
"The importance of the road infrastructure for the society could be compared with importance of blood vessels for human"

-A Wise man.

Table des matières

Introduction générale	4
1 Problématique	6
2 Anomalies dans une route	6
2.1 les nids-de-poule	6
2.2 Les fissures	7
2.3 Les Ralentisseurs (longs et courts)	8
3 Objectifs et analyse des besoins	9
4 Avantages	9
5 Solution proposée	9
1 La détection mobile «Mobile sensing»	11
1.1 Différents Capteurs	13
1.2 Capteurs d'un smartphone	13
1.2.1 Accéléromètre	13
1.2.2 Gyroscope	13
1.2.3 GPS	14
1.3 Applications utilisant la détection mobile.	14
1.3.1 Galaxy Watch	15
1.3.2 Uber	15
1.3.3 Samsung Health	16
2 État de l'art	18
2.1 Travaux Connexes	19
2.1.1 Wolverine	19
2.1.2 Real time pothole detection using Android smartphones with accelerometer	20
2.2 L'acquisition des données	25
2.2.1 Positionnement du smartphone	25
2.2.2 Méthode de récolte de données	26
2.3 Notre Approche	26
2.3.1 Windows split and Filtering	26

TABLE DES MATIÈRES

2.3.2	Lissage et valeur des seuils	27
2.3.3	Z Score	28
3	Développement	29
3.1	Architecture du système	30
3.1.1	Aperçu de système	30
3.1.2	Fonctionnement de système	31
3.2	Technologies utilisées	32
3.2.1	Python	32
3.2.2	REST API	32
3.2.3	Flask	33
3.2.4	Flutter	33
3.3	Application de collecte	34
3.4	Server	36
3.5	Conclusion	36

Introduction générale



TABLE DES MATIÈRES

Selon le centre national de prévention et de sécurité routière algérien : plus de 3.049 personnes avaient trouvé la mort et 29.095 personnes avaient été blessées dans 21.109 accidents enregistrés au niveau national lors des onze premiers mois de l'année 2019 [17]. Dans un scénario réel, la règle dit que



Figure 1 – Accident de route

plus la surface de la route est confrontée à des changements de climat (entre le froid et le chaud) et avec un manque de soins, plus elle subit des dégâts et affecte ensuite la vie des gens.

La surveillance de la surface des routes est un problème difficile dans le domaine des infrastructures de transport routier dans le monde entier. Une zone en mauvais état peut endommager les véhicules, les conducteurs et même provoquer un accident. Ils sont également à l'origine de poursuites et de dommages-intérêts coûteux, par exemple, en 2005, l'État du Michigan avait déposé plus de 7,500 réclamations pour dommages liés aux nids-de-poule¹, et les compagnies d'assurance recevaient plus de 500,000 réclamations pour nids-de-poule chaque année².

Les municipalités du monde entier dépensent des millions de dollars pour entretenir et réparer leurs routes. Malgré cet investissement, peu de gens sont satisfaits de la qualité de la route où ils vivent ou travaillent, car c'est toujours trop tard !!

1. <http://www.detnews.com/2005/specialreport/0510/18/A01-350197.htm>

2. <http://www.wktv.com/special/6733696.html>

1. PROBLÉMATIQUE

1 Problématique

De ce qui précède, l'entretien et la maintenance de la qualité de l'infrastructure routière est d'une importance vitale pour la société.

En Algérie, particulièrement, ce domaine est presque négligé, les routes endommagées sont plus nombreuses que les routes sûres. L'entretien de la route en Algérie passe par plusieurs phases :

- Détection des dégradations.
- Localisation des dégradations : les agents du service de contrôle de la route utilisent les points kilométriques 'PK' pour localiser les dégradations.
- L'entretien de la route se fait soit par la Direction des Travaux Publics 'DTP' ou une entreprise désignée par le Ministère des Travaux Publics 'MTP'.

La détection des dégradations jusqu'à présent se fait par constatation visuelle par des agents d'entretien de la DTP, cette étape est fastidieuse et longue nécessitant un savoir-faire particulier et une expérience dans le domaine. Par conséquent les dégradations (nids-de-poule/bosses) prennent beaucoup de temps pour être réparées et par conséquent elles causeront plus de dégâts.

2 Anomalies dans une route

Dans nos recherches lors de l'étude de cette problématique, nous observons que les anomalies les plus courantes sont les nids-de-poule (potholes) et les bosses (bumps). Ces deux catégories peuvent classer la plupart des anomalies trouvées dans notre vie quotidienne.

2.1 les nids-de-poule

Les nids-de-poule sont principalement causés par une mauvaise qualité de la chaussée ou des problèmes sous la surface.

Un nid-de-poule est une dépression dans la surface d'une route, généralement une chaussée asphaltée, où la circulation a enlevé des morceaux cassés de la chaussée. C'est généralement le résultat de l'eau dans la structure du sol sous-jacente et du trafic passant sur la zone affectée. L'eau affaiblit d'abord le sol sous-jacent ; le trafic fatigue et brise la surface asphaltée mal supportée de la zone touchée. L'action continue de la circulation éjecte l'asphalte et le sol sous-jacent pour créer un trou dans la chaussée.³

3. <https://en.wikipedia.org/wiki/Pothole>

2. ANOMALIES DANS UNE ROUTE



Figure 2 – Nids-de-poule

2.2 Les fissures

La fissuration est une dégradation majeure qui touche l'ensemble des chaussées avec différentes origines et différents mécanismes de développement. Elle est à la source d'une accélération des dégradations propres à chaque type de chaussée par la diminution de la portance du support lors de l'infiltration d'eau et par la perte des conditions mécaniques nécessaires au maintien de la résistance des matériaux [1].



Figure 3 – Les fissures

Par conséquent la fissuration est la cause directe d'une perte de valeur du patrimoine routier.

2. ANOMALIES DANS UNE ROUTE

Causes probables :

- Mauvaise construction du joint longitudinal entre deux bandes d'enrobés.
- Fatigue de la chaussée due à une structure insuffisante vis-à-vis du trafic ou une portance, du sol support insuffisante.
- Les caractéristiques du sol : tassement, retrait du sol argileux à la suite d'une longue période de sécheresse (Assèchement).

