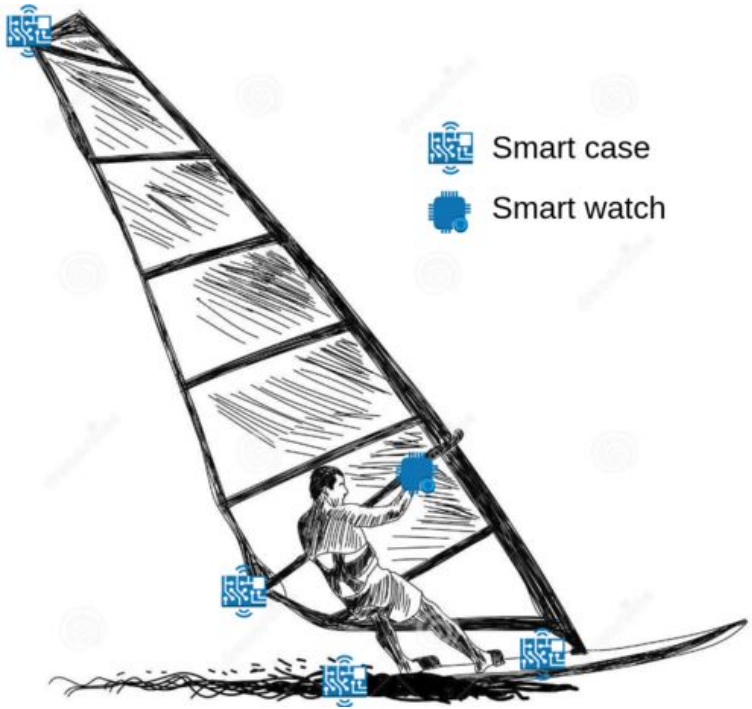
SmartMove

Anthony GEOURJON
Timothée LEMAIRE
Clément ROUQUIER
Vincent TURRIN

Rappel du sujet/besoins et cahier des charges

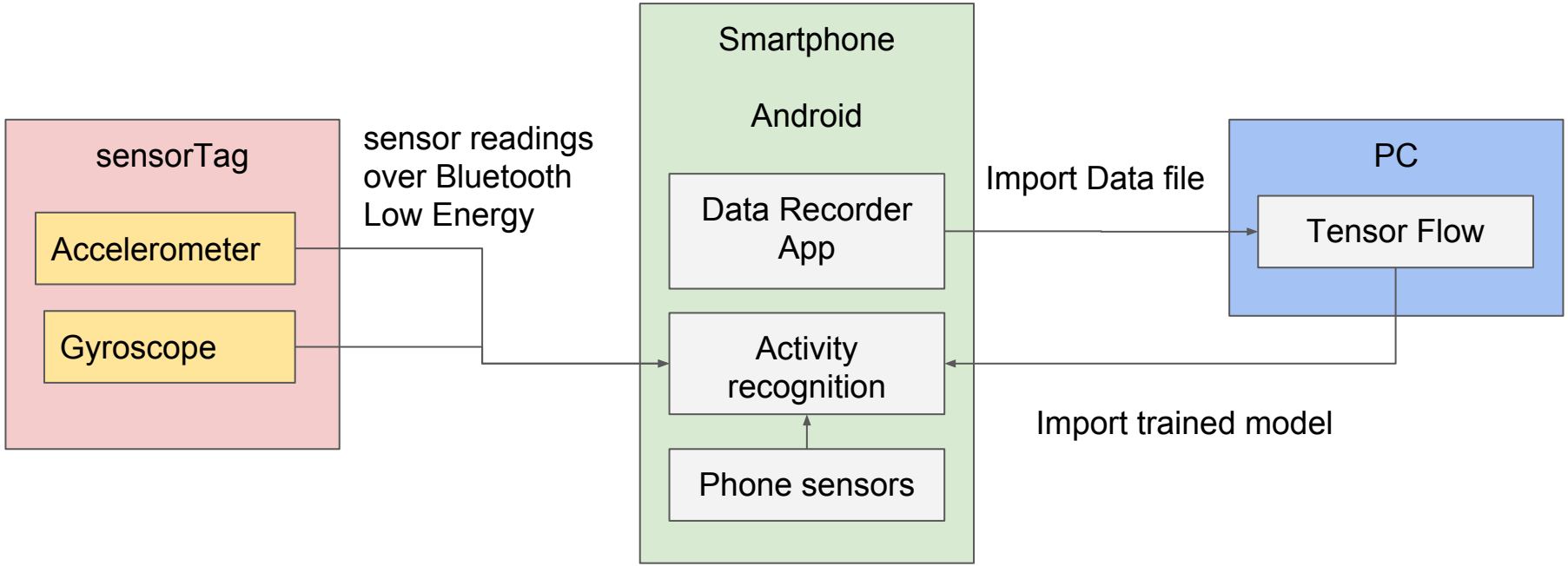
- Utiliser des accéléromètres et des gyroscopes pour enregistrer des mouvements
- Utiliser le machine learning pour séparer ces mouvements
- Proposer de la valeur ajoutée grâce aux mouvements reconnus
- Proposer des solutions à l’utilisateur quand il effectue mal des mouvements



