Compte rendu 1

Stage Laura Parisot
Détection du changement d'étage

Laura Parisot

07/05/2024

TEAM GEOLOC

Sommaire

01

Etat de l'art

https://fr.overleaf.com/5851543551rptttdhncptd#ecd6f2

Etat de l'art:

[4] (PDF) A Comprehensive Study of Smartphone-based Indoor Activity Recognition via XGBoost (researchgate.net)

[1] (PDF) SemanticSLAM: Using Environment Landmarks for Unsupervised Indoor Localization (researchgate.net)

IEEE Access

Received May 20, 2019, accepted June 10, 2019, date of publication June 14, 2019, date of current version July 1, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2922974

A Comprehensive Study of Smartphone-Based Indoor Activity Recognition via Xgboost

WENTING ZHANG, XIAOHUI ZHAO, (Member, IEEE), AND ZAN LI, (Member, IEEE)

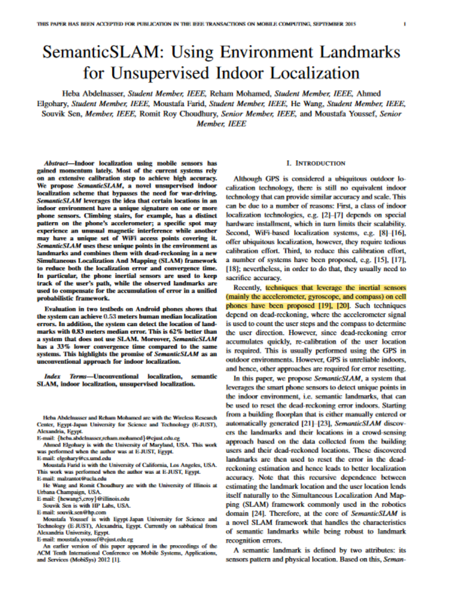
College of Communication Engineering, Jilin University, Changchun 130012, China

Corresponding author: Zan Li (zanli@jlu.edu.cn)

This work was supported by the Natural Science Foundation of China under Grant 61801189.

ABSTRACT Recently, multi-floor indoor positioning has become increasingly interesting for researchers, in which accurate recognition of indoor activities is critical for the detection of floor changes and the improvement of positioning accuracy according to indoor landmarks. However, we have not found a comprehensive study for recognizing indoor activities related to multi-floor indoor positioning based on a robust machine learning algorithm. In this work, we propose a framework for recognizing five indoor activities, i.e., walking, stillness, stair climbing, escalator, or elevator taking. In this framework, we investigate the relevant sensors and features to improve the recognition accuracy of these activities, especially some specific features in the frequency domain and wavelet domain. We propose to utilize a promising tree-based ensemble learning classifier, XGBoost, to recognize these activities. Based on our dataset created by 40 volunteers, we provide a comprehensive analysis of the proposed framework for indoor activity recognition. Considering both accuracy and computational cost, the XGBoost-based indoor activity recognition algorithm outperforms the other ensemble learning classifiers and single classifiers, and the average recognition F-score of XGBoost reaches 84.41%. In addition, our introduced specific features in the frequency domain and wavelet domain can significantly improve the recognition accuracy. Moreover, we use a publicly available dataset to verify our proposed framework and XGBoost classifier reaches 84.19% that outperforms the other classifiers.

INDEX TERMS Activity recognition, ensemble learning, XGBoost, smartphone.



SemanticSLAM: Using Environment Landmarks for Unsupervised Indoor Localization

Heba Abdelnasser, *Student Member, IEEE,* Reham Mohamed, *Student Member, IEEE,* Ahmed Elgohary, *Student Member, IEEE,* Moustafa Farid, *Student Member, IEEE,* He Wang, *Student Member, IEEE,* Souvik Sen, *Member, IEEE,* Romit Roy Choudhury, *Senior Member, IEEE,* and Moustafa Youssef, *Senior Member, IEEE*

***Abstract*—Indoor localization using mobile sensors has gained momentum lately. Most of the current systems rely on an extensive calibration step to achieve high accuracy. We propose *SemanticSLAM*, a novel unsupervised indoor localization scheme that bypasses the need for war-driving. *SemanticSLAM* leverages the idea that certain locations in an indoor environment have a unique signature on one or more phone sensors. Climbing stairs, for example, has a distinct pattern on the phone's accelerometer; a specific spot may experience an unusual magnetic interference while another may have a unique set of WiFi access points covering it. *SemanticSLAM* uses these unique points in the environment as landmarks and combines them with dead-reckoning in a new Simultaneous Localization And Mapping (SLAM) framework to reduce both the localization error and convergence time. In particular, the phone inertial sensors are used to keep track of the user's path, while the observed landmarks are used to compensate for the accumulation of error in a unified probabilistic framework.**

Evaluation in two testbeds on Android phones shows that the system can achieve 0.53 meters human median localization errors. In addition, the system can detect the location of landmarks with 0.83 meters median error. This is 62% better than a system that does not use SLAM. Moreover, *SemanticSLAM* has a 33% lower convergence time compared to the same systems. This highlights the promise of *SemanticSLAM* as an unconventional approach for indoor localization.

***Index Terms*—Unconventional localization, semantic SLAM, indoor localization, unsupervised localization.**

https://fr.overleaf.com/5851543551rptttdhncptd#ecd6f2

Etat de l’art:

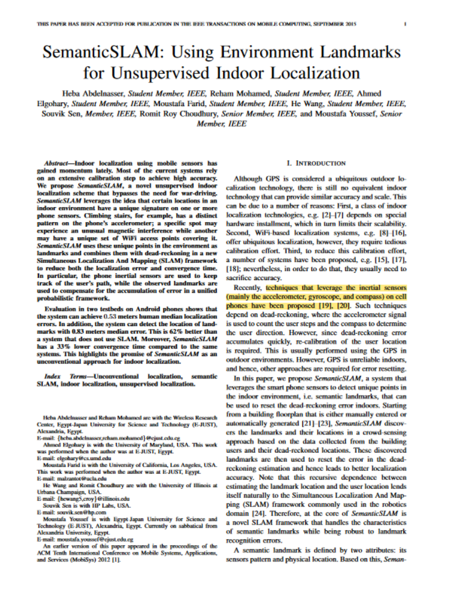
SLAM : Simultaneous Localisation And Mapping

Landmark : lieu positionné de manière précise dans un environnement
dans lequel on observes des « pattern » dans les signaux des capteurs du téléphone.

Correction des erreurs de Pedestrial Dead Reckoning (PDR) lorsque on passe sur l’un de ces lieux et amélioration du temps de convergence selon leur étude

Semantic Landmark = Escalator, Stair , Elevator soient des landmark (POI) qui ont un pattern dans les signaux reconnaissables quel que soient le bâtiment

[1] (PDF) SemanticSLAM: Using Environment Landmarks for Unsupervised Indoor Localization (researchgate.net)



THIS PAPER HAS BEEN ACCEPTED FOR PUBLICATION IN THE IEEE TRANSACTIONS ON MOBILE COMPUTING, SEPTEMBER 2015 1

SemanticSLAM: Using Environment Landmarks for Unsupervised Indoor Localization

Heba Abdelnasser, *Student Member, IEEE,* Reham Mohamed, *Student Member, IEEE,* Ahmed Elgohary, *Student Member, IEEE,* Moustafa Farid, *Student Member, IEEE,* He Wang, *Student Member, IEEE,* Souvik Sen, *Member, IEEE,* Romit Roy Choudhury, *Senior Member, IEEE,* and Moustafa Youssef, *Senior Member, IEEE*

***Abstract*—Indoor localization using mobile sensors has gained momentum lately. Most of the current systems rely on an extensive calibration step to achieve high accuracy. We propose *SemanticSLAM*, a novel unsupervised indoor localization scheme that bypasses the need for war-driving. *SemanticSLAM* leverages the idea that certain locations in an indoor environment have a unique signature on one or more phone sensors. Climbing stairs, for example, has a distinct pattern on the phone's accelerometer; a specific spot may experience an unusual magnetic interference while another may have a unique set of WiFi access points covering it. *SemanticSLAM* uses these unique points in the environment as landmarks and combines them with dead-reckoning in a new Simultaneous Localization And Mapping (SLAM) framework to reduce both the localization error and convergence time. In particular, the phone inertial sensors are used to keep track of the user's path, while the observed landmarks are used to compensate for the accumulation of error in a unified probabilistic framework.**

Evaluation in two testbeds on Android phones shows that the system can achieve 0.53 meters human median localization errors. In addition, the system can detect the location of landmarks with 0.83 meters median error. This is 62% better than a system that does not use SLAM. Moreover, *SemanticSLAM* has a 33% lower convergence time compared to the same systems. This highlights the promise of *SemanticSLAM* as an unconventional approach for indoor localization.