2.3 Les Ralentisseurs (longs et courts)

Normalement fabriquées par l'homme et généralement utilisées pour ralentir les véhicules à proximité des passages pour piétons. On constate le



Figure 4 – Ralentisseurs

plus souvent qu'ils imposent une limite de vitesse basse, inférieure à 40 km/h ou moins poussant les conducteurs à freiner d'avance. Les ralentisseurs en général ne sont pas des anomalies, mais les ralentisseurs mal faits ou ceux qui ont été exposés à des dégâts sans réparation sont considérés comme des anomalies. De plus, leur utilisation est parfois controversée car :

- Ils peuvent augmenter le bruit de la circulation.
- Peuvent endommager les véhicules s'ils sont traversés à une vitesse trop élevée et ralentir les véhicules d'urgence.
- Les ralentisseurs mal conçus qui se tiennent trop haut ou avec un angle trop aigu peuvent perturber les conducteurs et peuvent être difficiles à naviguer pour les véhicules à faible garde au sol, même à très basse vitesse. De nombreuses voitures de sport ont ce problème avec de tels ralentisseurs.
- Les ralentisseurs peuvent également présenter de graves dangers pour les motocyclistes et les cyclistes s'ils ne sont pas clairement visibles.

Les ralentisseurs coûtent entre 50 et 200 Dollar et peuvent devoir être remplacés au fil du temps en raison de l'usure [14].

3. OBJECTIFS ET ANALYSE DES BESOINS

3 Objectifs et analyse des besoins

Détecter les anomalies d'une route présente un vrai défi. Avant de commencer le projet, nous sommes amenés à répondre aux questions suivantes :

- Quels sont les différents caractéristiques d'une anomalie ?
- Quels sont les différentes difficultés et critères à prendre en compte pour décider si une anomalie est présente ?
- Peut-on offrir une solution informatique évoluée pour assister à cette décision ? Si oui :
 - Comment mesurer le niveau de qualité d'une route ?
 - Comment détecter/révéler les différentes causes d'une mauvaise route ?
 - Comment localiser géographiquement chaque anomalie ?
 - Comment présenter cette solution aux services concernés de manière simple et efficace ?

4 Avantages

Développer un outil de détection automatique de qualité de la route aurait pour apport :

- Un gain considérable de temps, efforts humains et une réduction de dépenses.
- Une surface de route solide et surtout sans danger pour les piétons et les conducteurs.
- Une meilleure circulation en ville en diminuant le nombre d'accidents de route et ainsi réduire les dégâts sur les véhicules.

5 Solution proposée

Nous proposons un outil permettant de :

- Fournir à la direction des travaux public "DTP" un moyen de se renseigner en temps réel sur la qualité des routes afin d'intervenir efficacement et le plus rapidement possible dans les travaux d'entretien. Ceci permettra d'améliorer les manques cités plus haut.
- Mieux assister le conducteur dans sa conduite en l'alertant des ralentisseurs/dégradations de la route, afin qu'il adapte sa conduite en fonction.

Pour cela, nous visons à développer un système qui proposera les fonctionnalités suivantes :

5. SOLUTION PROPOSÉE

- Déetecter les événements (les nids-de-poule et les ralentisseurs dans notre cas) en temps réel. La collecte de données brutes pour un post-traitement hors ligne est classée comme une fonctionnalité supplémentaire.
- Utiliser un smartphone basé sur Android-OS avec des capteurs accéléromètres comme plate-forme matérielle / logicielle. La portabilité vers d'autres plates-formes est classée comme une fonctionnalité supplémentaire.
- Pouvoir fonctionner sur différents modèles de smartphones avec différents paramètres. Au cours du processus de mise en œuvre du système, l'ensemble des paramètres minimaux du smartphone doit être déterminé et décrit.
- Déetecter les événements lors de la conduite dans différents types de véhicules à quatre roues tels que des voitures particulières, des mini-fourgonnettes et des bus. Les véhicules à deux roues tels que les motos et les scooters ne sont pas pris en compte.

Chapitre 1

La détection mobile «Mobile sensing»



1.1. DIFFÉRENTS CAPTEURS

Récemment, des petits capteurs de haute performance se sont répandus et sont intégrés dans divers types d'objets dans notre environnement.

La détection mobile "Mobile sensing" utilise des capteurs intégrés dans des objets en mouvement tels que des voitures, des vélos et des smartphones, et les considère comme des capteurs de détection [18].

De plus, le taux de propagation des smartphones de haute qualité augmente et continuera de le faire. Ces derniers incluent de différents types de capteurs tels que des capteurs d'accélération et des capteurs gyroscopiques.

1.1 Différents Capteurs

Le capteur est un appareil qui détecte les changements dans l'environnement proche et envoie ces données au système d'exploitation ou au processeur. Ils détectent et collectent les données pour lesquelles ils sont faits.

Il existe trois catégories principales de capteurs que possède un smartphone [20] :

Les capteurs de mouvements : ils mesurent les forces d'accélération et les rotations autour des trois axes. Ces capteurs sont capables de déterminer dans quelle direction est orienté l'appareil. A titre d'exemple, On trouve l'accéléromètre et le gyroscope.

Les capteurs de position et d'attitude : Ce genre de capteurs détermine la position et l'orientation de l'appareil. On trouve donc les capteurs d'orientation, le gyroscope et le magnétomètre ainsi que le GPS.

Les capteurs environnementaux : c'est des capteurs qui mesurent la pression atmosphérique, l'illumination et la température ambiante (Baromètre, photomètre et thermomètre).

1.2 Capteurs d'un smartphone

1.2.1 Accéléromètre

L'accéléromètre détecte les changements d'orientation des smartphones par rapport aux 3 axes orthogonaux (X, Y et Z). Cette accélération est utilisée pour déterminer la vitesse de l'appareil.