Technologies employées

- SensorTag (Abandon d’Arduino)
  - matériel de développement, BLE, capteurs précis.
- Python : librairie Tensorflow
  - écosystème mature, communauté, utilisable sur mobile
- Android
  - OS open-source, communauté forte

Architecture techniques



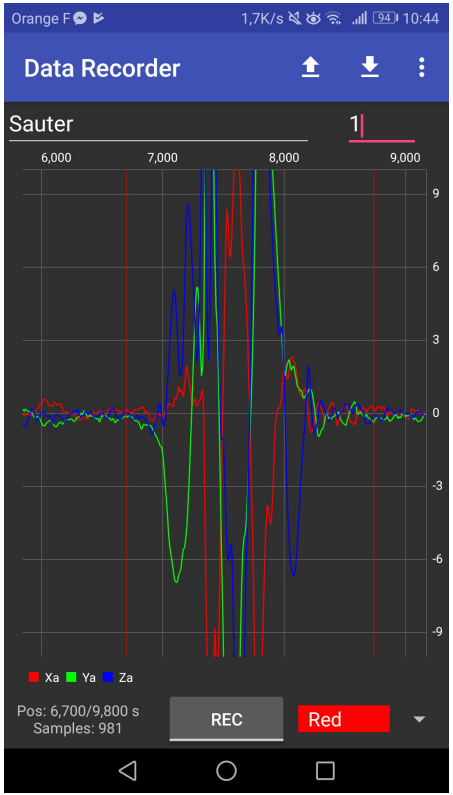
Réalisations techniques

Sensors :

Téléphone	SensorTag
+ Plus simple + Bonne fréquence de lecture (100 Hz) - Tous les téléphones n’ont pas Gyroscope	+ Facilité de placement - Capteur supplémentaire - Faible fréquence d’envoi (10 Hz)

Réalisations techniques

- Applis Android
  - Récupération **données**
  - Enregistrement et **labellisation** des données