***Index Terms*—Unconventional localization, semantic SLAM, indoor localization, unsupervised localization.**

Heba Abdelnasser and Reham Mohamed are with the Wireless Research Center, Egypt-Japan University for Science and Technology (E-JUST), Alexandria, Egypt.
E-mail: {heba.abdelnasser,reham.mohamed}@ejust.edu.eg
Ahmed Elgohary is with the University of Maryland, USA. This work was performed when the author was at E-JUST, Egypt.
E-mail: elgohary@cs.umd.edu
Moustafa Farid is with the University of California, Los Angeles, USA. This work was performed when the author was at E-JUST, Egypt.
E-mail: malzantot@ucla.edu
He Wang and Romit Choudhury are with the University of Illinois at Urbana-Champaign, USA.
E-mail: {hewang5,croy}@illinois.edu
Souvik Sen is with HP Labs, USA.
E-mail: souvik.sen@hp.com
Moustafa Youssef is with Egypt-Japan University for Science and Technology (E-JUST), Alexandria, Egypt. Currently on sabbatical from Alexandria University, Egypt.
E-mail: moustafa.youssef@ejust.edu.eg
An earlier version of this paper appeared in the proceedings of the ACM Tenth International Conference on Mobile Systems, Applications, and Services (MobiSys) 2012 [1].

I. Introduction

Although GPS is considered a ubiquitous outdoor localization technology, there is still no equivalent indoor technology that can provide similar accuracy and scale. This can be due to a number of reasons: First, a class of indoor localization technologies, e.g. [2]–[7] depends on special hardware installment, which in turn limits their scalability. Second, WiFi-based localization systems, e.g. [8]–[16], offer ubiquitous localization, however, they require tedious calibration effort. Third, to reduce this calibration effort, a number of systems have been proposed, e.g. [15], [17], [18]; nevertheless, in order to do that, they usually need to sacrifice accuracy.

Recently, techniques that leverage the inertial sensors (mainly the accelerometer, gyroscope, and compass) on cell phones have been proposed [19], [20]. Such techniques depend on dead-reckoning, where the accelerometer signal is used to count the user steps and the compass to determine the user direction. However, since dead-reckoning error accumulates quickly, re-calibration of the user location is required. This is usually performed using the GPS in outdoor environments. However, GPS is unreliable indoors, and hence, other approaches are required for error resetting.

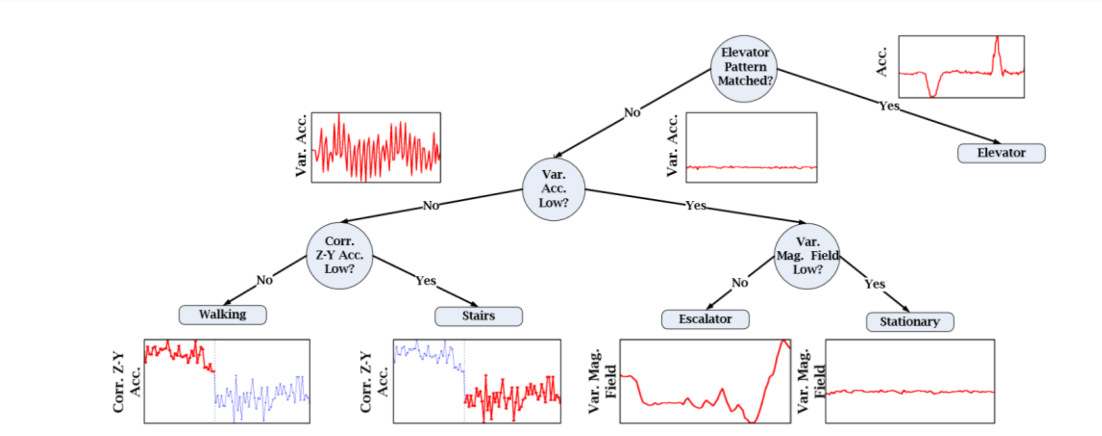
In this paper, we propose *SemanticSLAM*, a system that leverages the smart phone sensors to detect unique points in the indoor environment, i.e. semantic landmarks, that can be used to reset the dead-reckoning error indoors. Starting from a building floorplan that is either manually entered or automatically generated [21]–[23], *SemanticSLAM* discovers the landmarks and their locations in a crowd-sensing approach based on the data collected from the building users and their dead-reckoned locations. These discovered landmarks are then used to reset the error in the dead-reckoning estimation and hence leads to better localization accuracy. Note that this recursive dependence between estimating the landmark location and the user location lends itself naturally to the Simultaneous Localization And Mapping (SLAM) framework commonly used in the robotics domain [24]. Therefore, at the core of *SemanticSLAM* is a novel SLAM framework that handles the characteristics of semantic landmarks while being robust to landmark recognition errors.

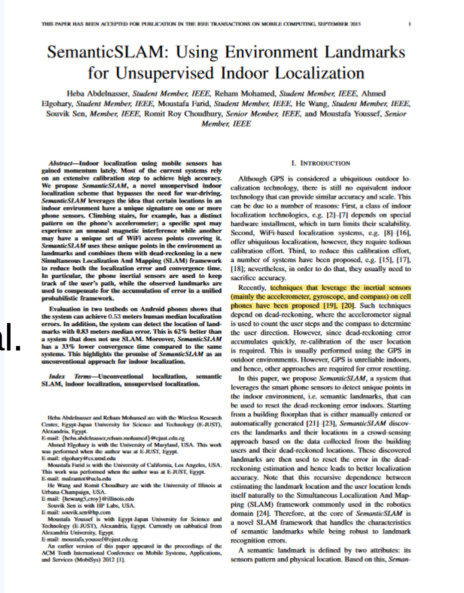
A semantic landmark is defined by two attributes: its sensors pattern and physical location. Based on this, *Seman-*

https://fr.overleaf.com/5851543551rptttdhncptd#ecd6f2

Etat de l’art:

[[1] (PDF) SemanticSLAM: Using Environment Landmarks for Unsupervised Indoor Localization (researchgate.net)]





THIS PAPER HAS BEEN ACCEPTED FOR PUBLICATION IN THE IEEE TRANSACTIONS ON MOBILE COMPUTING, SEPTEMBER 2015 1

SemanticSLAM: Using Environment Landmarks for Unsupervised Indoor Localization

Heba Abdelnasser, *Student Member, IEEE,* Reham Mohamed, *Student Member, IEEE,* Ahmed Elgohary, *Student Member, IEEE,* Moustafa Farid, *Student Member, IEEE,* He Wang, *Student Member, IEEE,* Souvik Sen, *Member, IEEE,* Romit Roy Choudhury, *Senior Member, IEEE,* and Moustafa Youssef, *Senior Member, IEEE*

***Abstract*—Indoor localization using mobile sensors has gained momentum lately. Most of the current systems rely on an extensive calibration step to achieve high accuracy. We propose *SemanticSLAM*, a novel unsupervised indoor localization scheme that bypasses the need for war-driving. *SemanticSLAM* leverages the idea that certain locations in an indoor environment have a unique signature on one or more phone sensors. Climbing stairs, for example, has a distinct pattern on the phone's accelerometer; a specific spot may experience an unusual magnetic interference while another may have a unique set of WiFi access points covering it. *SemanticSLAM* uses these unique points in the environment as landmarks and combines them with dead-reckoning in a new Simultaneous Localization And Mapping (SLAM) framework to reduce both the localization error and convergence time. In particular, the phone inertial sensors are used to keep track of the user's path, while the observed landmarks are used to compensate for the accumulation of error in a unified probabilistic framework.**

Evaluation in two testbeds on Android phones shows that the system can achieve 0.53 meters human median localization errors. In addition, the system can detect the location of landmarks with 0.83 meters median error. This is 62% better than a system that does not use SLAM. Moreover, *SemanticSLAM* has a 33% lower convergence time compared to the same systems. This highlights the promise of *SemanticSLAM* as an unconventional approach for indoor localization.