1.2.2 Gyroscope

Le gyroscope ou capteur gyroscopique est une version avancée de l'accéléromètre. Alors que l'accéléromètre détecte la détection de mouvement basée

1.3. APPLICATIONS UTILISANT LA DÉTECTION MOBILE.

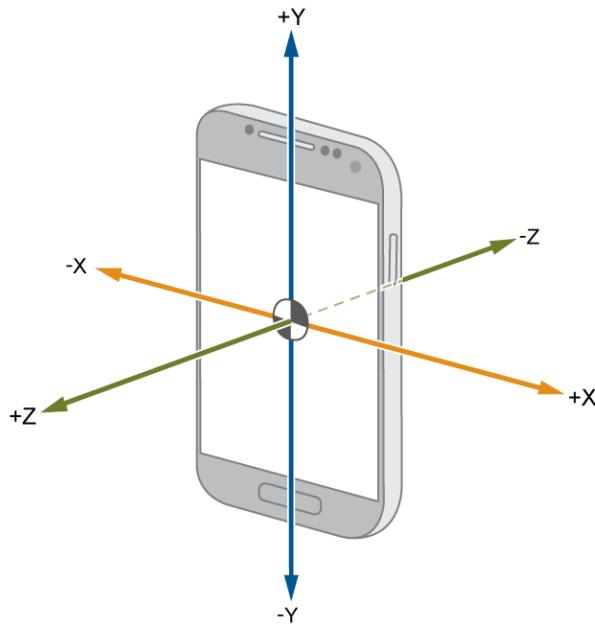


Figure 1.1 – Axes orthogonaux utilisés

sur l'axe, le gyroscope fonctionne avec l'accéléromètre et détecte chaque degré de changement d'orientation. Il fournit une détection de mouvement très précise.

1.2.3 GPS

Le GPS ou le système de positionnement global est également très courant dans la plupart des téléphones modernes. Il aide à localiser l'emplacement sur Terre et aide à la navigation.

1.3 Applications utilisant la détection mobile.

Avec l'évolution des smartphones, le monde a commencé à tirer de plus en plus d'avantages de leurs capacités pour développer de nombreuses applications mobiles. Dont certaines faisant usage des capteurs du smartphone pour réaliser et automatiser des tâches qui étaient avant impossibles sans des appareils sophistiqués.

1.3. APPLICATIONS UTILISANT LA DÉTECTION MOBILE.

1.3.1 Galaxy Watch

Galaxy Watch est une montre intelligente développée par Samsung Electronics. Elle est déclinée en version 41 et 45 mm et embarque une multitude de capteurs : Accéléromètre, baromètre, capteur gyroscopique, capteur cardiaque électrique (ECG), capteur de fréquence cardiaque optique et un capteur de lumière.

Le capteur de fréquence cardiaque mesure votre fréquence cardiaque en battements par minute à l'aide d'une source de lumière LED optique et d'un capteur de lumière LED. La lumière brille à travers votre peau et le capteur mesure la quantité de lumière réfléchie. Les reflets de la lumière varient au fur et à mesure que le sang pulsera sous votre peau au-delà de la lumière.



Figure 1.2 – Galaxy Watch

1.3.2 Uber

Uber est un service de voiture à la demande qui vous permet de demander les services d'un chauffeur privé grâce à une application fonctionnant sur iPhone et Android. Ce service utilise un logiciel qui permet d'envoyer au chauffeur le plus proche votre localisation.

La cartographie est un élément clé du service de VTC (Voiture de transport avec chauffeur). Les opérateurs locaux d'Uber doivent en effet pouvoir visualiser en temps réel les chauffeurs qui circulent dans leur ville sans avoir à passer par des requêtes SQL fastidieuses.

Reposant sur le principe du géocodage inversé, la géolocalisation de l'adresse du points de prélèvement (position du client) est calculée par Uber en se basant sur la position GPS du smartphone du client. Côté VTC, un moteur de prédiction (basé sur des technologies de machine learning) oriente le chauffeur vers une destination plutôt qu'une autre en fonction notamment d'un

1.3. APPLICATIONS UTILISANT LA DÉTECTION MOBILE.

historique clients. Enfin, Uber utilise aussi le suivi par GPS pour détecter les comportements dangereux au volant (notamment les excès de vitesse) et améliorer ainsi la sécurité routière.

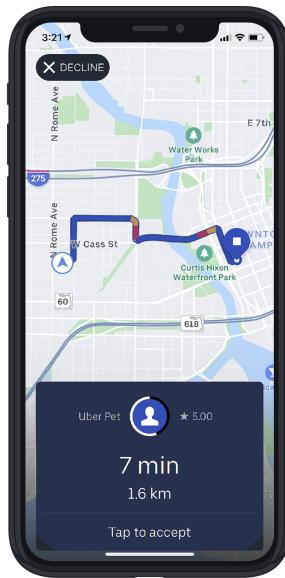


Figure 1.3 – Uber

1.3.3 Samsung Health

Samsung Health, à l'origine S Health, est une application gratuite développée par Samsung qui sert à suivre différents aspects de la vie quotidienne comme l'activité physique, l'alimentation et le sommeil.

Certaines fonctionnalités sont suivies par les entrées de l'utilisateur (nourriture / calories, poids, quantité d'eau, etc ...), et d'autres fonctionnalités sont suivies par des tests avec des accessoires de téléphone (Fitbit, Galaxy Active, Galaxy Fit, etc.) ou avec des capteurs de téléphone. L'accéléromètre par exemple mesure l'accélération dans de nombreux cas, par exemple : quand l'utilisateur fait du sport ou conduit sa voiture...

Comment l'application mesure-t-elle les pas ? Le smartphone détecte en permanence les mouvements du corps sur un accéléromètre à 3 axes. Les données sont enregistrées tout le temps qu'elles sont portées et allumées, ce qui permet au traqueur de suivre si l'individu marche en avant, court vite ou même immobile.

1.3. APPLICATIONS UTILISANT LA DÉTECTION MOBILE.

Les données collectées sont exécutées via un algorithme personnalisé. Cela permet au logiciel de détecter ce qu’impliquent réellement les différents mouvements enregistrés.

L’application permet à l’individu de savoir combien de pas ont été faits, à quelle vitesse, le rythme de l’individu, et même le nombre de calories susceptibles d’avoir été brûlées. L’application permet à l’individu d’interagir avec les informations d’une manière conviviale.



Figure 1.4 – Samsung Health

Chapitre 2

État de l'art



2.1. TRAVAUX CONNEXES

Avec cette évolution dans le domaine des mobiles, il est possible de développer un système pratique et efficace à faible coût capable de recueillir divers types d'informations afin de détecter la qualité de la surface des routes et aussi les anomalies routières telles que les ralentisseurs et les nids de-poule.

2.1 Travaux Connexes

2.1.1 Wolverine

Wolverine [15] est une méthode qui utilise des données de capteurs des smartphones pour déterminer les conditions et l'état de la route, et utilise aussi des techniques d'apprentissage automatique “Machine learning” (clustering K-means et Support Vector Machine (SVM)) pour la surveillance et suivi de l'état de la route. Le processus consiste en deux parties : un algorithme pour réorienter virtuellement les axes de coordonnées d'un téléphone désorienté et techniques d'apprentissage automatique pour identifier les événements de bosse et de freinage.

