

Compte rendu 2

Stage Laura Parisot

Détection du changement d'étage



Laura Parisot
21/05/2024



Sommaire

01

Etat de l'art

02

La sélection des
caractéristiques

03

Collecte des données

04

Rendu ENSG
rapport
installation

05

La suite ...

01

Etat de l'art



Etat de l'art:

(PDF) Feature Selection for Floor-changing Activity Recognition in Multi-Floor Pedestrian Navigation (researchgate.net)

2014 Seventh International Conference on Mobile Computing and Ubiquitous Networking (ICMU)

Feature Selection for Floor-changing Activity Recognition in Multi-Floor Pedestrian Navigation

Sara Khalifa^{*1}, Mahbub Hassan^{*1}, and Aruna Seneviratne^{†1}

^{*1}School of Computer Science and Engineering, University of New South Wales, Sydney, NSW 2052, Australia
Email: sarak, mabhba@csse.unsw.edu.au

^{†1}School of Electrical and Telecommunications Engineering, University of New South Wales, Sydney, NSW 2052, Australia
Email: aruna.seneviratne@unsw.edu.au

^{‡1}National ICT Australia, Locked Bag 9013, Alexandria, NSW 1435, Australia
Email: sara.khalifa, mahbub.hassan, aruna.seneviratne@nicta.com.au

Abstract—In large shopping malls and airports, pedestrians often change floors using conveniently located lifts and escalators. Floor changing activity recognition (FCAR) therefore needs to be a vital aid to pedestrian navigation systems. The main objective of this paper is to achieve accurate FCAR with the minimal number of features and experimental data. To complete the performance of various feature selection methods, we compare the performance of a MLP classifier trained on a dataset of features extracted from a smartphone's accelerometer. The MLP classifier is trained to detect whether the user is using an escalator or a lift. The results show that an accelerometer embedded in a smartphone can achieve 94% recognition accuracy using only 5 features.

I. INTRODUCTION

In multi-floor pedestrian navigation systems, such as pedestrian floor decking (PFD) implemented in a subway station [1], it is common for a pedestrian to move between locations in a new floor plus whenever the user moves from one floor to another. Since pedestrians often change floors using conveniently located lifts and escalators, floor changing activity recognition (FCAR) as riding an escalator or a lift, is considered a vital component of these emerging personal indoor navigation systems [2].

Activity recognition involves computation of a set of features (feature extraction) from smartphone sensors and feeding them to a pre-trained classifier that matches the input features to one of the known output classes or activities. For example, an FCAR module may consider three accelerometer features, such as the average accelerations in x , y , and z directions in the last 3 seconds, to detect whether a pedestrian is riding an escalator or a lift using a multi-layer perceptron (MLP) classifier, which is trained with a large amount of accelerometer data collected from non-pedestrians.

The number of features used for activity recognition directly contributes to the computational complexity (or overhead) of the overall FCAR process. First, a higher number of features means the FCAR will need to extract (compute) more information from the accelerometer each time it needs to make a decision. Second, the FCAR will need to process more data in seconds in some practical systems [3]. Second, a classifier trained to work with a larger set of inputs or features is expected to have a larger computational requirement. For example, a three-layer MLP executes a total of $(N_i + N_o)^2 + 1.5(N_i + 3N_o)$ additions, multiplications, and function evaluation operations, where N_o represents the number of output classes, and N_i the number of input features (assuming that it employs $\frac{128 \times 256}{128 \times 128}$ neurons in its hidden layers) [3]. Therefore, given the same number of activities, the number of operations required to be completed by the MLP classifier increases quadratically with the number of features. These observations motivate us to investigate feature selection that leads to the minimal number of features for FCAR without losing detection accuracy.

To carry out our investigation, we collected accelerometer data from several volunteers riding escalators and lifts in different buildings. We then applied a number of different feature selection methods and classified to this data and evaluated their performance in terms of classification accuracy and the number of features. The three floor-changing activities considered are riding an escalator (E) and a lift (L) in a standing position, which initially gives two output classes for the classifier. However, because standing still anywhere in a building is not possible, we also consider walking in a standing position. This gives three output classes for the classifier. We consider three activities in our studies. The results show that an accelerometer embedded in a smartphone can achieve 94% recognition accuracy using only 5 features.

