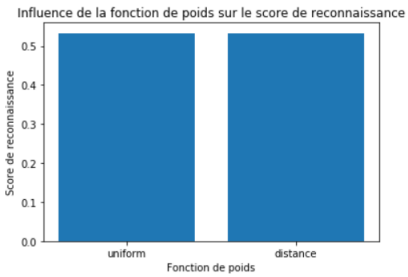
Avec ces valeurs, on obtient un score de reconnaissance de 0.825. On a bien amélioré le modèle, ce sont ces valeurs que nous retiendrons.

2.2 K-Nearest Neighbors

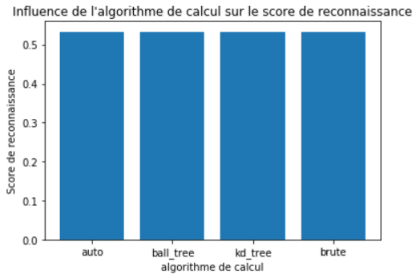
Nous limiterons ici notre étude sur le **nombre de voisins**, la **fonction de poids** à utiliser, et l'**algorithme** utilisé pour calculer les voisins les plus proches.



(a)



(b)



(c)

Figure 3: Evolution du score de reconnaissance en fonction des paramètres du KNN

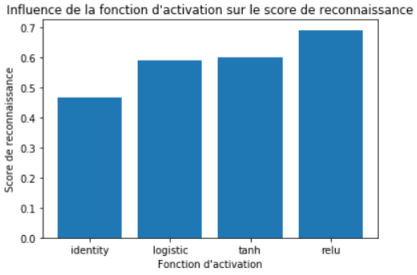
Comme précédemment, on effectue une analyse conjointe et on obtient les résultats suivants :

- **nombre de voisins = 50**
- **fonction de poids = 'distance'**
- **algorithme de calcul = 'brute'**

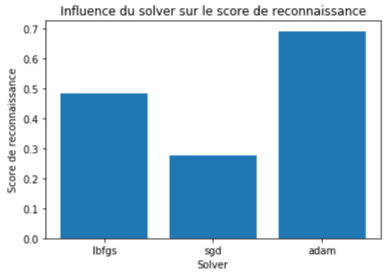
Avec ces valeurs, on obtient un score de reconnaissance de 0.566. L'amélioration du modèle n'est pas très significative.

2.3 MLP Classifier

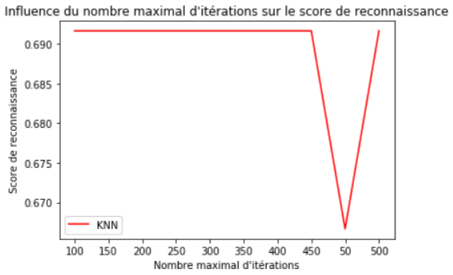
Nous limiterons ici notre étude sur la **fonction d'activation**, le **solver** et le **nombre maximum d'itérations**.



(a)



(b)



(c)

Figure 4: Evolution du score de reconnaissance en fonction des paramètres du MLP

Pour un paramètre précis (en gardant les autres paramètres par défaut), les valeurs qui optimiseraient le score de reconnaissance de notre modèle sont :

- **activation = 'logistic'**
- **solver = 'adam'**
- **nombre maximal d'itérations = 50**

Contrairement à précédemment, on effectue une analyse conjointe en obtenant les mêmes résultats.

Avec ces valeurs, on obtient un score de reconnaissance de 0.842. On a encore amélioré le modèle, celui-ci étant plus efficace que le modèle de Random Forest. Ce sont donc ces valeurs que nous retiendrons.

3 Ajout de nouvelles classes

Dans cette section, nous allons reprendre les versions les plus efficaces des modèles testés auparavant, puis allons étudier l'impact que l'ajout de nouvelles classes a sur leur score de reconnaissance. **Attention**, le modèle déterminé ici ne sera pas forcément optimal pour ces nouvelles classes, car on reprend les paramètres obtenus en Section 1, qui le sont pour une étude à 3 classes. L'idée ici est uniquement de déterminer l'impact qu'a ce changement sur les modèles, pas de déterminer le modèle optimal pour chaque nombre de classes.

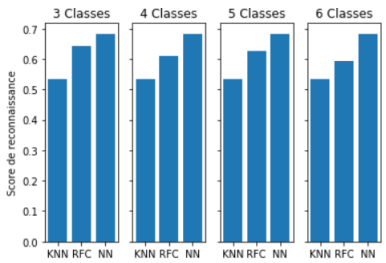


Figure 5: Evolution du score de reconnaissance des modèles pour l'ajout de nouvelles classes

On voit bien ici que le nombre de classes a un faible impact sur le score de reconnaissance des modèles utilisés, avec cependant de meilleurs résultats pour 3 classes.