Pour la 1ere étape, ils ont développé une application qui traite les lectures de l'accéléromètre et du magnétomètre ainsi que les informations GPS, puis produit l'accélération linéaire du mobile, dans le cadre de référence du véhicule. Ils ont développé un algorithme qui étudie les axes (x, y, z) du téléphone et étudie aussi les axes (x', y', z') du véhicule.

La 2ème étape consistera à utiliser un algorithme d'apprentissage non supervisé de clustering K-means, avec $K = 2$, pour partitionner l'ensemble des points de données en deux classes ; Ces classes sont ensuite étiquetées manuellement comme bosselées ou lisses (Bumpy ; smoothy).

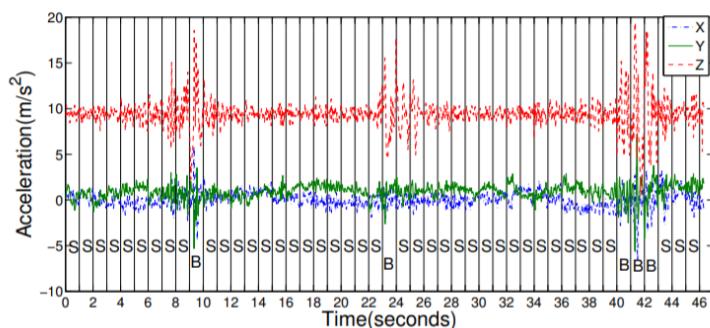


Figure 2.1 – Accelerometer Data for three speedbreakers

Une fois cet étiquetage manuel terminé, ils auront un ensemble de points

2.1. TRAVAUX CONNEXES

de données étiquetés. Ceux-ci sont ensuite utilisés pour former un classificateur Support Vector Machine. Ce SVM entraîné, à son tour, est utilisé pour classer les points de données qui sont générés pendant la phase de test, et donc pour prédire l'état du véhicule (Figure 2.1).

Et pour la détection du freinage du véhicule, la technique utilisée est identique à celle de la détection des bosses, avec deux classes aussi étiquetées doux et freinage (smooth 'S', braking 'R') (Figure 2.2).

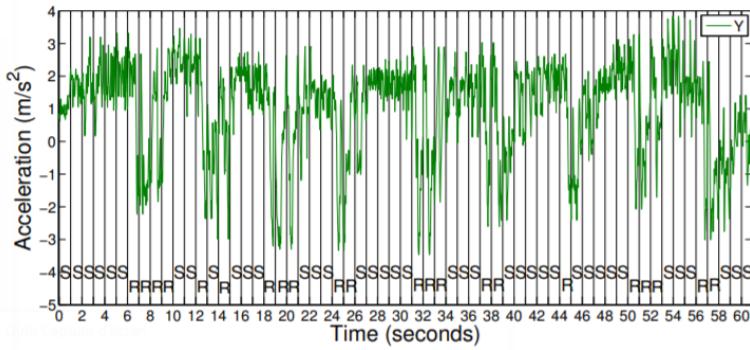


Figure 2.2 – Braking events with generated class labels

Ce travail a été fait à Mumbai. L'algorithme identifie correctement 18 des 20 événements de bosse. Aucun tronçon de route lisse n'est identifié à tort comme une bosse. Ainsi, ils obtiennent un taux de faux négatifs de 10% pour la détection des bosses et un taux de faux négatifs de 21,6% et un taux de faux positifs de 2,7% pour la détection de freinage.

2.1.2 Real time pothole detection using Android smart-phones with accelerometer

Mednis et al. ont proposé un système de détection des nids-de-poule en temps réel basés sur des données d'accéléromètre pour le déploiement sur des appareils avec des ressources matérielles / logicielles limitées et leur évaluation sur des données du monde réel acquises à l'aide de différents smartphones basés sur Androïd OS.

Ils ont utilisé un dispositif de collier LynxNet modifié [5] sur une route avec divers nids-de-poule pour la collecte des données des capteurs de l'accéléromètre.

Après l'acquisition du premier jeu de données de test, une recherche de fonctionnalités liées aux événements potentiels a été effectuée : Le premier et le plus simple algorithme de détection d'événements ZTHRESH a été testé

2.1. TRAVAUX CONNEXES

sur l'ensemble des données acquises . Il est similaire à l'algorithme z-peak utilisé dans les systèmes Pothole Patrol [7], Nericell [16], et limite l'amplitude de l'accélération sur l'axe Z. Les valeurs dépassant des seuils spécifiques sont utilisés pour identifier et classifier les nids de poules (Figure 2.3).

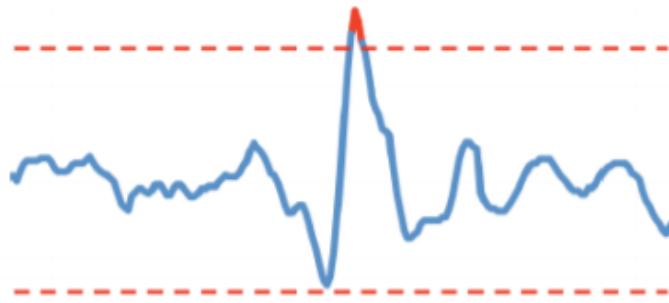


Figure 2.3 – Pothole detection algorithm Z-THRESH. Les évènements sont représentés par des mesures avec des valeurs dépassant un seuil spécifié.

Pour la réorientation virtuelle, ils ont utilisé une approche plus simple : un placement contrôlé de l'accéléromètre, éliminant le traitement supplémentaire.

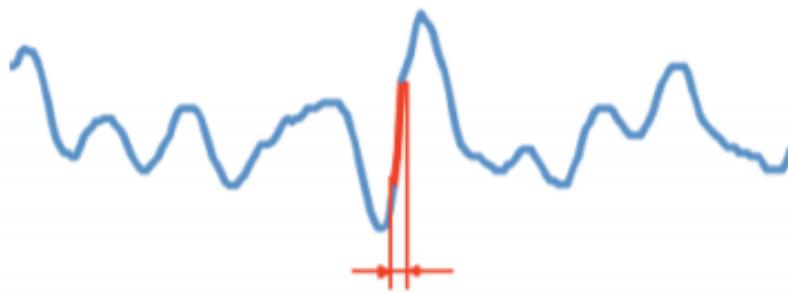


Figure 2.4 – Pothole detection algorithm Z-DIFF.