II. RELATED WORK

FCAR is a part of the Human Activity Recognition (HAR). Research on HAR has been an area of significant research in the literature over the past years. All the approaches for HAR share 3 basic components: data collection, feature extraction, and classification. In the data collection phase, most of the published work relied on attaching accelerometer sensors to the subjects' body and collecting data from those sensors. However, the popularity of smartphones in the recent years has shifted the focus to smartphone-based HAR. Table I summarizes some of the studies related to performing HAR using either wearable sensors or smartphones, in terms of the

978-1-4799-2231-4/14/\$1.00 ©2014 IEEE

Etat de l'art:

(PDF) Feature Selection for Floor-changing Activity Recognition in Multi-Floor Pedestrian Navigation (researchgate.net)

2014 Seventh International Conference on Mobile Computing and Ubiquitous Networking (ICMU)

Feature Selection for Floor-changing Activity Recognition in Multi-Floor Pedestrian Navigation

Sara Khalifa^{1*}, Malibah Hassan¹, and Aruna Seneviratne^{1,2}

¹School of Computer Science and Engineering, University of New South Wales, Sydney, NSW 2052, Australia

E-mail: sarak, malibah@cs.unsw.edu.au

¹School of Electrical and Telecommunication Engineering, University of New South Wales, Sydney, NSW 2052, Australia

E-mail: a.seneviratne@unsw.edu.au

^{1,2}National ICT Australia, Locked Bag 9013, Alexandria, NSW 1435, Australia

E-mail: sara.khalifa, malibah.hassan

Abstract— In large, shopping malls, and airports, pedestrian often changes floor using conveniently located lifts and escalators. **Floor changing pedestrian recognition (FCAR)** therefore can be a vital aid to multi-floor pedestrian navigation systems. The focus of this paper is to propose a feature selection method to reduce the number of features. Using experimental data, we compare the performance of various feature selection methods and classifiers trained using the selected features to recognize a pedestrian using a lift. The results show that an accelerometer embedded in a smartphone can achieve 94% recognition accuracy using only 5 features.

I. INTRODUCTION

In multi-floor pedestrian navigation systems, such as pedestrian dead reckoning (PDR) implemented in a smartphone [1], it is necessary to reset the pointer to a new starting location in a new floor plan whenever the user moves from one floor to another. Since pedestrians often change floors using lifts or escalators, floor changing pedestrian activity recognition (FCAR), such as riding an escalator or a lift (L), is considered a critical component of these emerging personal indoor navigation systems [2].

Activity recognition requires specific computation of a set of features (feature extraction from smartphone sensors and feeding them to a pre-trained classifier that matches the input features to one of the known output classes or activities). For example, an FCAR module needs three accelerometers to detect features such as the x, y, and z accelerations; x , y , and z directions in the last 3 seconds, to detect whether the pedestrian is riding an escalator or a lift using a multi-layer perceptron (MLP) classifier, which was trained *a priori* with a large amount of accelerometer data collected from many pedestrians.

The number of features used for activity recognition directly contributes to the computational complexity (or overhead) of the overall FCAR process. First, a higher number of features means the FCAR module will extract more information from the sensor data. Second, if the time it takes to make a decision about the current activity, which could be every 2.4 seconds in some practical systems [2]. Second, a classifier trained to work with a larger set of inputs or features is expected to have a larger computational requirement. For example, a three-layer MLP requires a total of $(N_1 + N_2) + 1.5(N_1 + N_2)$

additions, multiplication, and function evaluation operations, where N_1 represent the number of output classes, and N_2 the number of input features (assuming that it employs $\frac{(N_1+N_2)}{2}$ neurons in its hidden layer) [3]. Therefore, given a target set of activities, the number of operations to be completed by the MLP classifier increases quadratically with the number of features. Using the observation, we propose to investigate feature selection that leads to the minimal number of features for FCAR without sacrificing the detection accuracy.

To carry out our investigations, we collected accelerometer data from 10 volunteers walking in different locations in different buildings. We then applied a number of different feature selection methods and classifiers to this data and compared their performances in terms of detection accuracy and the number of features. The main floor-changing activities considered are riding an escalator (E) and a lift (L) in a multi-floor building. The floor changing pedestrian activity recognition (FCAR), such as riding an escalator or a lift, is considered a critical component of these emerging personal indoor navigation systems [2].

Activity recognition requires specific computation of a

set of features (feature extraction from smartphone sensors and feeding them to a pre-trained classifier that matches the input features to one of the known output classes or activities). For example, an FCAR module needs three accelerometers to detect features such as the x, y, and z accelerations; x , y , and z directions in the last 3 seconds, to detect whether the pedestrian is riding an escalator or a lift using a multi-layer perceptron (MLP) classifier, which was trained *a priori* with a large amount of accelerometer data collected from many pedestrians.