***Index Terms*—Unconventional localization, semantic SLAM, indoor localization, unsupervised localization.**

Heba Abdelnasser and Reham Mohamed are with the Wireless Research Center, Egypt-Japan University for Science and Technology (E-JUST), Alexandria, Egypt.
E-mail: {heba.abdelnasser,reham.mohamed}@ejust.edu.eg
Ahmed Elgohary is with the University of Maryland, USA. This work was performed when the author was at E-JUST, Egypt.
E-mail: elgohary@cs.umd.edu
Moustafa Farid is with the University of California, Los Angeles, USA. This work was performed when the author was at E-JUST, Egypt.
E-mail: malzantot@ucla.edu
He Wang and Romit Choudhury are with the University of Illinois at Urbana-Champaign, USA.
E-mail: {hewang5,croy}@illinois.edu
Souvik Sen is with HP Labs, USA.
E-mail: souvik.sen@hp.com
Moustafa Youssef is with Egypt-Japan University for Science and Technology (E-JUST), Alexandria, Egypt. Currently on sabbatical from Alexandria University, Egypt.
E-mail: moustafa.youssef@ejust.edu.eg
An earlier version of this paper appeared in the proceedings of the ACM Tenth International Conference on Mobile Systems, Applications, and Services (MobiSys) 2012 [1].

I. INTRODUCTION

Although GPS is considered a ubiquitous outdoor localization technology, there is still no equivalent indoor technology that can provide similar accuracy and scale. This can be due to a number of reasons: First, a class of indoor localization technologies, e.g. [2]–[7] depends on special hardware installment, which in turn limits their scalability. Second, WiFi-based localization systems, e.g. [8]–[16], offer ubiquitous localization, however, they require tedious calibration effort. Third, to reduce this calibration effort, a number of systems have been proposed, e.g. [15], [17], [18]; nevertheless, in order to do that, they usually need to sacrifice accuracy.

Recently, techniques that leverage the inertial sensors (mainly the accelerometer, gyroscope, and compass) on cell phones have been proposed [19], [20]. Such techniques depend on dead-reckoning, where the accelerometer signal is used to count the user steps and the compass to determine the user direction. However, since dead-reckoning error accumulates quickly, re-calibration of the user location is required. This is usually performed using the GPS in outdoor environments. However, GPS is unreliable indoors, and hence, other approaches are required for error resetting.

In this paper, we propose *SemanticSLAM*, a system that leverages the smart phone sensors to detect unique points in the indoor environment, i.e. semantic landmarks, that can be used to reset the dead-reckoning error indoors. Starting from a building floorplan that is either manually entered or automatically generated [21]–[23], *SemanticSLAM* discovers the landmarks and their locations in a crowd-sensing approach based on the data collected from the building users and their dead-reckoned locations. These discovered landmarks are then used to reset the error in the dead-reckoning estimation and hence leads to better localization accuracy. Note that this recursive dependence between estimating the landmark location and the user location lends itself naturally to the Simultaneous Localization And Mapping (SLAM) framework commonly used in the robotics domain [24]. Therefore, at the core of *SemanticSLAM* is a novel SLAM framework that handles the characteristics of semantic landmarks while being robust to landmark recognition errors.

A semantic landmark is defined by two attributes: its sensors pattern and physical location. Based on this, *Seman-*

Il est possible de distinguer stair et ecalator avec des statistiques sur le signal.

Utilise Magnétomètre

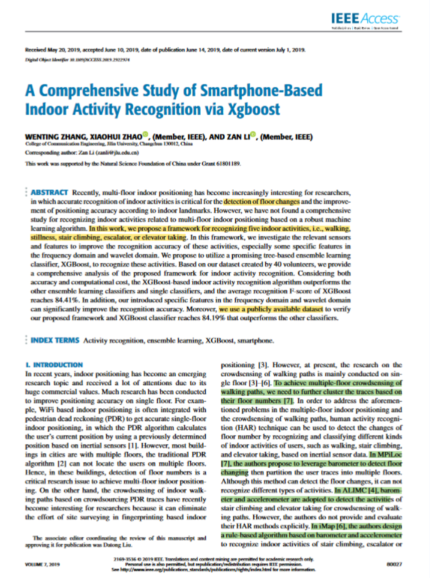
Elevator plus difficile de le distinguer statistiquement sans avoir tout le signal

Etat de l’art:

[4] (PDF) A Comprehensive Study of Smartphone-based Indoor Activity Recognition via XGBoost (researchgate.net)

Toute un protocole pour classifié avec du machine learning les 5 activités suivantes :

+Walk
+Standing Still
+Escalator
+Elevator
+Stair



IEEE Access

Received May 20, 2019, accepted June 10, 2019, date of publication June 14, 2019, date of current version July 1, 2019.
Digital Object Identifier 10.1109/ACCESS.2019.2922974

A Comprehensive Study of Smartphone-Based Indoor Activity Recognition via Xgboost

WENTING ZHANG, XIAOHUI ZHAO, (Member, IEEE), AND ZAN LI, (Member, IEEE)
College of Communication Engineering, Jilin University, Changchun 130012, China
Corresponding author: Zan Li (zanli@jlu.edu.cn)

This work was supported by the Natural Science Foundation of China under Grant 61801189.

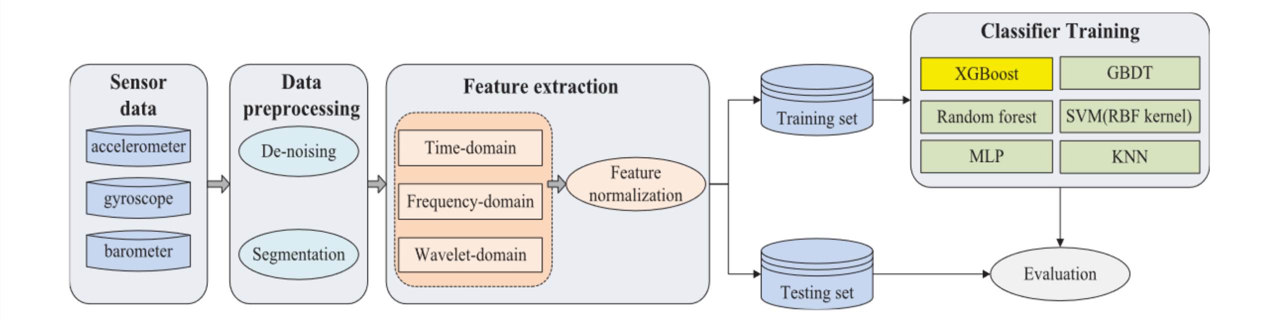
ABSTRACT Recently, multi-floor indoor positioning has become increasingly interesting for researchers, in which accurate recognition of indoor activities is critical for the detection of floor changes and the improvement of positioning accuracy according to indoor landmarks. However, we have not found a comprehensive study for recognizing indoor activities related to multi-floor indoor positioning based on a robust machine learning algorithm. In this work, we propose a framework for recognizing five indoor activities, i.e., walking, stillness, stair climbing, escalator, or elevator taking. In this framework, we investigate the relevant sensors and features to improve the recognition accuracy of these activities, especially some specific features in the frequency domain and wavelet domain. We propose to utilize a promising tree-based ensemble learning classifier, XGBoost, to recognize these activities. Based on our dataset created by 40 volunteers, we provide a comprehensive analysis of the proposed framework for indoor activity recognition. Considering both accuracy and computational cost, the XGBoost-based indoor activity recognition algorithm outperforms the other ensemble learning classifiers and single classifiers, and the average recognition F-score of XGBoost reaches 84.41%. In addition, our introduced specific features in the frequency domain and wavelet domain can significantly improve the recognition accuracy. Moreover, we use a publicly available dataset to verify our proposed framework and XGBoost classifier reaches 84.19% that outperforms the other classifiers.

INDEX TERMS Activity recognition, ensemble learning, XGBoost, smartphone.