Données d’un **mouvement soudain**

Données sur la **durée**

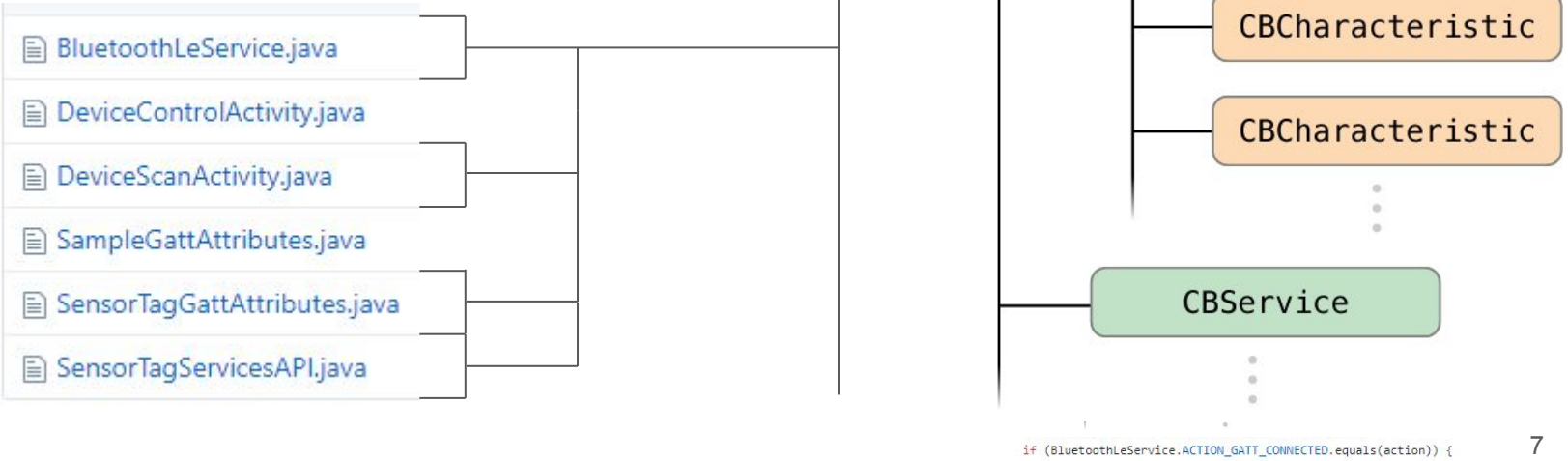
Réalisations techniques

- Communication Sensor / Smartphone avec BLE GATT
  - Basé sur le GoogleSample BleGATT
  - Système Master (smartphone) / Slave (SensorTag)

```
public boolean connect(final String address) {
final BluetoothDevice device = mBluetoothAdapter.getRemoteDevice(address);
mBluetoothGatt = device.connectGatt(this, false, mGattCallback);
public void onConnectionStateChange(BluetoothGatt gatt, int status, int newState) {
```

```
mBluetoothGatt.discoverServices());
public void onServicesDiscovered(BluetoothGatt gatt, int status) {
```

```
SensorTagServicesAPI.setAccelerometerPeriod(mBluetoothGatt,(byte)10);
```




```
if (BluetoothLeService.ACTION_GATT_CONNECTED.equals(action)) {
```