The number of features used for activity recognition directly contributes to the computational complexity (or overhead) of the overall FCAR process. First, a higher number of features means the FCAR module will extract more information from the sensor data. Second, if the time it takes to make a decision about the current activity, which could be every 2.4 seconds in some practical systems [2]. Second, a classifier trained to work with a larger set of inputs or features is expected to have a larger computational requirement. For example, a three-layer MLP requires a total of $(N_1 + N_2) + 1.5(N_1 + N_2)$

Feature Selector :

IG Information Gain CLS Correlation Decision Tree Puning

TABLE II
THE INITIAL FEATURE SET BEFORE APPLYING FEATURE SELECTION (7 BASIC FEATURES YIELDING A TOTAL OF 19 EXTRACTED FEATURES)

Feature Name	Computational Formula
Mean of the three axes	$\mu(x) = \frac{1}{n} \sum_{i=1}^n x_i$
Standard deviation (Std) of the three axes	$\sigma(x) = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2}$
Skewness of the three axes	$Skew(x) = \frac{\frac{1}{n} \sum_{i=1}^n (x_i - \mu)^3}{\sigma^3}$
Kurtosis of the three axes	$Kurt(x) = \frac{\frac{1}{n} \sum_{i=1}^n (x_i - \mu)^4}{\sigma^4} - 3$
Average Absolute Deviation of the three axes	$AAD(x) = \frac{1}{n} \sum_{i=1}^n x_i - \mu $
Pairwise-Correlation of the three axes	$Corr(x, y) = \frac{Cov(x, y)}{\sigma(x)\sigma(y)}$
Average Resultant Acceleration	$ARA(x, y, z) = \frac{1}{n} \sum_{i=1}^n \sqrt{x_i^2 + y_i^2 + z_i^2}$

The "minus 3" in the kurtosis equation, is often used as a correction to make the kurtosis of the normal distribution equal to zero.

TABLE III
DEFAULT WEKA CONFIGURATIONS FOR THE FIVE CLASSIFIERS.

Classifier	Configuration
DTL	Evaluation Measure: Accuracy, Search Algorithm: BestFirst.
DT (J48)	Confidence Factor = 0.25, The minimum number of instances per leaf = 2.
NB	A default precision of 0.1 for numeric attributes.
KNN	K = 1, No distance weighting, Nearest Neighbour Search Algorithm: LinearNNSearch (Euclidean Distance).
MLP	Hidden Layers = (Features+classes)/2, Learning Rate = 0.3, Momentum = 0.2, Training time = 500 epochs.

02

La sélection des caractéristiques



NE PAS EFFACER

merci

- ① Nb échantillon :
- + 100 ↔ 80
 - + 200 ↔ 240
 - + 300 ↔ 320
 - + 400 ↔ 400
 - + 500 ↔ 400
- frequencce
échantillonnage
capteur

- ② % recouvrement entre échantillons
- + (None)
 - + (5 échantillons)
 - + 50% ($\frac{1}{2}$)
 - + 75% ($\frac{3}{4}$)
 - + 25% ($\frac{1}{4}$)

- ③ Filtrage pour denoising + Nome
- + Mean Filter
 - + Gaußien Filter
 - + Kalman Filter
 - + Wavelet Filter
 - + 5-Spot triple Smoothing algorithm

④ Caractéristiques Temporales

↳ fait sur vector normalisé
vector/mom(vecteur)

- + Mean
- + Std
- + Pairwise Correlation
- + Skewness
- + Kurtosis
- + SMA
- + AAD (Absolute Deviation)
- + ARC (Average Reciprocal Correlation)

⑤ Caractéristiques Fréquentielles

- + Nome
- + Index 1st Peak FFT
- + Index 2nd Peak FFT
- + Value 1st Peak FFT
- + Value 2nd Peak FFT
- + Spectrum Entropy
- + Cepstral Energy

⑥ Sélection des caractéristiques

- ? → Nombre de caractéristiques minimum ?
- + Nome
 - + IG3 seuil? \Rightarrow 0.5
 - + CLS3 seuil $> 0.1 \Rightarrow$ corrélations entre mode et la feature
 - + DTF3 importance > 0.05

⑦ Classifieur

- + MLP
- + Naïves Bayes
- + Decision Tree
- + SVM
- + XGBoost
- + KNN

⑧ Comment décider le meilleur

- + Meilleur accuracy
- + Temps d'exécution
- + Poids du modèle
- + Moins de mémoire requise

① Nb échantillon :

80 Hz
fréquence
échantillonnage
capteur

+1s \leftrightarrow 80
+2s \leftrightarrow 160
+3s \leftrightarrow 240
+4s \leftrightarrow 320
+5s \leftrightarrow 400

↳

② % recouvrement
entre échantillons

+ (None)

+ (5 échantillons)

+ 50% ($\frac{\text{nb}}{2}$)

+ 75% ($\frac{\text{nb}}{4}$)

+ 25% ($\frac{\text{nb} \times 3}{4}$)