I. INTRODUCTION

In recent years, indoor positioning has become an emerging research topic and received a lot of attentions due to its huge commercial values. Much research has been conducted to improve positioning accuracy on single floor. For example, WiFi based indoor positioning is often integrated with pedestrian dead reckoning (PDR) to get accurate single-floor indoor positioning, in which the PDR algorithm calculates the user's current position by using a previously determined position based on inertial sensors [1]. However, most buildings in cities are with multiple floors, the traditional PDR algorithm [2] can not locate the users on multiple floors. Hence, in these buildings, detection of floor numbers is a critical research issue to achieve multi-floor indoor positioning. On the other hand, the crowdsensing of indoor walking paths based on crowdsourcing PDR traces have recently become interesting for researchers because it can eliminate the effort of site surveying in fingerprinting based indoor positioning [3]. However, at present, the research on the crowdsensing of walking paths is mainly conducted on single floor [3]–[6]. To achieve multiple-floor crowdsensing of walking paths, we need to further cluster the traces based on their floor numbers [7]. In order to address the aforementioned problems in the multiple-floor indoor positioning and the crowdsensing of walking paths, human activity recognition (HAR) technique can be used to detect the changes of floor number by recognizing and classifying different kinds of indoor activities of users, such as walking, stair climbing, and elevator taking, based on inertial sensor data. In MPiLoc [7], the authors propose to leverage barometer to detect floor changing then partition the user traces into multiple floors. Although this method can detect the floor changes, it can not recognize different types of activities. In ALIMC [4], barometer and accelerometer are adopted to detect the activities of stair climbing and elevator taking for crowdsensing of walking paths. However, the authors do not provide and evaluate their HAR methods explicitly. In iMap [6], the authors design a rule-based algorithm based on barometer and accelerometer to recognize indoor activities of stair climbing, escalator or

The associate editor coordinating the review of this manuscript and approving it for publication was Datong Liu.



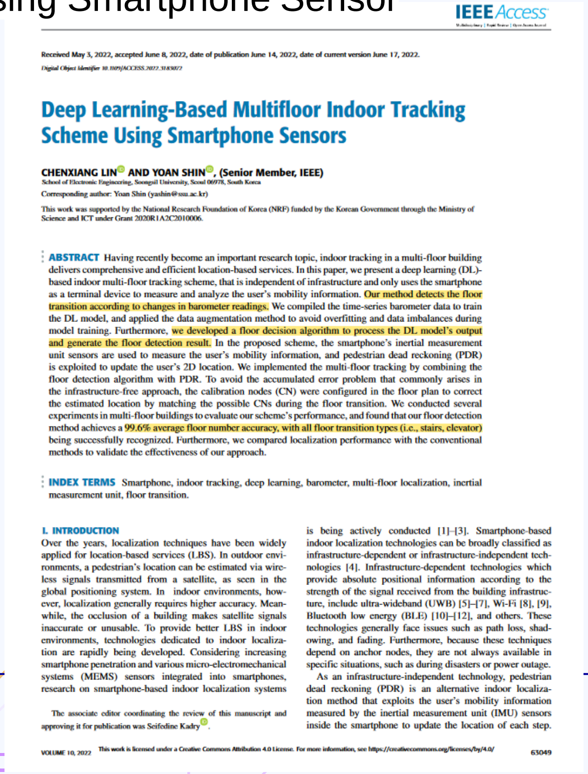
XGBoost une classifieur d’ensemble a obtenu les meilleurs résultats

L’utilisateur de feature dans le domaine frequentiel ameliore le resultat tou
Comme l’utilisation des valeurs du barometre

La position de telephone lors de la collecte des données influent car statistique différentes (poche suit le mouvement du corps , pas le sac)

Etat de l’art:

[3]Deep Learning-Based Multifloor Indoor Tracking Scheme Using Smartphone Sensor



IEEE Access
Received May 3, 2022, accepted June 8, 2022, date of publication June 14, 2022, date of current version June 17, 2022.
Digital Object Identifier 10.1109/ACCESS.2022.3183072

Deep Learning-Based Multifloor Indoor Tracking Scheme Using Smartphone Sensors

CHENXIANG LIN AND YOAN SHIN, (Senior Member, IEEE)
School of Electronic Engineering, Soongsil University, Seoul 06978, South Korea
Corresponding author: Yoan Shin (yashin@ssu.ac.kr)

This work was supported by the National Research Foundation of Korea (NRF) funded by the Korean Government through the Ministry of Science and ICT under Grant 2020R1A2C2010006.

ABSTRACT Having recently become an important research topic, indoor tracking in a multi-floor building delivers comprehensive and efficient location-based services. In this paper, we present a deep learning (DL)-based indoor multi-floor tracking scheme, that is independent of infrastructure and only uses the smartphone as a terminal device to measure and analyze the user’s mobility information. Our method detects the floor transition according to changes in barometer readings. We compiled the time-series barometer data to train the DL model, and applied the data augmentation method to avoid overfitting and data imbalances during model training. Furthermore, we developed a floor decision algorithm to process the DL model’s output and generate the floor detection result. In the proposed scheme, the smartphone’s inertial measurement unit sensors are used to measure the user’s mobility information, and pedestrian dead reckoning (PDR) is exploited to update the user’s 2D location. We implemented the multi-floor tracking by combining the floor detection algorithm with PDR. To avoid the accumulated error problem that commonly arises in the infrastructure-free approach, the calibration nodes (CN) were configured in the floor plan to correct the estimated location by matching the possible CNs during the floor transition. We conducted several experiments in multi-floor buildings to evaluate our scheme’s performance, and found that our floor detection method achieves a 99.6% average floor number accuracy, with all floor transition types (i.e., stairs, elevator) being successfully recognized. Furthermore, we compared localization performance with the conventional methods to validate the effectiveness of our approach.

INDEX TERMS Smartphone, indoor tracking, deep learning, barometer, multi-floor localization, inertial measurement unit, floor transition.

I. INTRODUCTION

Over the years, localization techniques have been widely applied for location-based services (LBS). In outdoor environments, a pedestrian’s location can be estimated via wireless signals transmitted from a satellite, as seen in the global positioning system. In indoor environments, however, localization generally requires higher accuracy. Meanwhile, the occlusion of a building makes satellite signals inaccurate or unusable. To provide better LBS in indoor environments, technologies dedicated to indoor localization are rapidly being developed. Considering increasing smartphone penetration and various micro-electromechanical systems (MEMS) sensors integrated into smartphones, research on smartphone-based indoor localization systems is being actively conducted [1]–[3]. Smartphone-based indoor localization technologies can be broadly classified as infrastructure-dependent or infrastructure-independent technologies [4]. Infrastructure-dependent technologies which provide absolute positional information according to the strength of the signal received from the building infrastructure, include ultra-wideband (UWB) [5]–[7], Wi-Fi [8], [9], Bluetooth low energy (BLE) [10]–[12], and others. These technologies generally face issues such as path loss, shadowing, and fading. Furthermore, because these techniques depend on anchor nodes, they are not always available in specific situations, such as during disasters or power outage.

As an infrastructure-independent technology, pedestrian dead reckoning (PDR) is an alternative indoor localization method that exploits the user’s mobility information measured by the inertial measurement unit (IMU) sensors inside the smartphone to update the location of each step.

The associate editor coordinating the review of this manuscript and approving it for publication was Seifedine Kadry.

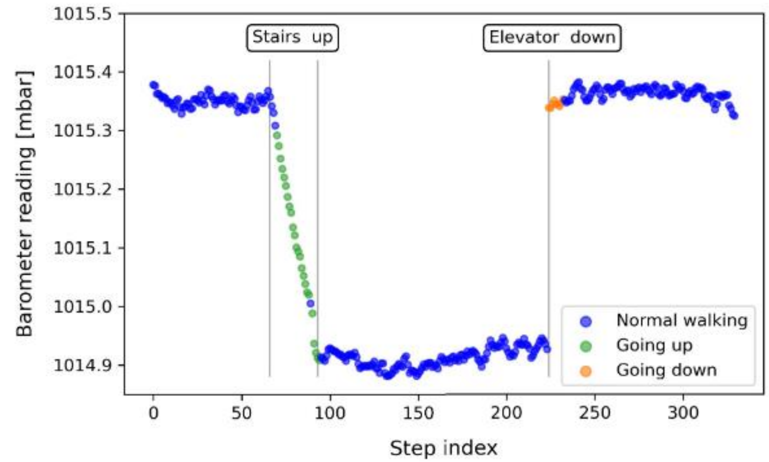
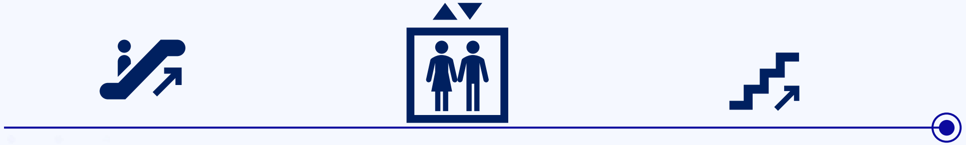
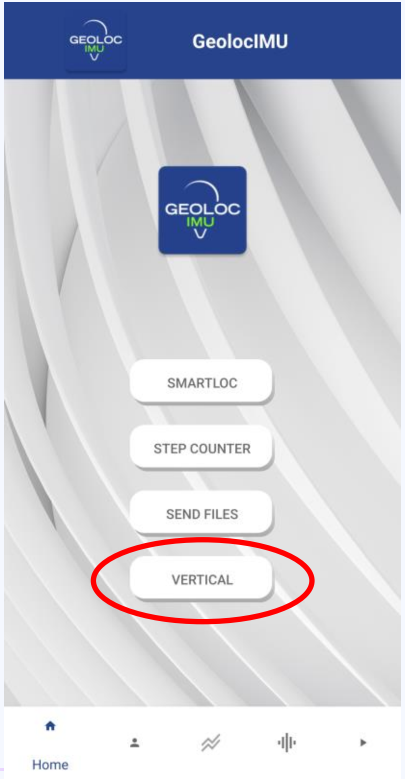