Ensuite, un algorithme légèrement plus avancé était le Z-DIFF (Figure 2.4) testé sur l'ensemble de données acquis. Contrairement à Z-THRESH, une recherche de deux mesures consécutives avec une différence entre les valeurs au-dessus du niveau de seuil spécifique a été effectuée. Ainsi, l'algorithme a détecté des changements rapides dans les données d'accélération verticale. L'algorithme nécessite la détermination de la position de l'axe Z de manière similaire à l'approche précédente.

2.1. TRAVAUX CONNEXES

Les auteurs ont décidé de mettre en œuvre certaines des techniques utilisées pour le post-traitement. Une technique prometteuse pour la mise en œuvre sur un appareil à ressources limitées utilisait un écart type de l'accélération de l'axe vertical. Il a été implémenté dans l'algorithme STDEV (Z) (Figure 2.5).

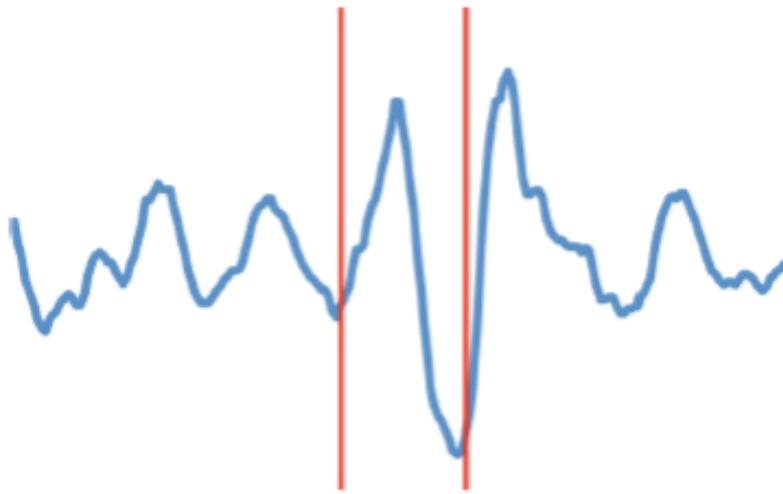


Figure 2.5 – Pothole detection algorithm STDEV(Z).

En utilisant des outils d'analyse visuelle des données et en recherchant des modèles de données spécifiques, les auteurs ont constaté qu'il existait certains événements caractérisés par un tuple de mesure spécifique. Toutes les données à trois axes de ce tuple avaient des valeurs proches de 0g. ils ont donc nommé cet algorithme G-ZERO (Figure 2.6).

Les algorithmes (Z-THRESH., Z-DIFF, STDEV(Z) et G-ZERO) ont détecté des irrégularités sur la route principale pour 100% des grands nids-de-poule et 83 à 90% des grappes de nids-de-poule. Selon l'algorithme, 78 à 89% pour les petits nids-de-poule ont été détectés. Il est à noter que 9 d'entre eux (50%) ont été détectés par les 4 algorithmes pour chacune des 10 sessions d'essai sur la route.

En gros, les tests d'évaluation ont abouti à une configuration optimale pour chaque algorithme sélectionné et l'analyse des performances dans le contexte de différentes classes d'irrégularité routière montre des taux de vrais positifs pouvant atteindre 90%.

Notre algorithme pour la détection de l'état de la route se distingue des travaux connexes de plusieurs côtés :

2.1. TRAVAUX CONNEXES

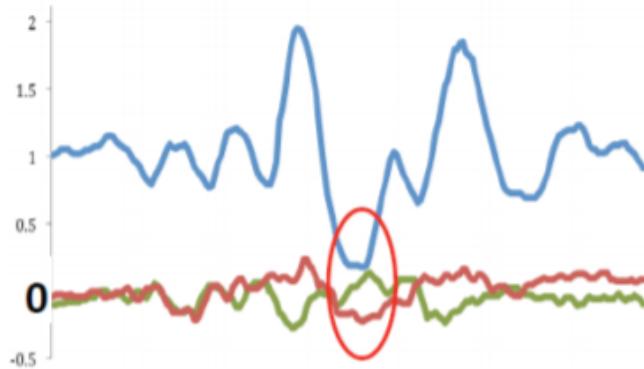


Figure 2.6 – Pothole detection algorithm G-ZERO.

- Nous ne détectons pas d'anomalies spécifiques, par contre nous concluons une évaluation de l'état de la route en général.
- Notre solution sera plus simple en utilisant des ressources matérielles et logicielles limitées.
- Nous visons à réaliser un prototype rapide, en nous assurant que le système et algorithmes soient extensibles ou facilement remplaçables à tout moment pour des améliorations dans le futur.

2.1. TRAVAUX CONNEXES

Parameters	Wolverine	Mednis
Capteurs	Accelerometre, Magnetometre, GPS	Accelerometre, GPS
Methodse pour la reorientation	Reorientation utilisant magnetometre et GPS bearing	Non défini
Valeur / Axes pris en compte	Z-axis Stdev sur une fenêtre de 1 sec du l'accéléromètre	Valeurs de l'axe Z de l'accéléromètre
Techniques utilisées	Machine Learning utilisé pour détecter les bosses et le freinage	Z-Thresh, ZDiff, STDEV, G-Zero
Taux d'échantillonnage réel de l'accéléromètre	50	100
Lieu d'expérimentation	IIT Bombay Campus	Vairoga iela, Riga, Latvia
Taux d'échantillonnage pris en compte par seconde	50 values	100 values
Équipement utilisé pour la collecte de données	Capteurs de smartphone (Google Nexus S, HTC Wildfire S)	Samsung i5700 Samsung Galaxy S HTC Desire HTC HD2
Véhicules utilisés	Suzuki Access 125, Autorickshaw	BMW 323 Touring (4-Wheeler)
Distance parcourue (km) / temps de trajet (heures)	Non reporté	174 KM / Quelques semaines
Points de montage	Non défini (placé dans une certaine orientation arbitraire)	Tableau de bord avant
Objectif de la détection	Bosses, freinage	Nids de poule
Seuil 'threshold' utilisé	Non reporté	$Z\text{-Thresh} = 0.4g$ $Z\text{-Diff} = 0.2g$ $STDEV = 0.2g$ $G\text{-Zero} = 0.8g$
Output	FN - 10% (2 roues) FP - 8% (3 roues)	90% des vrais positifs
Consommation d'énergie	Consomme 58% moins d'énergie que Nericel	Non reporté

Table 2.1 – Comparaison technique des travaux existants

2.2 L'acquisition des données

Notre système dépend de l'accéléromètre du smartphone attaché à un véhicule, produisant des résultats pour une condition de route donnée et sur la localisation précise de ces résultats à partir du GPS de ce smartphone. Dans cette section, nous décrivons quelques expériences effectuées par Pothole Patrol [6] pour valider le fonctionnement des capteurs d'un smartphone et la meilleure position où l'attacher.