③ Filtrage

③

Filtrage
pour denoising + Noise

+ 25% (nb: $\frac{3}{4}$)

(5)

+ Mean Filter

+ Gaussian Filter

+ Kalman Filter

+ Wavelet Filter

+ 5-Spot triple Smoothing
algorithm

+

+

④ Caractéristiques Temporelles

↳ fait sur vector normalisé

vector / norm(vector)

- + Mean
- + Std
- + Pairwise Correlation
- + Skewness
- + Kurtosis
- + SNA
- + AAD (Average Absolute Deviation)
- + ARC (Average Resultant Correlation)

⑤ Caractéristiques fréquentielles

- + Nome
- + Index 1^{er} Peak FFT
- + Index 2nd Peak FFT
- + Value 1^{er} Peak FFT
- + Value 2nd Peak FFT
- + Spectrum Entropy
- + Andebelle Energy

⑥

Selection
des caractéristiques

} → Nombre
de caractéristiques
minimum ?

- + None
- + IIG } seuil? $\nearrow 0.5$
- (cr) + CLS } seuil $> 0.1 \Rightarrow$ correlation
entre mode
- + DTP } importance
 > 0.05 et la feature

⑦ Classifieur

+ MLP

+ Decision Tree

+ XGBoost

+ Naives Baye

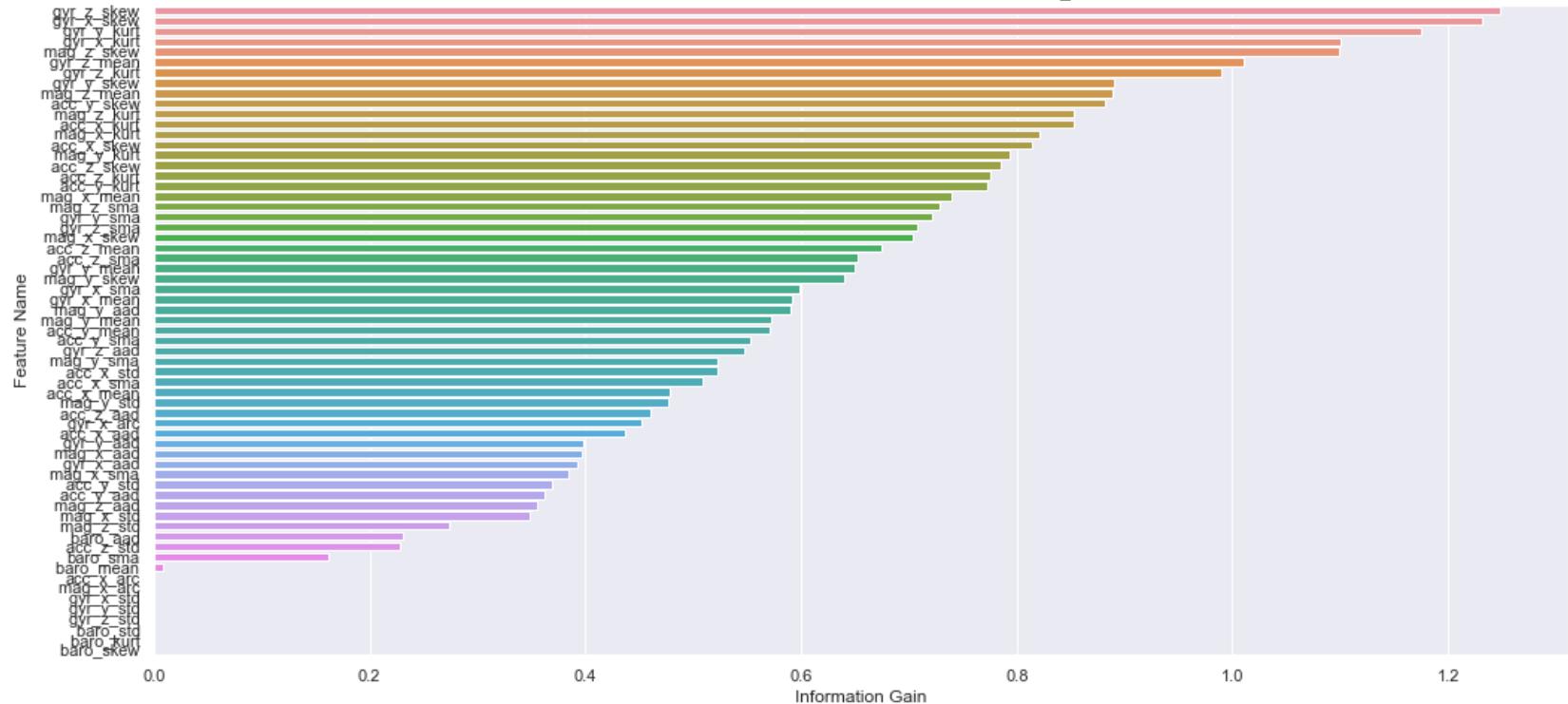
+ SVM

+ KNN

⑧ Comment décider le meilleur

- + Meilleur accuracy
- + Temps d'exécution
- + Poids du modèle
- + Moins de mémoire requise