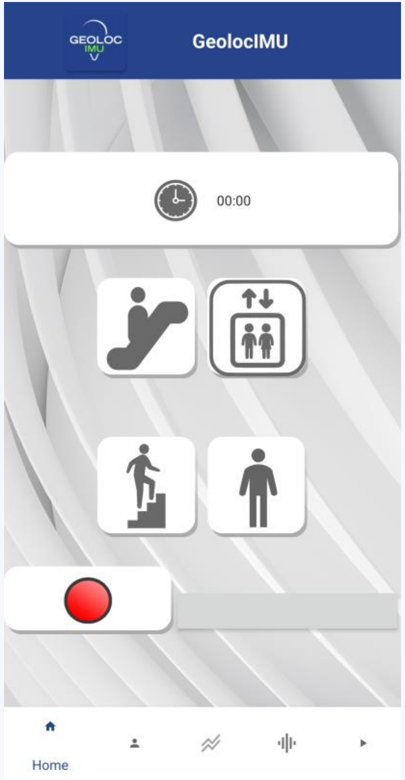
FIGURE 5. Example of step action recognition.

02 Changement dans l'application pour collecte de données



Ajout d'une nouvelle page dans l'application:





Création d'une nouvelle page VerticalFragment

Création d'un bouton sur la HomePage pour accéder à la page

4 modes de changement d'étage possible d'enregistré :

+ escalator
+ ascenseur
+ escalier
+ immobile debout

https://ensgeu-my.sharepoint.com/:b:/g/personal/laura_parisot_ensg_eu/ETo4d_xP_H5MnxIHxdpJpRgBTBhlMDDHP0pFzbSFL9Hf9g?e=TCWdn7

Baromètre:

ACC.txt	01/05/2024 12:20	Document texte	596 Ko
BARO.txt	01/05/2024 12:20	Document texte	61 Ko
GRV.txt	01/05/2024 12:20	Document texte	358 Ko
GYR.txt	01/05/2024 12:20	Document texte	498 Ko
MAG.txt	01/05/2024 12:20	Document texte	262 Ko
ROT.txt	01/05/2024 12:20	Document texte	363 Ko

```
Mode,Time [s], Pression [hPa]
escalator,296364.599,1003.133056640625
escalator,296364.636,1003.125244140625
escalator,296364.677,1003.136474609375
escalator,296364.719,1003.143310546875
escalator,296364.757,1003.141357421875
escalator,296364.795,1003.132568359375
escalator,296364.838,1003.12353515625
escalator,296364.875,1003.149169921875
escalator,296364.916,1003.182373046875
escalator,296364.955,1003.180908203125
escalator,296364.998,1003.163330078125
escalator,296365.042,1003.146484375
escalator,296365.077,1003.146484375
escalator,296365.115,1003.131591796875
escalator,296365.157,1003.1572265625
escalator,296365.198,1003.182861328125
escalator,296365.237,1003.205078125
```

+ ajout de la capture de la presson

Lien vers protocole:

Protocole en français : protocole.pptx

Protocole en anglais : protocole_anglais.pptx

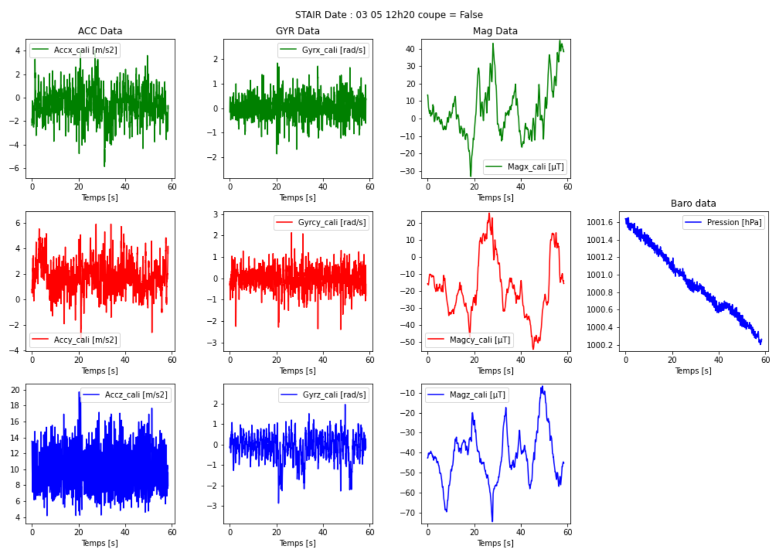
4 positions possible pour le téléphone :
+main bras le long du corps
+sac
+main bras coudée
+poche

03 Traitement des données collectées

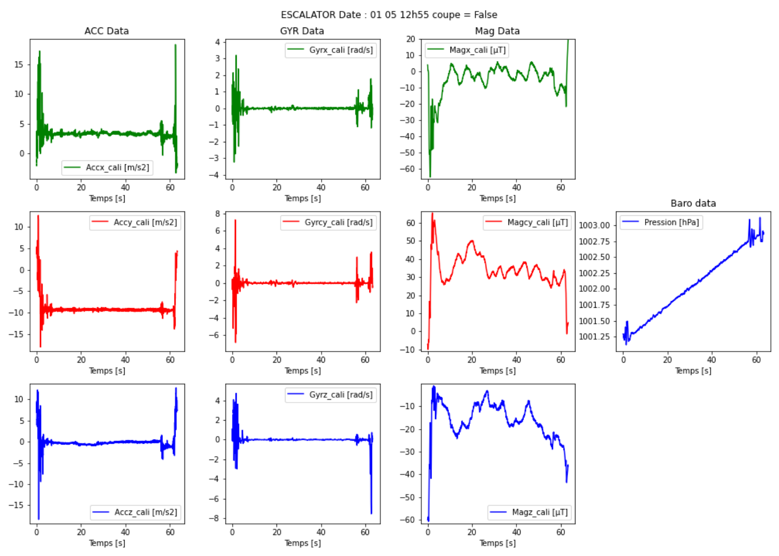
Etape 1: Interpolation des données

```python
import pandas as pd
import matplotlib.pyplot as plt
import os

def interpolate_gyrmag(chemin):

    fichiers=os.listdir(chemin)

    data_acc = np.array(pd.read_csv(rf'{chemin}\ACC.txt', delimiter=','))
    data_gyr = np.array(pd.read_csv(rf'{chemin}\GYR.txt', delimiter=','))
    data_mag = np.array(pd.read_csv(rf'{chemin}\MAG.txt', delimiter=','))
    data_baro=np.array(pd.read_csv(rf'{chemin}\BARO.txt', delimiter=','))



    t_acc = data_acc[:, 1].astype(float)
    t_mag = data_mag[:, 1].astype(float)
    t_gyr = data_gyr[:, 1].astype(float)

    data_mag_inter = np.column_stack([
        np.interp(t_acc, t_mag, data_mag[:, i].astype(float)) for i in range(2, 5)
    ])
    data_mag_inter = np.column_stack((t_acc, data_mag_inter))

    data_gyr_inter = np.column_stack([
        np.interp(t_acc, t_gyr, data_gyr[:, i].astype(float)) for i in range(2, 5)
    ])
    data_gyr_inter = np.column_stack((t_acc, data_gyr_inter))



    t_baro = data_baro[:, 1].astype(float)
    data_baro_inter = np.column_stack([
        np.interp(t_acc, t_baro, data_baro[:, 2].astype(float))
    ])
    data_baro_inter = np.column_stack((t_acc, data_baro_inter))

    data_baro=pd.DataFrame(data_baro_inter,columns=['Time [s]','Pression [hPa]'])
```