En définitive, il semblerait que l'on doive privilégier 3 classes pour maximiser le score de reconnaissance. C'est assez pratique étant donné que toute l'analyse précédente (partie 2) a été effectuée avec 3 classes. Ainsi, à priori, on peut dire que pour maximiser le score de reconnaissance, on devrait utiliser un RFC avec les paramètres obtenus dans la section 2.1.

4 Reconnaissance d'utilisateurs

Enfin, au lieu de détecter des différents modes de mouvements, nous allons essayer de détecter les utilisateurs.

On va donc reprendre la même analyse, mais cette fois sur de la reconnaissance d'utilisateurs. Pour effectuer celle-ci, nous devons dimensionner les données de tel sorte à avoir une classification fiable à la reconnaissance d'utilisateurs, et ensuite enchainer par une variation des différents paramètres. On effectuera à chaque fois une analyse globale permettant d'obtenir des paramètres optimaux.

4.1 Dimensionnement de la donnée

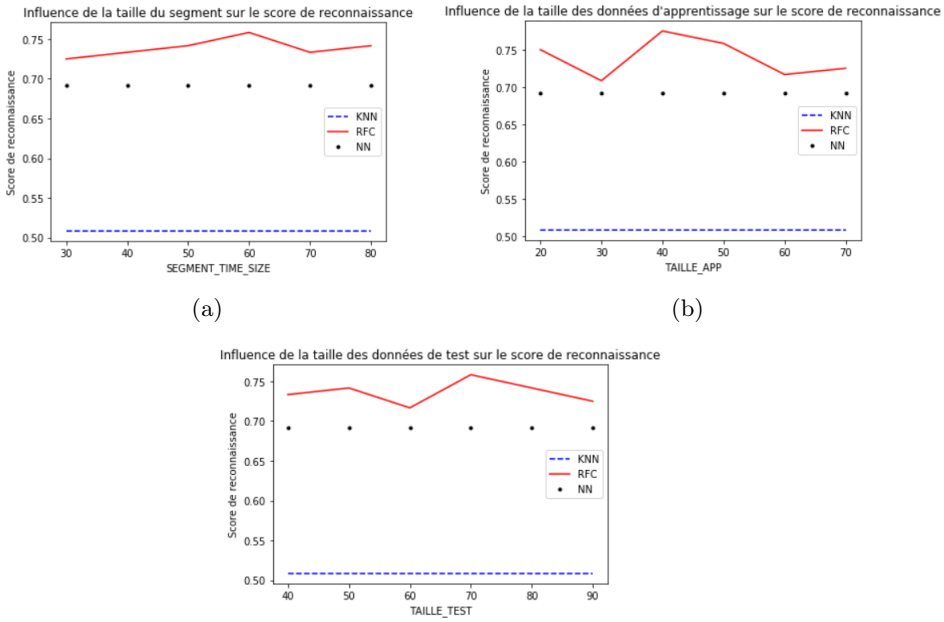


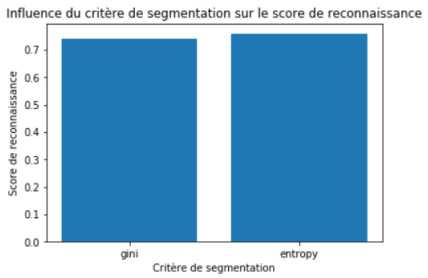
Figure 6: Evolution du score de reconnaissance en fonction des différents paramètres de dimensionnement pour la reconaissance d'utilisateurs

Analyse Globale :

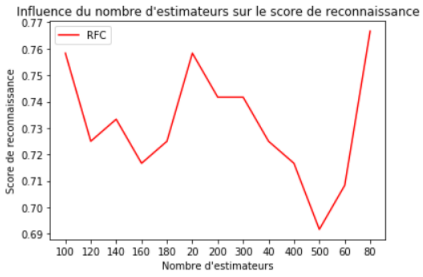
- Taille du Segment = 50
- Taille du jeu de Test = 80
- Taille du jeu d'Apprentissage = 40

SCORE OBTENU = 0.77

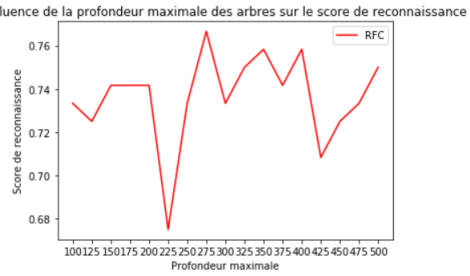
4.2 Random Forest Classifier



(a)



(b)



(c)

Figure 7: Evolution du score de reconnaissance en fonction des paramètres du RFC pour la reconnaissance d'utilisateurs