Information Gain of Features 80 échantillons 50 window_slide



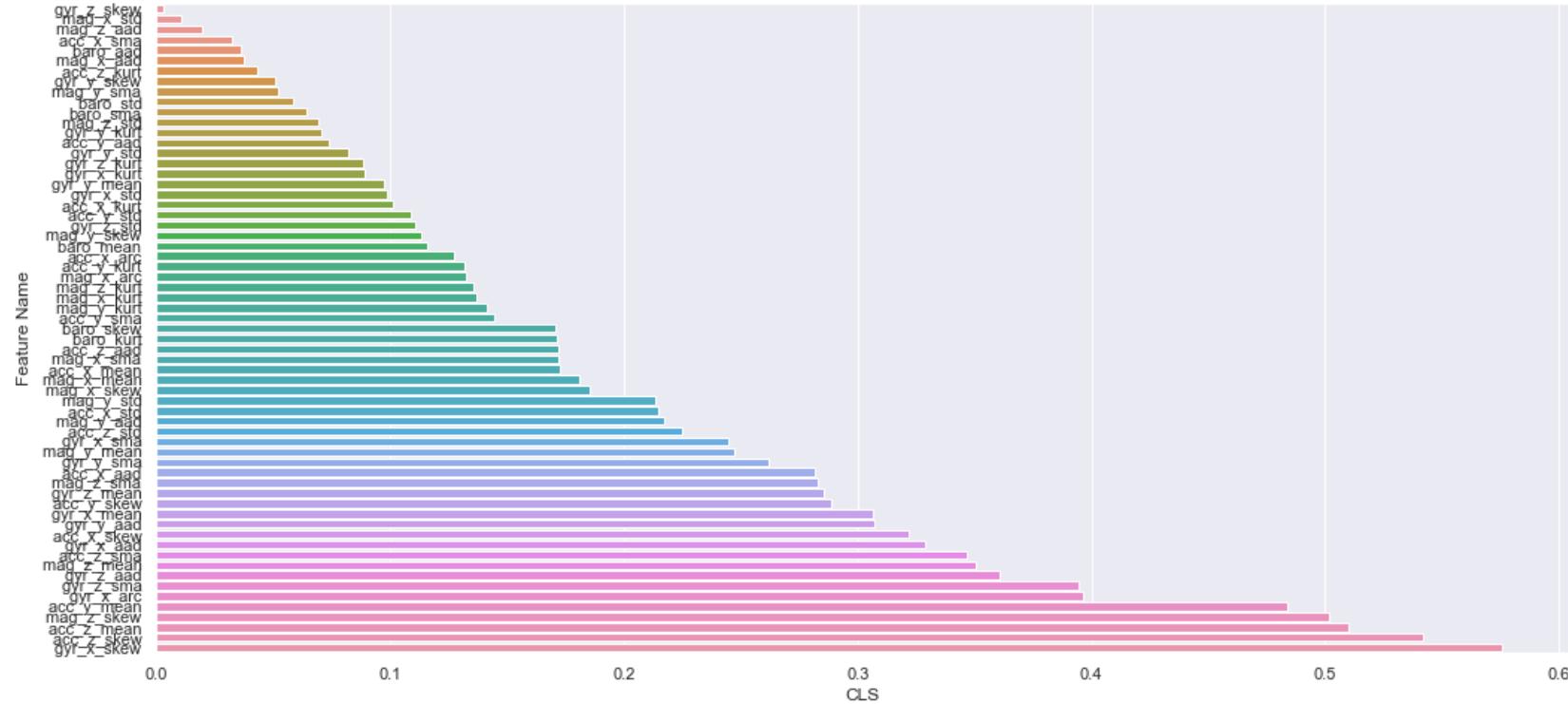
50/80

```

5 : ['gyr_z_skew', 'gyr_x_skew', 'gyr_y_kurt', 'gyr_x_kurt', 'mag_z_skew']
au dessus de 0.5 37: ['gyr_z_skew', 'gyr_x_skew', 'gyr_y_kurt', 'gyr_x_kurt', 'mag_z_skew',
'gyr_z_mean', 'gyr_z_kurt', 'gyr_y_skew', 'mag_z_mean', 'acc_y_skew',
'mag_z_kurt', 'acc_x_kurt', 'mag_x_kurt', 'acc_x_skew', 'mag_y_kurt',
'acc_z_skew', 'acc_z_kurt', 'acc_y_kurt', 'mag_x_mean', 'mag_z_sma',
'gyr_y_sma', 'gyr_z_sma', 'mag_x_skew', 'acc_z_mean', 'acc_z_sma',
'gyr_y_mean', 'mag_y_skew', 'gyr_x_sma', 'gyr_x_mean', 'mag_y_aad',
'mag_y_mean', 'acc_y_mean', 'acc_y_sma', 'gyr_z_aad', 'mag_y_sma',
'acc_x_std', 'acc_x_sma']