Etape 2: Visualisation des données

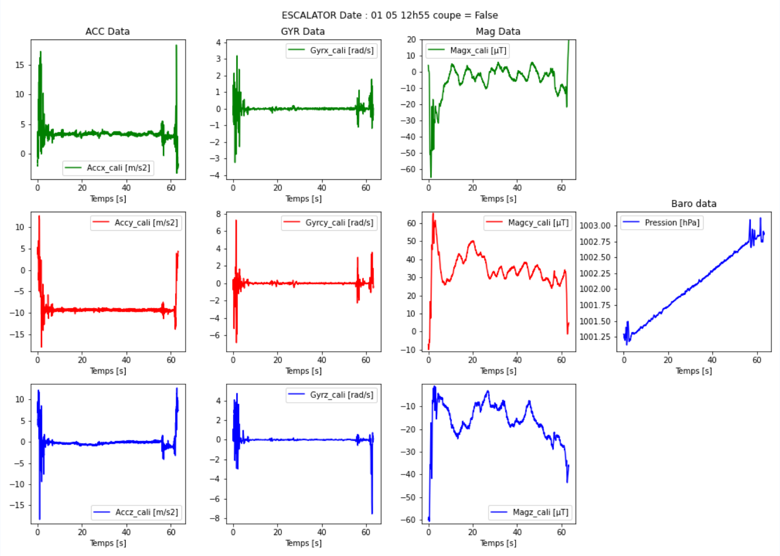


Escalier

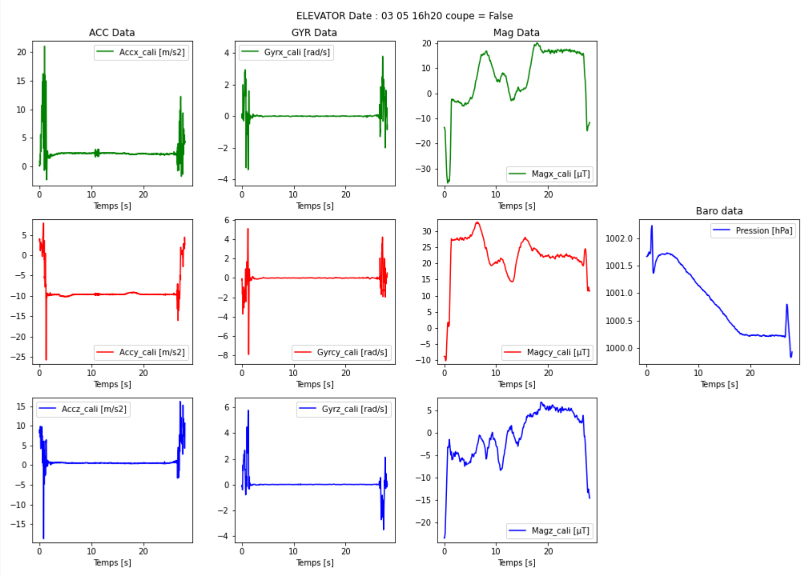


Escalator

Etape 2: Visualisation des données



Escalator



Elevator

Etape 3: Création du fichier entrée pour le code cpp

acc_x,acc_y,acc_z,gyr_x,gyr_y,gyr_z,mag_x,mag_y,mag_z

1 ligne = 1 temps d’acquisition basée sur l’acc car gyr et mag interpolée sur acc
1 fichier par mode de changement d’étage

```
-1.5170495510101318,2.091326951980591,9.300901412963867,-0.476321816444397,0.8823941946029663,-0.2266308367252349,-3.281250238418579,-1.4812500476837158,-23.137500762939453,995.72094
-1.5170495510101318,2.091326951980591,9.300901412963867,-0.476321816444397,0.8823941946029663,-0.2266308367252349,-3.281250238418579,-1.4812500476837158,-23.137500762939453,995.72094
-1.5170495510101318,2.091326951980591,9.300901412963867,-0.476321816444397,0.8823941946029663,-0.2266308367252349,-3.281250238418579,-1.4812500476837158,-23.137500762939453,995.72094
-1.7515461444854736,2.730210542678833,9.489934921264648,-0.4219548404216766,0.3882045745849609,-0.265115320682525 6,-3.281250238418579,-1.4812500476837158,-23.137500762939453,995.7209
-2.0195422172546387,2.4598217010498047,9.659825325012209,-0.3339903056621551,-0.0528398044407367,-0.3457494676113128,-2.933035816754659,-1.025892913720169,-23.633036749291488,995.720
-1.584048628807068,2.234896421432495,9.332008361816406,-0.1343597081725892,-0.9796133281300983,-0.4786430847368016,-2.7093750835283577,-0.8156250417505362,-23.8031258583341 7,995.7209
-1.584048628807068,2.256431818008423,8.726624488830566,-0.1284503976045181 8,-1.4711713922542 86,-0.5535399224574357,-2.7862500792020004,-0.832500031721429,-23.688751029444393,995.7209
-1.856830358505249,2.3880369663238525,9.382257461547852,-0.135982866763297,-1.7788818905521824,-0.6035343834281262,-2.921250057045836,-0.8662500261823 88,-23.553751373465634,995.72094
-1.6606189012527466,2.3736801147460938,9.53061294555664,-0.0768162459135055,-1.433394193649292,-0.5271763205528259,-2.981250047683716,-0.8562500476490684,-23.443751017183306,995.7209
-1.203589797019958 5,1.959721803665161,9.69093132019043,0.0168068211105529 4,-1.3438864383403601,-0.4590166516254247,-2.9901786190132946,-0.8500000578323601,-23.403572173513943,995.72
-1.2011969089508057,1.4333007335662842,10.085747718811035,0.1718057248838476 5,-1.192866205762445,-0.342542424466887 1,-3.106250047856953,-0.9312500358840529,-23.206250508331 8,995.7209
-0.442672193050384 5,2.0171494483947754,10.767702102661133,0.376445412635803 2,-1.0320560932159424,-0.20586143434047 7,-3.208223757087679,-0.9651315964908672,-22.981579127774374,995.720
-0.232103794813156 1,1.816152334213256 8,10.468599319458008,0.4097036121985168 6,-1.382624585062853 4,-0.0962451418148364 8,-3.24572378708278,-0.943421065964551 6,-22.869078987595685,995.7
0.502492725849151 6,1.619940876960754 4,9.387042999267578,0.4220312089766881 7,-1.729281857128062 6,-0.002061668947065715,-3.263487012750679 6,-0.827960550863669 3,-22.81578987526757,995.7
1.093519926071167,1.667797327041626,8.796015739440918,0.3181587505616217,-2.2129086264849804,0.04484765486406735,-3.279276546703231,-0.7253289817191643,-22.768421775343768,995.713623
0.705882668495178 2,2.021935224533081,8.516056060791016,0.2695440948009491,-2.123671293258667,0.040317077189683 9,-3.711684838783291 7,-0.838043536886679 3,-22.51141307653400 5,995.706759
```

Etape 4: Statistiques avec code executable cpp

Fichier bat d'exécution dans cmd

```
@echo off
g++ -o C:\STAGE_2024\test_cpp_solo\great\mon_executable C:\STAGE_2024\test_cpp_solo\great\main.cpp Sensors.cpp C:\STAGE_2024\test_cpp_sol
C:\STAGE_2024\test_cpp_solo\great\mon_executable C:\STAGE_2024\data_collect\code_python\output_data\txt\escalator.txt C:\STAGE_2024\data
C:\STAGE_2024\test_cpp_solo\great\mon_executable C:\STAGE_2024\data_collect\code_python\output_data\txt\stair.txt C:\STAGE_2024\data_coll
C:\STAGE_2024\test_cpp_solo\great\mon_executable C:\STAGE_2024\data_collect\code_python\output_data\txt\elevator.txt C:\STAGE_2024\data_c
```