Analyse Globale :

- Nombre d'estimateurs = 160
- Critère de segmentation = Gini
- Profondeur maximale = 350

SCORE OBTENU = 0.81

4.3 K-Nearest Neighbors

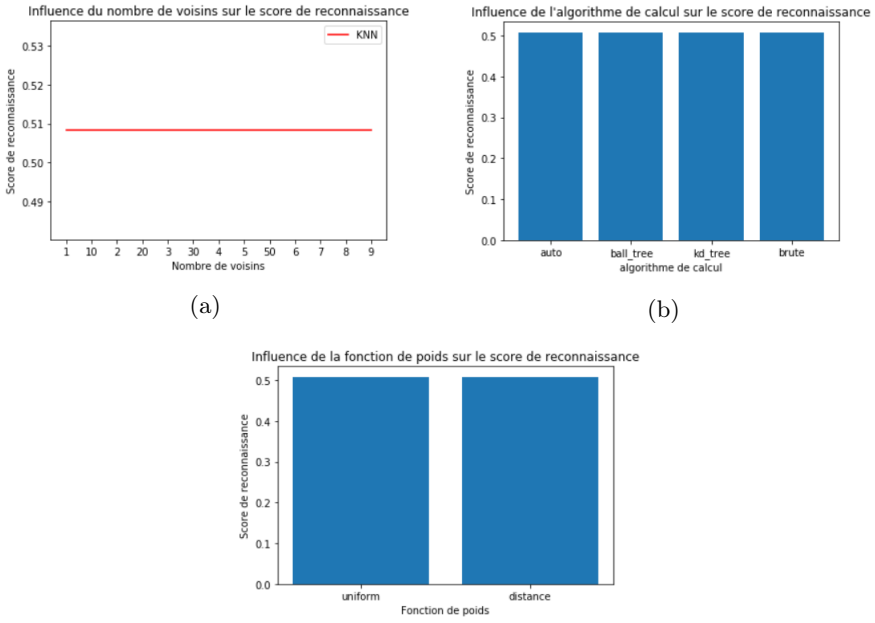


Figure 8: Evolution du score de reconnaissance en fonction des paramètres du KNN pour la reconnaissance d'utilisateurs

Analyse Globale :

- Nombre de voisins = 50
- Fonction de poids = Distance
- Algorithme = Brute

SCORE OBTENU = 0.51

4.4 Multi-Layer Perceptron

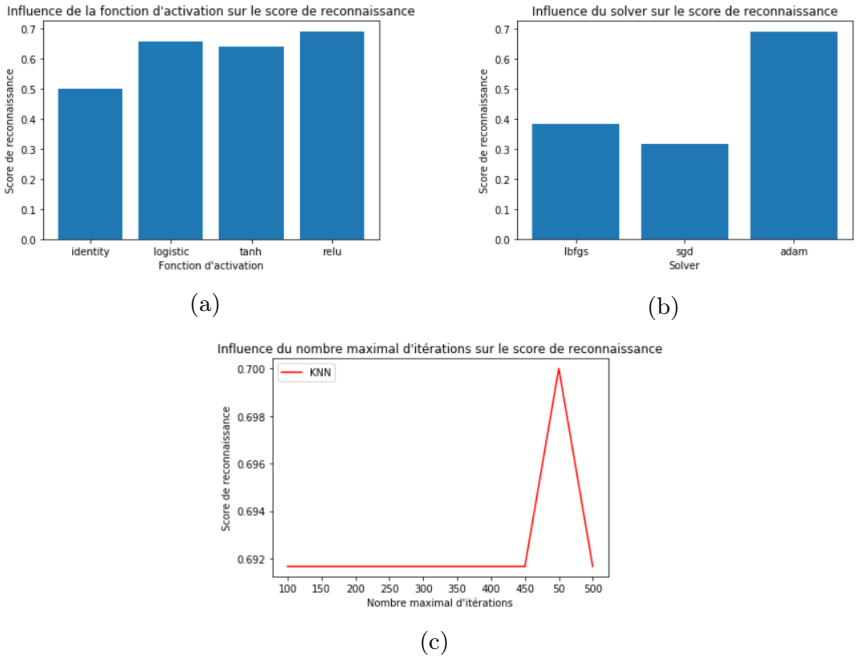


Figure 9: Evolution du score de reconnaissance en fonction des paramètres du MLP pour la reconnaissance d'utilisateurs

Analyse Globale :

- Fonction d'activation = ReLu
- Solver = Adam
- Nombre maximal d'itérations = 50

SCORE OBTENU = 0.70

5 Conclusion

En définitive, on obtient (avec les paramètres étudiés plus haut), un score de reconnaissance de **0.84 pour la détection d'activité** et de **0.81 pour la détection d'utilisateurs**.