```

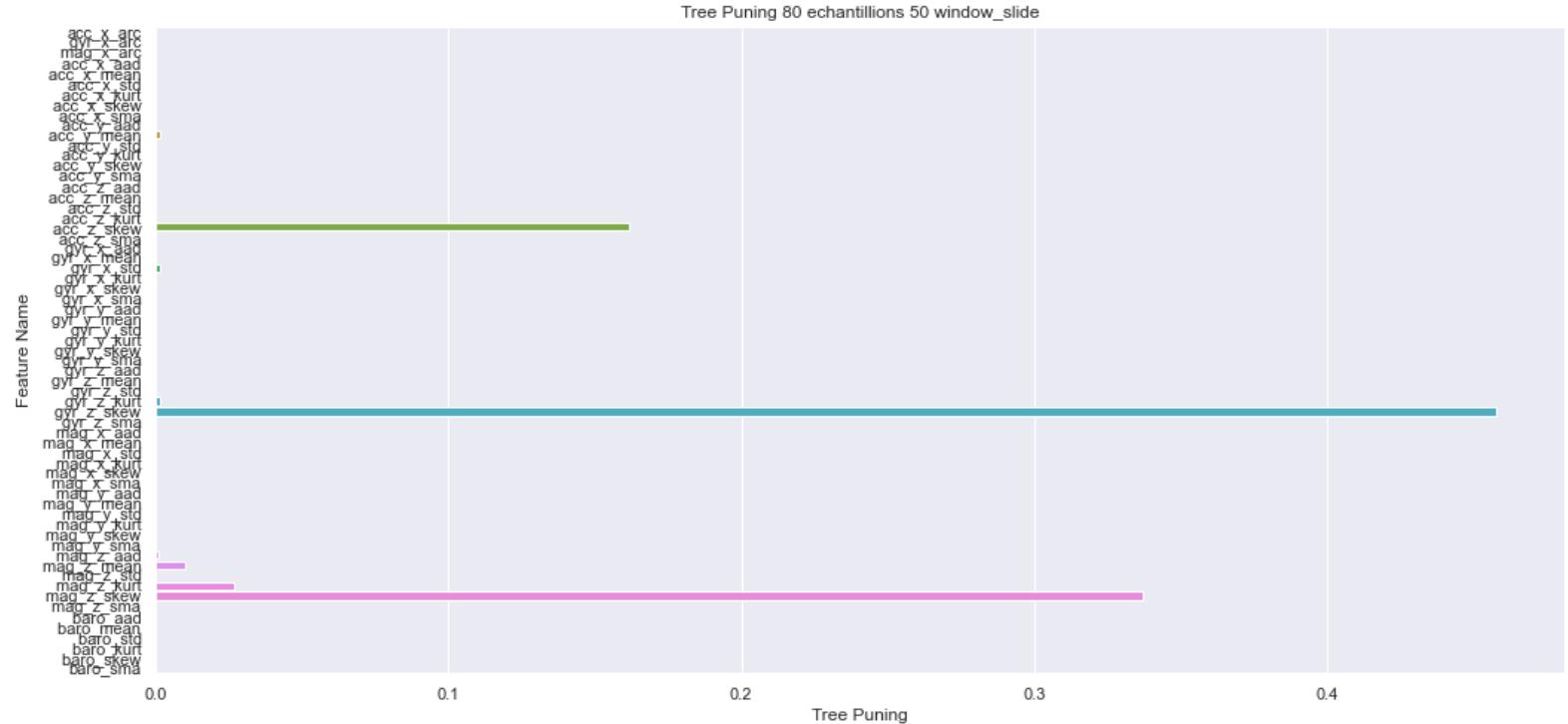
CLS 80 echantillons 50 window_slide



50 80

```
44 : ['acc_z_skew', 'acc_z_sma', 'gyr_x_aad', 'gyr_x_mean', 'gyr_x_std', 'gyr_x_kurt', 'gyr_x_skew', 'gyr_x_sma', 'gyr_y_aad', 'gyr_y_mean', 'gyr_y_std', 'gyr_y_kurt', 'gyr_y_skew', 'gyr_y_sma', 'gyr_z_aad', 'gyr_z_mean', 'gyr_z_std', 'gyr_z_kurt', 'gyr_z_skew', 'gyr_z_sma', 'mag_x_aad', 'mag_x_mean', 'mag_x_std', 'mag_x_kurt', 'mag_x_skew', 'mag_x_sma', 'mag_y_aad', 'mag_y_mean', 'mag_y_std', 'mag_y_kurt', 'mag_y_skew', 'mag_y_sma', 'mag_z_aad', 'mag_z_mean', 'mag_z_std', 'mag_z_kurt', 'mag_z_skew', 'mag_z_sma', 'baro_aad', 'baro_mean', 'baro_std', 'baro_kurt', 'baro_skew', 'baro_sma']
```

au dessus de 0.1

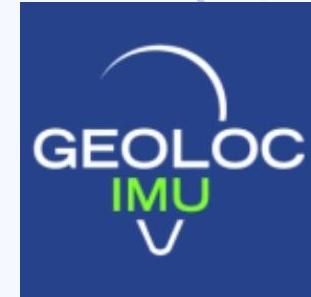


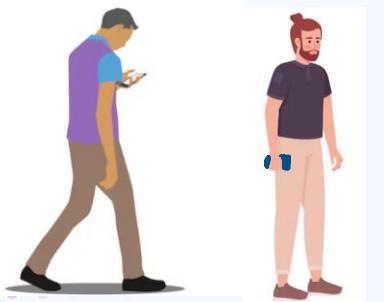
50 80

3 : [' acc_z_skew', ' gyr_z_skew', ' mag_z_skew']

au dessus de 0.1

03 Collecte de données





Position du téléphone conseillée

1. Allez sur la page de collecte des données verticales.
2. Sélectionnez l'activité que vous allez effectuer et démarrez l'enregistrement.
3. Vous pouvez changer d'activité tout en continuant l'enregistrement.
4. Arrêtez l'enregistrement et envoyez-nous vos données.

Merci de votre contribution

Pour télécharger l'application
GeolocIMU au format APK
scannez le QRcode
Attention !
Seulement sur téléphone Android



Avec le fichier APK , utiliser
l'application « Apk Installer » sur
PlayStore
ou autre pour installer l'application

Pour toutes questions envoyer un
mail à:
laura.parisot@univ-eiffel.fr

04 Rapport installation pour ENSG



Rapport d'installation TFE Laura Parisot Détection du mode de déplacement à l'intérieur des bâtiments GeolocIMU



Laura PARISOT

Mai 2024

I Condition d'arrivée

Le stage se déroule au sein du laboratoire Geoloc de l'Université Gustave Eiffel. Cependant, le bureau principal de Geoloc est principalement situé sur le campus de l'UGE à Nantes. Mon maître de stage, Tommy Moisan, est également basé sur le campus de Marne la Vallée, où je suis également installé.