Pas baro pour l'instant

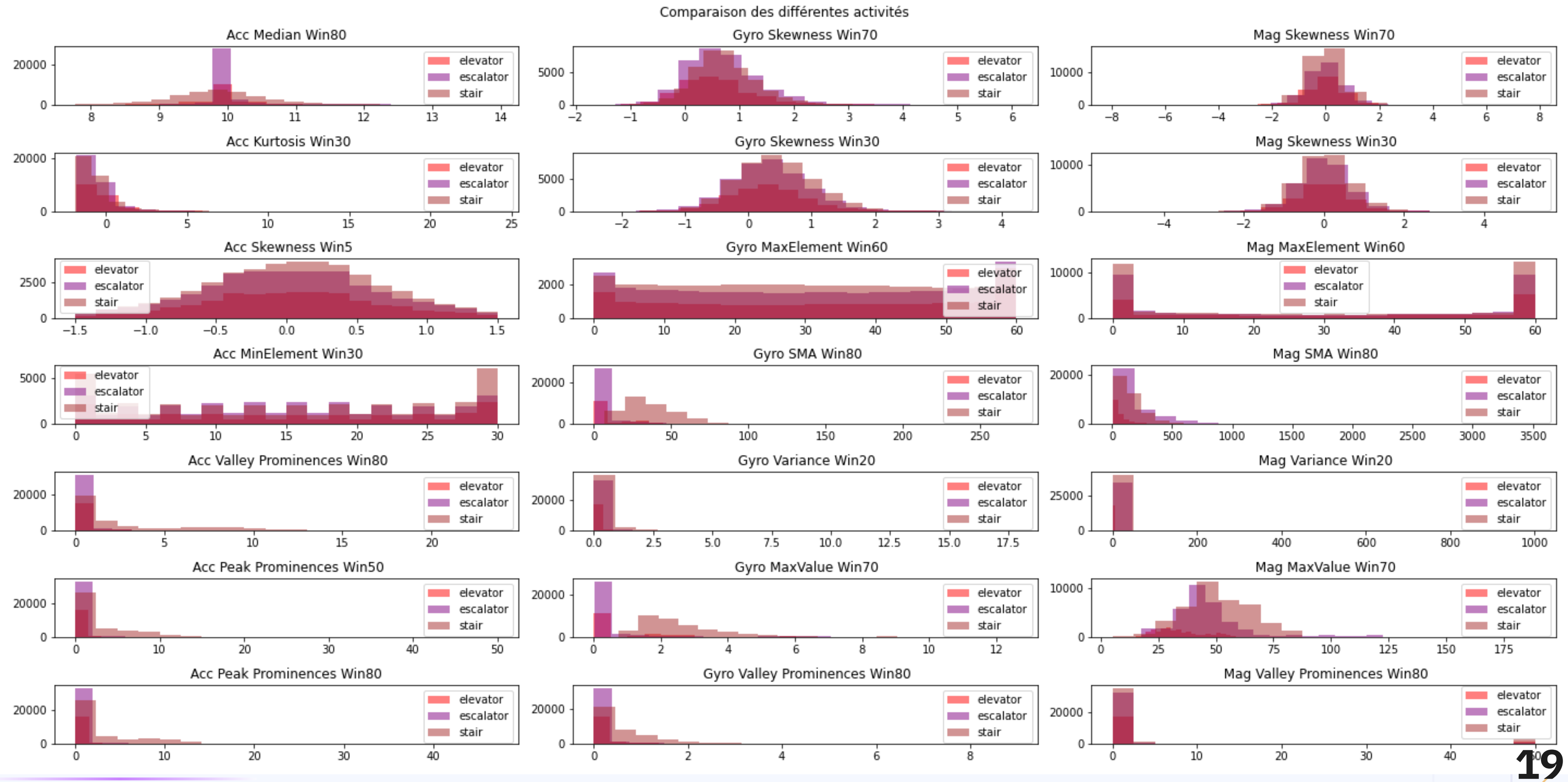
21 caractéristiques dans le domaine temporelle

Echantillon de 129 acquisitions pour faire statistiques

```
features  = featureCalc.computeFeatures(&sensors, ligneCourante);

if (ligneCourante>= 129){

 outputFile << features->get_gyro_skewness_win70() << "; ";
 outputFile << features->get_gyro_skewness_win30() << "; ";
 outputFile << features->get_gyro_maxElement_win60() << "; ";
 outputFile << features->get_gyro_sma_win80() << "; ";
 outputFile << features->get_gyro_variance_win20() << "; ";
 outputFile << features->get_gyro_maxValue_win70() << "; ";
 outputFile << features->get_gyro_valley_prominences_win80() << "; ";

 outputFile << features->get_acc_median_win80() << "; ";
 outputFile << features->get_acc_kurtosis_win30() << "; ";
 outputFile << features->get_acc_skewness_win5() << "; ";
 outputFile << features->get_acc_minElement_win30() << "; ";
 outputFile << features->get_acc_valley_prominences_win80() << "; ";
 outputFile << features->get_acc_peak_prominences_win50() << "; ";
 outputFile << features->get_acc_peak_prominences_win80() << ";";
 outputFile << features->get_mag_skewness_win70() << "; ";
 outputFile << features->get_mag_skewness_win30() << "; ";
 outputFile << features->get_mag_maxElement_win60() << "; ";
 outputFile << features->get_mag_sma_win80() << "; ";
 outputFile << features->get_mag_variance_win20() << "; ";
 outputFile << features->get_mag_maxValue_win70() << "; ";
 outputFile << features->get_mag_valley_prominences_win80() << "\n ";


}
```

Attention
Dans FeatureCalculator pas de filtrage des données ,ni calcul dans le domaine fréquentiel

Etape 5: Visualisation des statistiques



Etape 6: Création dataset pour entrainer modèle

```python
if 'escalator' in mode:
    numero=0

elif 'stair' in mode:
    numero=1

elif 'elevator' in mode:
    numero=2

elif 'still' in mode:
    numero=3
```

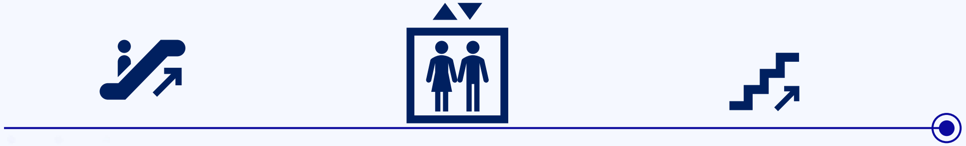
```python
    if data_vstack is not None:
        data_vstack = np.vstack((data_vstack, data))
        label_vstack=np.vstack((label_vstack,label))
        label_numero_vstack=np.vstack((label_numero_vstack,label_numero))
        print(label_numero)
    else:
        data_vstack=data
        label_vstack=label
        label_numero_vstack=label_numero


seed = 42
np.random.seed(seed)
np.random.shuffle(data_vstack)
np.random.seed(seed)
np.random.shuffle(label_vstack)
np.random.seed(seed)
np.random.shuffle(label_numero_vstack)


np.save('../output_data/train_dataset/data.npy',data_vstack)
np.save('../output_data/train_dataset/label.npy',label_vstack)
np.save('../output_data/train_dataset/label_numero.npy',label_numero_vstack)
```