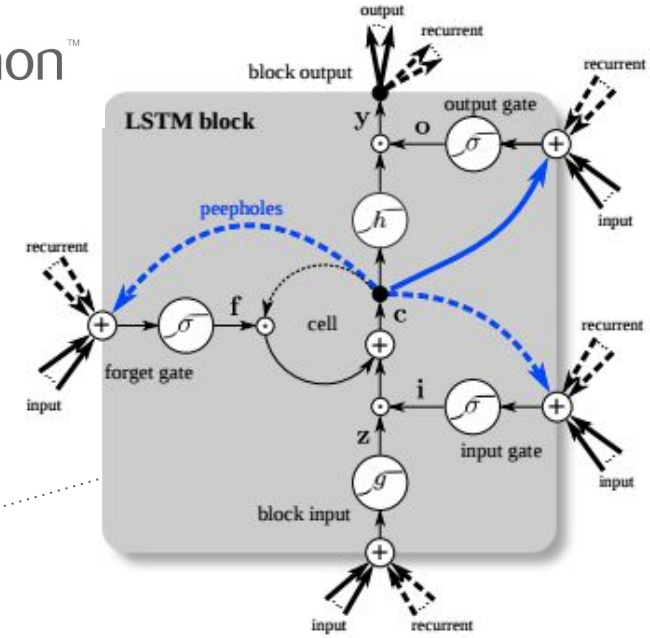
Processus Learning

data : .log
label, **user**, **acc-x**, **acc-y**, **acc-z**

Formatage

Création du modèle

Long short-term memory LSTM



Apprentissage à partir des data / Validation avec jeu de test

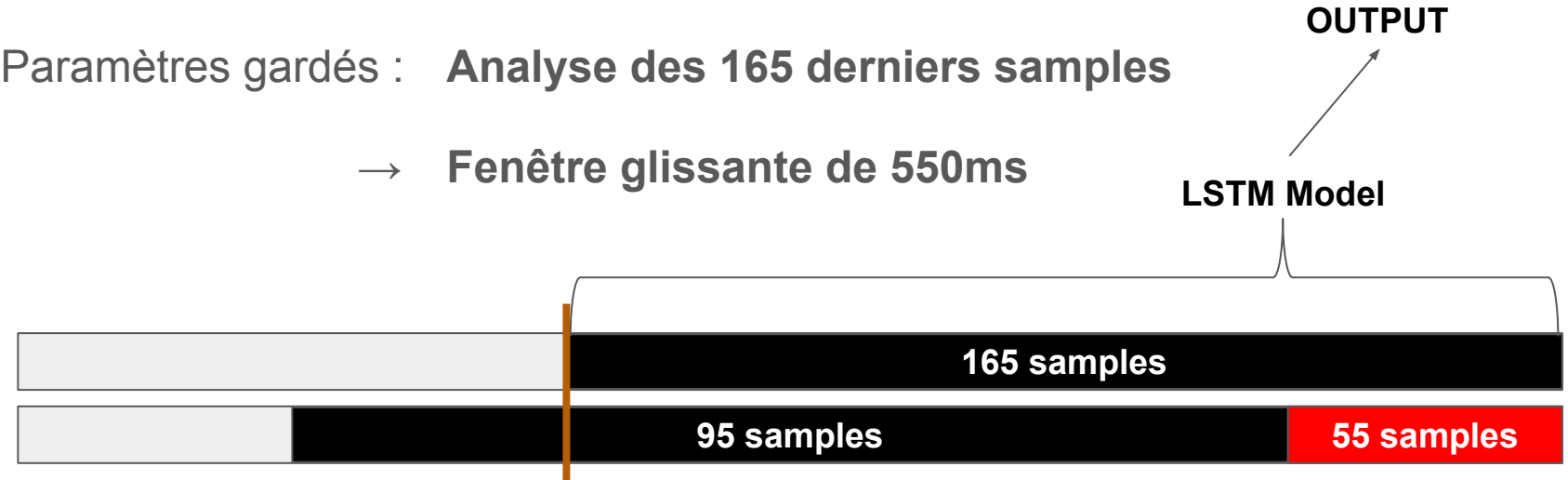
export du modèle pour Android