Mon lieu de travail est situé dans les locaux du laboratoire LAMA (Laboratoire d'Analyse Mathématiques Appliquées) sous la supervision de Thomas Bonis. Ces locaux se trouvent au bâtiment Copernic de l'Université Gustave Eiffel à Champs-sur-Marne. Je partage le bureau avec un autre stagiaire du LAMA.

Il est prévu que je fasse ponctuellement des déplacements sur le campus de Nantes afin de m'intégrer et de discuter avec les autres membres du laboratoire Geoloc.

Un ordinateur portable m'a été mis à disposition, et j'ai également la possibilité de travailler avec mon propre ordinateur personnel. Une adresse e-mail affiliée à l'université Gustave Eiffel m'a été affiliée tout comme une carte "MRS" pour accéder à la cantine.

Mon stage a débuté par une réunion de démarrage où étaient présents mes deux maîtres de stage, Tommy Moisan et Valérie Renaudin, ainsi que Thomas Bonis, qui me conseillera également tout au long de mon stage. Des réunions de compte rendu d'avancement sont prévues pendant le stage avec ces mêmes personnes, à raison d'une réunion toutes les deux semaines.

II Reformulation du sujet

Le sujet de mon stage concerne la détection automatique, via les capteurs des smartphones, du mode de changement d'étage. Il a été établi que les modes de changement d'étage pris en compte sont l'ascenseur, les escaliers et les escalators. Cette détection automatique devra également distinguer ces modes de déplacement d'une activité de marche ou d'immobilité. Le but est de pouvoir attribuer un label dans un parcours à l'intérieur d'un bâtiment, indiquant la situation de déplacement de la personne portant le téléphone.

Pour ce faire, l'application GeolocIMU développée par le laboratoire Geoloc se verra introduire plusieurs fonctionnalités concernant cette problématique de changements d'étage.