2.2.1 Positionnement du smartphone

Une préoccupation est que le placement du smartphone à l'intérieur du véhicule pourrait affecter la qualité du signal de l'accéléromètre. Pothole Patrol ont placé leurs accéléromètres à deux endroits à l'intérieur de la cabine d'une seule voiture. La (Figure 2.7) montre le signal de l'accéléromètre pour un tronçon fixe de chaussée à partir de deux positions de montage différentes : fixé au tableau de bord, et fixé sur le côté droit du pare-brise. Les signaux du tableau de bord et du pare-brise semblent assez similaires.

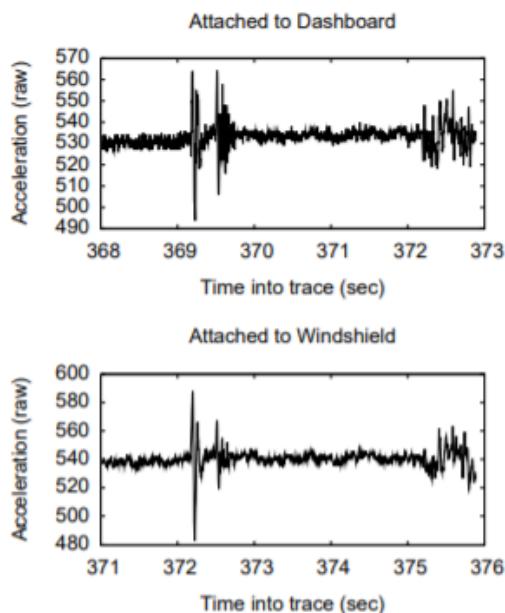


Figure 2.7 – Comment le signal de l'accéléromètre (axe z) varie avec le placement à l'intérieur de la voiture selon Pothole Patrol.

Ainsi, ils ont fermement fixé l'accéléromètre au tableau de bord à l'intérieur de la boîte à gants de la voiture, qui est un endroit relativement facile

2.3. NOTRE APPROCHE

pour installer des capteurs et qui les maintient hors de portée des passagers dans la cabine. Il en va de même pour les smartphones.

2.2.2 Méthode de récolte de données

Pour la collecte de données, nous avons créé une application mobile qui utilise les capteurs du smartphone. Principalement l'accéléromètre et le GPS, ainsi que la vitesse du véhicule et des données d'autres capteurs tels que le capteur gyroscopique.

Ceci sera discuté dans la section 3.3.

2.3 Notre Approche

Notre approche utilisera principalement les données de l'accéléromètre comme entrées (input). Ce Dernier signale l'accélération du véhicule le long des trois axes du capteur.

L'accélération mesurée comprend à la fois l'accélération physique du véhicule ainsi que celle de la gravité. La mesure est signalée dans les champs Accel_X, Accel_Y et Accel_Z.

En s'inspirant des travaux précédemment cités, considérer uniquement l'axe Z du capteur, en respectant le positionnement du téléphone donné, semble être une méthode à la fois simple et qui donne de bons résultats.

Dans cette section, nous décrivons l'algorithme que nous avons développé pour détecter les conditions routières dans un enregistrement de données de capteurs.

Le problème de l'identification des conditions routières à partir des données de l'accéléromètre est difficile en raison de plusieurs facteurs tels que le comportement du conducteur (virages, dérapages, freinage brusque, conduite très basse, etc.) ou la grande variation et les conditions routières étranges ici en Algérie, donc notre algorithme suit 3 étapes principales :

2.3.1 Windows split and Filtering

Cette étape est divisée en deux parties :

- **Windows split** : Les données provenant de la partie d'enregistrement varient en taille, ils peuvent être des enregistrements d'une minute de conduite comme cela peut l'être pendant 1 heure de conduite, notre objectif principal de ce système est de détecter l'état de la route tout au long du chemin, et donc on doit savoir l'état de chaque petite

2.3. NOTRE APPROCHE

partie de la route enregistrée. L'algorithme proposé divise les données en plusieurs intervalles (fenêtres) en fonction de la distance.

On calcule la distance entre le 1er point et le 100 ème, en utilisant la longitude et la latitude de chacun, si la distance calculée est comprise entre 90 et 110 mètres on va utiliser cette fenêtre, sinon on augmente ou diminue l'intervalle de points choisi.

Pour faire cela nous avons utilisé la formule de Haversine :

Nous convertissons d'abord le long et le lat de chaque point en radians :

Valeur de la latitude en radians, $\text{lat} = \text{Latitude} / (180/\pi)$.

Valeur de la longitude en radians, $\text{long} = \text{Longitude} / (180/\pi)$.

Avec : $(180/\pi) = 57.29577951$.

Ensuite, pour obtenir la distance entre le point A et le point B, nous utilisons la formule suivante :

Distance, $d = 3963.0 * \arccos[(\sin(\text{latA}) * \sin(\text{latB})) + \cos(\text{latA}) * \cos(\text{latB}) * \cos(\text{longB} - \text{longA})]$.

Puis nous convertissons la distance obtenue de miles en km :

d in kilometers = $1.609344 * d$ in miles.

d in metres = d in kilometers * 1000.

- **Speed filtering :** Une fois les données divisées en fenêtres, nous devons filtrer les fenêtres inutiles avant le traitement. Dans cette partie, nous filtrons chaque fenêtre selon la vitesse du véhicule. Les fenêtres dans lesquelles le véhicule ne se déplace pas ou se déplace très lentement sont ignorées (nous avons utilisé 20 m/s comme seuil de vitesse).