Réalisations techniques

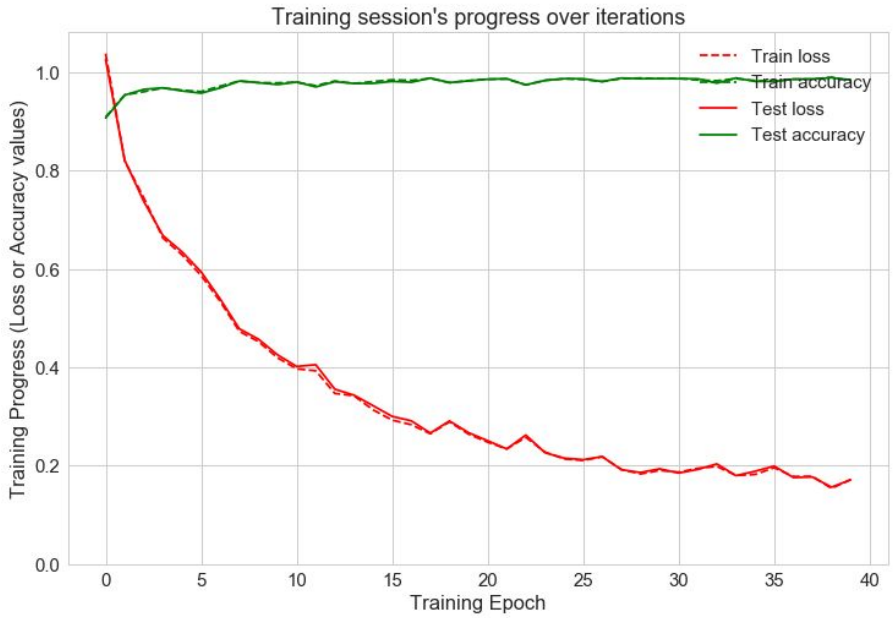
- Nombreux tests pour les paramètres optimaux
  - Accéléromètre/Gyroscope à la jambe
  - Accéléromètre/Gyroscope à la ceinture

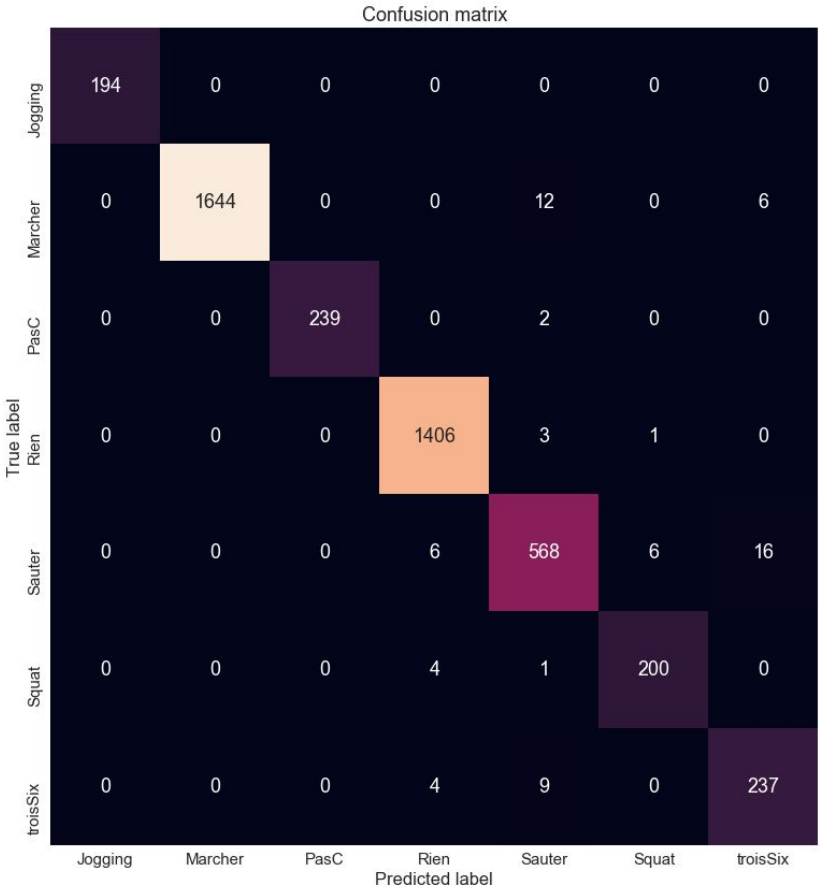
Paramètres gardés : **Analyse des 165 derniers samples**