Tout d'abord, ajouter la fonctionnalité de collecte de données labellisées manuellement selon le mode de déplacement à l'intérieur des bâtiments. Ensuite, la mise en place d'un modèle de prédiction automatique intégré dans l'application permettant de faire une prédiction sur le mode de changement d'étage, qui pourra être visualisé par l'utilisateur du smartphone. Au final, la nouvelle fonctionnalité de l'application devra permettre, grâce au modèle, de collecter des données labellisées automatiquement.

Les données collectées sont les données d'accéléromètre, de gyroscope, de magnétomètre et de baromètre.

Une contrainte du modèle qui sera formé est qu'il doit être adapté à une utilisation sous application Android, donc remplir des conditions telles qu'avoir un temps d'exécution le plus rapide possible tout en prenant le moins de mémoire de l'appareil possible, et aussi avec un poids le plus petit possible. Les nouvelles fonctionnalités ne doivent pas nuire aux autres fonctionnalités de l'application GeolocIMU.

III Principales tâches et solutions envisagées

1. Ajout de la fonctionnalité de collecte de données avec les modes de déplacement à l'intérieur des bâtiments dans l'application GeolocIMU. Pour cela, il faudra s'aider du code de l'application déjà existante pour ajouter une nouvelle page codée en Kotlin.
2. Lancer une collecte des données par des personnes volontaires pour avoir le plus de situations possibles et des données sur lesquelles entraîner le classifieur. Demander au collègue du laboratoire Geoloc/LAMA et aux personnes de l'entourage.
3. Faire une sélection des caractéristiques à prendre en compte dans les données pour former le meilleur modèle, optimisé le plus possible. De nombreux tests et comparaisons devront être faits concernant les points suivants : combien de données contient le vecteur mis en entrée lors de l'extraction de caractéristiques, le pourcentage de recouvrement entre les données, la méthode de filtrage des données, quelles caractéristiques temporelles et fréquentielles utiliser pour extraire des informations des données brutes, la méthode de sélection de ces caractéristiques, et quel classifieur utiliser pour former le modèle de détection automatique.
4. Former le modèle le plus optimal pour la problématique. Faire une étape de fine tuning et s'aider de l'étape de sélection des caractéristiques.
5. Intégration du modèle dans l'application pour visualiser les prédictions. Développer un code en C++ et Kotlin en essayant de le connecter avec les autres fonctionnalités de l'application déjà présentes.
6. Évaluer le modèle en pratique. Faire des tests dans différents bâtiments et situations.
7. Intégrer la fonctionnalité de labellisation des données automatiquement par le modèle. Créer des fichiers txt associés à chaque capteur.

Tâches intermédiaires : Former des rapports intermédiaires sur l'état de l'art de la problématique, sur les différents changements effectués dans l'application et les protocoles mis en place. Écrire la documentation sur le code développé.

IV Calendrier prévisionnel

Le stage a commencé le Lundi 22 Avril 2024 et finira le Vendredi 25 Octobre 2024.

- **Avril 2024** : Recherche bibliographie, Prise de connaissance de l'application, ajout de la fonctionnalité de collecte de données.
- **Mai 2024** : Formation du protocole de collecte des données, lancement de la collecte, et établissement du protocole de sélection des caractéristiques.
- **Juin 2024** : Continuation de la sélection des caractéristiques, Mise en forme et tri des premières données collectées.
- **Juillet 2024** : Formation du meilleur modèle (fine-tuning des paramètres).
- **Août 2024** : Mise en place du modèle dans l'application et évaluation de sa performance, préparation du rapport final et de l'oral de stage.
- **Septembre 2024** : Écriture des rapports intermédiaire et final, préparation et présentation de l'oral de stage.
- **Octobre 2024** : Perfectionnement du code et du modèle développé.

Durant tout le long du stage, des rapports intermédiaires des différents changements dans l'application et protocoles mis en place seront écrits. De plus, les recherches bibliographiques seront faites tout au long du stage, au vu des différentes questions soulevées par la problématique, et seront regroupées dans un rapport d'état de l'art.

FIN

05 La suite ...



1. Lancer la phase test de la collecte de données
2. Continuer l' évaluation des caractéristiques
3. Préparer la conférence de Lundi prochain

FIN
merci de votre
attention

Bibliographie

- [2] Sara KHALIFA, Mahbub HASSAN et Aruna SENEVIRATNE. "Feature Selection for Floor-changing Activity Recognition in Multi-Floor Pedestrian Navigation". In : (1^{er} jan. 2014).
DOI : [10.1109/ICMU.2014.6799049](https://doi.org/10.1109/ICMU.2014.6799049).