2.3.2 Lissage et valeur des seuils

Une fois que l'algorithme a obtenu les données filtrées, il commence par les lisser pour éliminer les vibrations normales, puis calcule la valeur moyenne de toutes les données afin de l'utiliser comme référence pour détecter les valeurs anormales.

La méthode de lissage que nous avons utilisée consiste à calculer la moyenne des 2 valeurs avant et après chaque point concerné. Le moment où la valeur moyenne est détectée et en nous inspirant de nombreux articles de recherche dans ce domaine, nous avons choisi 0,85 g comme valeur de sensibilité afin de calculer le seuil.

Nous avons calculé les valeurs de seuil comme suit :

Up Threshold = Average * SensitivityValue.

DownThreshold = Average * 1/SensitivityValue.

Tel que :

Average : La valeur moyenne de toutes les données de la fenêtre.

SensitivityValue : 0.85 g.

2.3. NOTRE APPROCHE

2.3.3 Z Score

Une fois les données prêtes à être traitées, et que les valeurs seuil sont calculées. On s'intéresse aux points dépassant ces valeurs de seuil qui représentent probablement des anomalies dans la route.

On compte ces valeurs pour calculer le score de qualité de ce morceau de route.

Chapitre 3

Développement



3.1. ARCHITECTURE DU SYSTÈME

Après avoir expliqué notre approche dans le chapitre précédent, dans ce chapitre nous expliquerons en détail comment le système proposé met en œuvre l'approche et comment elle fonctionne.

3.1 Architecture du système

Notre système se compose de plusieurs parties ; une application mobile qui recueille des données, un serveur où les données en cours de traitement et stocke le résultat du traitement dans une base de données, une copie des données vers le DTP et enfin une carte interactive avec l'état de la route en temps réel pour les conducteurs.

3.1.1 Aperçu de système

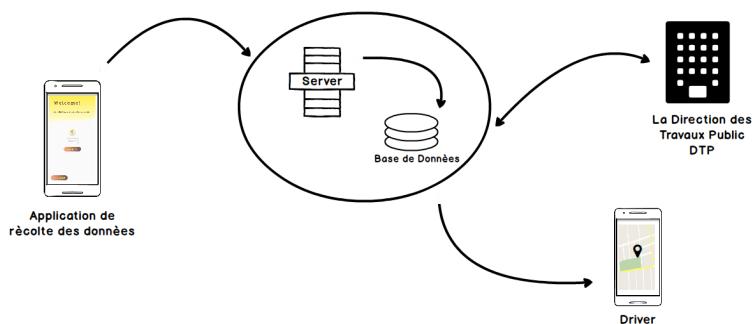


Figure 3.1 – Un aperçu du système proposé.

- **Application de récolte des données** : Une application mobile où les capteurs du smartphone sont utilisés pour collecter les données du journal de conduite et envoyer les données au serveur.
- **Server** : Après avoir reçu les données, le serveur estime l'état de la route et enregistre les résultats dans une base de données.
- **Base de données** : Il contient les résultats estimés de l'état de la route et l'enregistre pour être desservi par d'autres parties.
- **DTP** : La Direction des Travaux Publics ; la partie principale qui utilise les résultats de la base de données. Nous pouvons étendre ce système avec une application de tableau de bord pour eux, qui :
 - Leur permettre de surveiller régulièrement l'état de la route pour prendre plus rapidement des décisions d'entretien et de réparation.
 - Envoyer les données fixes sur l'état de la route à la base de données après la réparation.

3.1. ARCHITECTURE DU SYSTÈME

- **Driver :** Nous pouvons également étendre ce système avec une application mobile, qui fournit au conducteur moyen une carte interactive montrant les conditions routières actuelles pour une meilleure expérience de conduite.

3.1.2 Fonctionnement de système

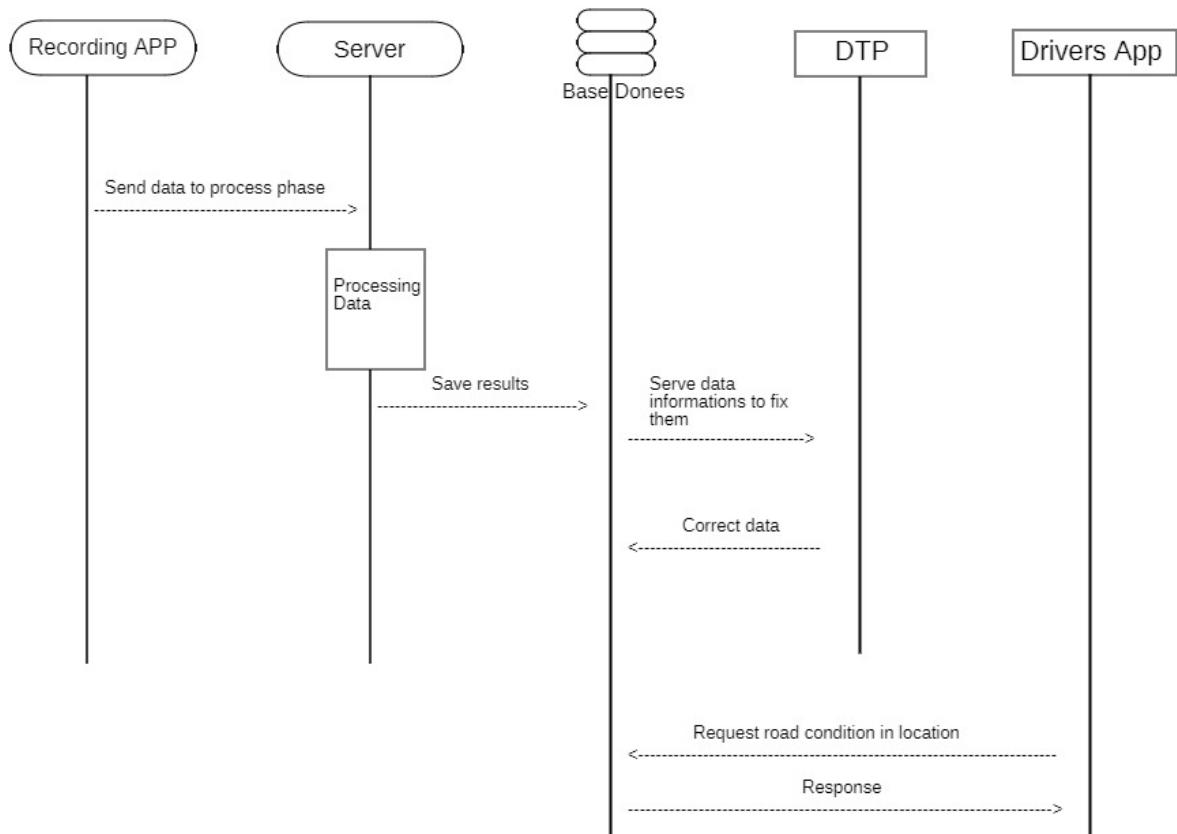


Figure 3.2 – Diagramme de séquence

Notre système sous forme de diagramme montre l'envoi des données du journal de collecte de l'application d'enregistrement au serveur sous forme de fichier CSV, dès qu'il existe ; Le serveur facture l'étape du processus d'enregistrement des résultats dans la base de données.