→ **Fenêtre glissante de 550ms**



Réalisations techniques



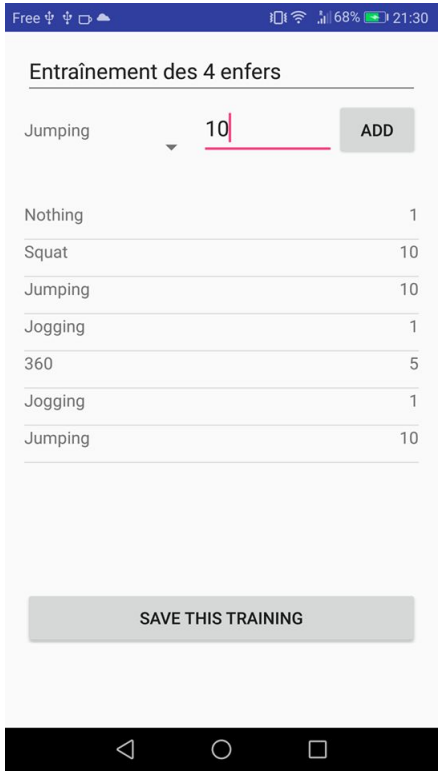


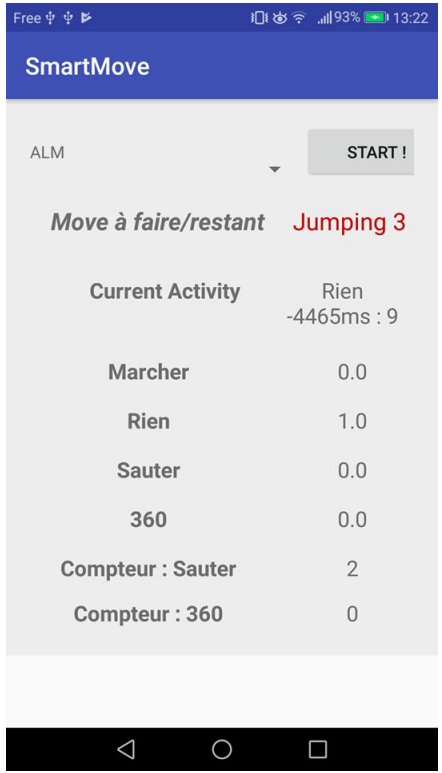
Précision finale : 98.5%

Avec 10Mo de data : **20 min de calcul** sur **GTX 960**

Réalisations techniques

- Reconnaissance du mouvement
- Compteur de mouvements spécifiques
- Création d'entraînement
- Réalisation d'entraînement





Gestion de projet : organisation du travail

- Clément et Timothée : partie bluetooth, SensorTag/Arduino
- Vincent : Machine learning
- Anthony : Gestion des entrainements

- Gestion des tâches avec Trello
  - Différents états : À faire, en cours, déployable
  - Tags : Android, machine learning, Bluetooth
  - Affectable aux membres

Gestion de projet : Temps ingénieur, Budget, etc

- Projet de 6 semaines, 4*6*25 = 600 heures ingénieur
- 1 chef de projet 3750 €, 3 développeurs 2500 € brut
- Coût salarial = 1.5 * (1.5 * (3750 + 3 * 2500)) = <u>25 312,5 €</u>
- Coût d’amortissement du matériel négligeable

- 2329 lignes de code (en plus des 500 lignes du projet que nous avons forké)

Conclusion (Retour d'expérience)

- Deep Learning grosse zone d’inconnu au départ
- Beaucoup d’apprentissage sur le protocole BLE et son implémentation sur Android (GATT).
- Impressionnés de la précision de la reconnaissance avec le peu de data fournies

Démonstration

- Création d’une session d’entraînement
- Réalisation de l’entraînement (reconnaissance des mouvements)
- Exemple d’enregistrement de data

Perspectives d’améliorations

- Affiner le modèle
- Récolter plus de data pour entraîner le modèle plus finement
- Résoudre les problématiques de l’utilisation des SensorTag (rafraîchissement, capteur multiple)
- Sauvegarder les entraînements en ligne

Smart Move

Merci pour votre attention.