Etape 7: Entrainement modèle

```
Epoch 17/20
605/605 [==============================] - 40s 66ms/step - loss: 0.0644 - accuracy: 0.9769 - val_loss: 0.0827 - val_accuracy:
0.9711
Epoch 18/20
605/605 [==============================] - 43s 71ms/step - loss: 0.0604 - accuracy: 0.9775 - val_loss: 0.1150 - val_accuracy:
0.9604
Epoch 19/20
605/605 [==============================] - 45s 74ms/step - loss: 0.0646 - accuracy: 0.9755 - val_loss: 2.2078 - val_accuracy:
0.6311
Epoch 20/20
605/605 [==============================] - 36s 60ms/step - loss: 0.0533 - accuracy: 0.9807 - val_loss: 0.2470 - val_accuracy:
0.9223
591/591 [==============================] - 7s 11ms/step - loss: 0.2475 - accuracy: 0.9242
```

sans categorie « still » classification en 3 classes

04 Mise en place du modèle dans l'application



```
JNIEXPORT jdoubleArray JNICALL
Java_fr_univeiffel_geolocimu_ImuData_InferenceVertical(JNIEnv *env, jobject,
                                                       jdouble Accx, jdouble Accy,
                                                       jdouble Accz, jdouble Gyrx,
                                                       jdouble Gyry, jdouble Gyrz,
                                                       jdouble Magx,
                                                       jdouble Magy,
                                                       jdouble Magz,
                                                       jint iterator) {
    Sensors* vectorSensors= new Sensors();
    // set values of acceleration
    vectorSensors->set_acc_x(Accx);
    vectorSensors->set_acc_y(Accy);
    vectorSensors->set_acc_z(Accz);

    // set values of gyroscope
    vectorSensors->set_gyro_x(Gyrx);
    vectorSensors->set_gyro_y(Gyry);
    vectorSensors->set_gyro_z(Gyrz);

    vectorSensors->set_mag_x(Magx);
    vectorSensors->set_mag_y(Magy);
    vectorSensors->set_mag_z(Magz);

    std::vector<double> result = smartvertical->Inference(vectorSensors, iterator);

    jdoubleArray resultArray = env->NewDoubleArray(result.size());
    env->SetDoubleArrayRegion(resultArray, 0, result.size(), result.data());

    return resultArray;
```

+Ajout capture du magnetometre et du baromètre dans cpp

+Ajout classe :

FeatureExtractorWithoutFilter
SmartVertical
VerticalModel
VerticalProbabilities

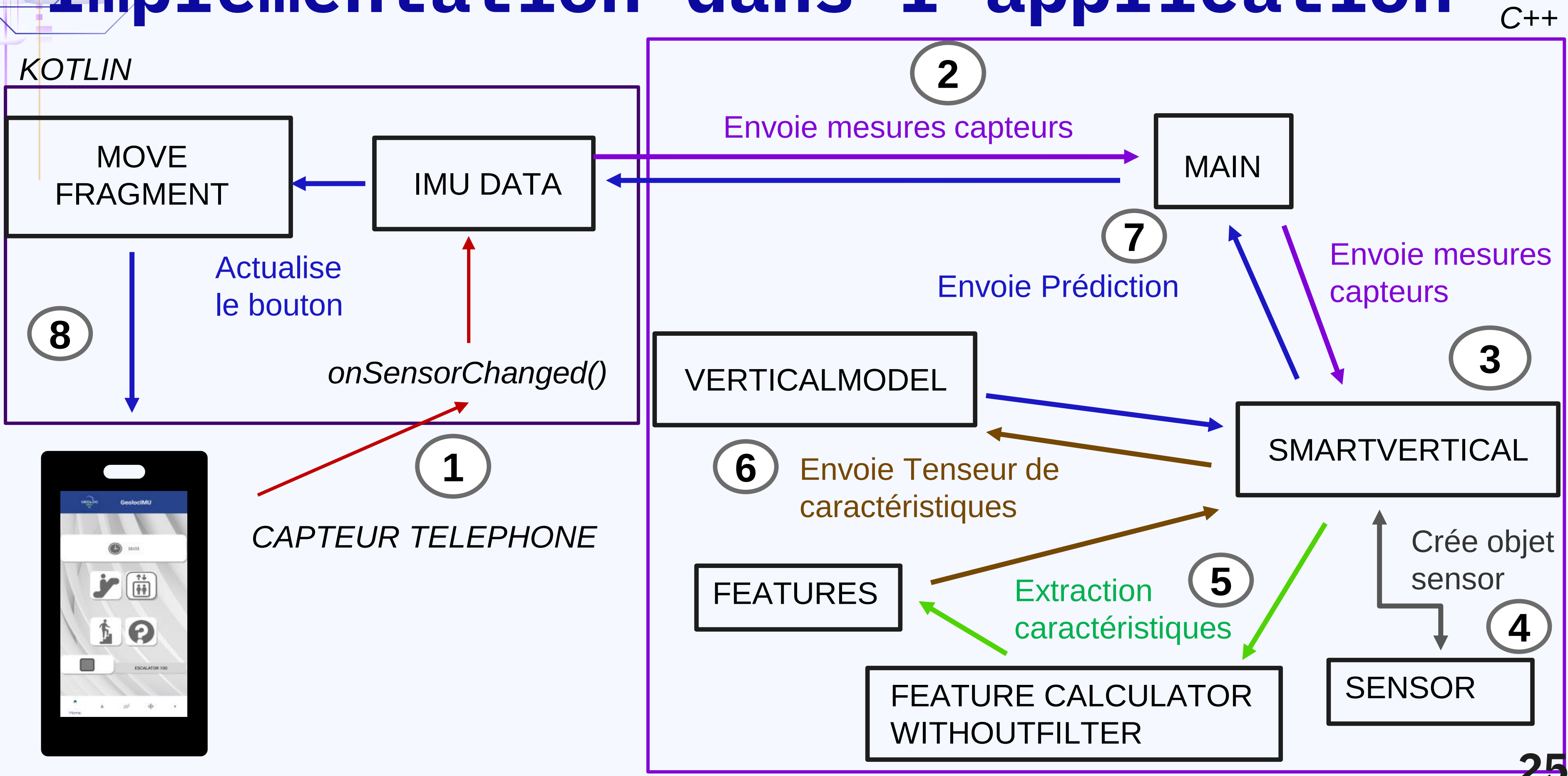
+Ajout Fonction pour déterminer
Montée ou descente avec le baromètre
- BarometerCalculator
- InferenceBarometer

- AccelerometerModel.cpp
- AccelerometerModel.hpp
- AnomalyFinder.cpp
- AnomalyModel.cpp
- AnomalyProbabilities.hpp
- Features.cpp
- Features.hpp
- FeaturesCalculator.cpp
- FeaturesCalculator.hpp
- FeaturesCalculatorWithoutFilter.cpp
- FeaturesCalculatorWithoutFilter.hpp
- FFT.cpp
- FFT.hpp
- Filter.cpp
- Filter.hpp
- GyroscopeModel.cpp
- GyroscopeModel.hpp
- PeakFinder.cpp
- PeakFinder.hpp
- Sensors.cpp
- Sensors.hpp
- Smartstep.cpp
- Smartstep.hpp
- SmartVertical.cpp
- SmartVertical.hpp
- StepProbabilities.cpp
- StepProbabilities.hpp
- VerticalModel.cpp
- VerticalModel.hpp
- VerticalProbabilities.cpp
- VerticalProbabilities.hpp

Ajout texte pour affichage dans VerticalFragment

+affiche seulement si la probabilité estimé est superieur à 90 pourcent

Implémentation dans l’application



05 La suite …

1. Corrigé les retours du compte-rendu du mardi 06 05 2024

2. Lancer la collecte de données officiellement

3.Traiter les données de la collecte

4. Entrainer plusieurs types de classifieurs pour pouvoir les comparer et avoir le meilleur modèle

5. Optimiser les caractéristiques pour trouver la meilleure combinaison

Question pour la suite :

+ Comment intégrer les filtrages dans mon cpp executable
+ Mettre option pour sélection de la position du téléphone poche lors de la collecte des données
+ Ajout fonctionnalité de détection du nombre d’étage monté ou descendu?
+ Ajuster un algorithme PDR avec le floor positionning ?

FIN
merci de votre attention

Bibliographie

[1] E. Gorgone et al. “SemanticSLAM : Using Environment Landmarks for Unsupervised Indoor Localization”. In : (2015). doi : 10.1109/TMC.2015.2478451.

[2] Sara Khalifa, Mahbub Hassan et Aruna Seneviratne. “Feature Selection for Floor-changing Activity Recognition in Multi-Floor Pedestrian Navigation”. In : (1er jan. 2014). doi : 10.1109/ICMU.2014.6799049.

[3] Chenxiang Lin et Yoan Shin. “Deep Learning-Based Multifloor Indoor Tracking Scheme Using Smartphone Sensors”. In : *IEEE Access* 10 (2022), p. 82641-82652. issn : 2169-3536. doi : 10.1109/ACCESS.2022.3183072.

[4] Wenting Zhang, Xiaohui Zhao et Zan Li. “A Comprehensive Study of Smartphone-Based Indoor Activity Recognition via XGBoost”. In : *IEEE Access* 7 (2019), p. 80027-80042. doi : 10.1109/ACCESS.2019.2922974.