L'application de tableau de bord DTP utilise la base de données pour corriger les défauts, tandis que l'application Driver l'utilise pour une meilleure expérience du conducteur.

3.2. TECHNOLOGIES UTILISÉES

3.2 Technologies utilisées

3.2.1 Python

Python [10] est un langage de programmation qui peut s'utiliser dans de nombreux contextes et s'adapter à tout type d'utilisation grâce à des bibliothèques spécialisées. Il est cependant particulièrement utilisé comme langage de script pour automatiser des tâches simples mais fastidieuses. On l'utilise également comme langage de développement de prototype lorsqu'on a besoin d'une application fonctionnelle avant de l'optimiser avec un langage de plus bas niveau. Il est particulièrement répandu dans le monde scientifique, et possède de nombreuses bibliothèques optimisées destinées au calcul numérique [8].

Pandas : Pandas [4] est une bibliothèque écrite pour le langage de programmation Python permettant la manipulation et l'analyse des données. Elle propose en particulier des structures de données et des opérations de manipulation de tableaux numériques et de séries temporelles. Pandas est un logiciel libre sous licence BSD.

Les principales structures de données sont les séries (pour stocker des données selon une dimension - grandeur en fonction d'un index), les Data-Frames (pour stocker des données selon 2 dimensions - lignes et colonnes), les Panels (pour représenter des données selon 3 dimensions, les Panels 4D ou les Data Frames avec des index hiérarchiques aussi nommés Multi Index (pour représenter des données selon plus de 3 dimensions - hypercube) [13].



Figure 3.3 – Python and Pandas Logos

3.2.2 REST API

API est une abréviation de : Application Programming Interface (ou interface de programmation d'application, en français). Pour faire simple : c'est un moyen de communication entre deux logiciels, que ce soit entre différents composants d'une application ou entre deux applications différentes.

3.2. TECHNOLOGIES UTILISÉES

REST signifie Representational State Transfer (ou transfert d'état de représentation, en français), et constitue un ensemble de normes, ou de lignes directrices architecturales qui structurent la façon de communiquer les données entre votre application et le reste du monde, ou entre différents composants de votre application.

Nous utilisons l'adjectif RESTful pour décrire les API REST. Toutes les API REST sont un type d'API – mais toutes les API ne sont pas RESTful. Les API RESTful se basent sur le protocole HTTP pour transférer les informations – le même protocole sur lequel la communication web est fondée [3].

3.2.3 Flask

Flask [9] est un framework open-source de développement web en Python. Son but principal est d'être léger, afin de garder la souplesse de la programmation Python, associé à un système de templates. Flask a été créé initialement par Armin Ronacher comme étant un poisson d'Avril. Le souhait de Ronacher était de réaliser un framework web contenu dans un seul fichier Python mais pouvant maintenir des applications très demandées. En 2018, Flask était élu "Framework web le plus populaire" par le Python Developers Survey. En janvier 2020, il cumulait plus de 49000 étoiles sur Github, plus que n'importe quel autre framework de développement web Python[12].



Figure 3.4 – Logo Flask

3.2.4 Flutter

Flutter [2] est un SDK pour applications mobiles permettant de créer des applications hautes performances et haute fidélité pour iOS et Android à partir d'une seule base de code. L'objectif est de permettre aux développeurs de proposer des applications hautes performances qui se sentent naturelles sur différentes plates-formes.

3.3. APPLICATION DE COLLECTE

Comme React Native, Flutter fournit également des vues de style réactif. Flutter adopte une approche différente pour éviter les problèmes de performances causés par la nécessité d'un pont JavaScript en utilisant un langage de programmation compilé, à savoir Dart [19].

- **Dart :**

Dart est un langage de programmation polyvalent développé à l'origine par Google et ensuite approuvé en tant que norme par Ecma (ECMA-408). Il est utilisé pour créer des applications Web, serveur, bureau et mobiles. Dart est un langage basé sur les objets, orienté objet, défini par la classe et utilisant une syntaxe de style C qui transcompile éventuellement en JavaScript. Il prend en charge les interfaces, mixins, classes abstraites, génériques réifiés, typage statique et un système de type sonore [11].

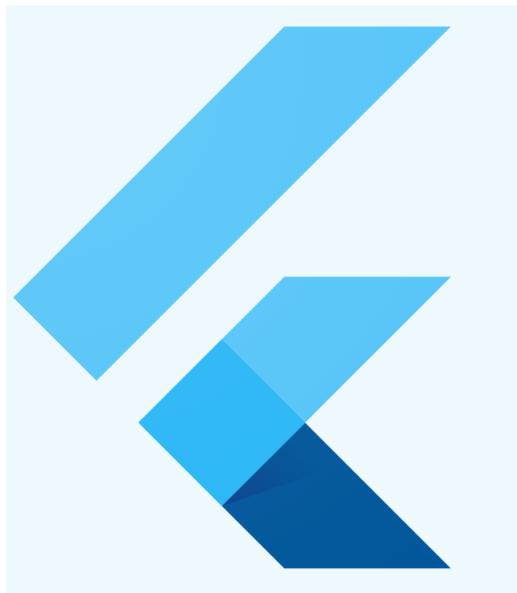


Figure 3.5 – Flutter Logo

3.3 Application de collecte

Nous avons développé une application mobile utilisant Flutter pour la collecte de données à l'aide des capteurs de téléphone.

- D'abord, l'application demande à l'utilisateur d'entrer la durée d'enregistrement en minutes - Montre en capture d'écran «par seconde» à des objectifs du test - (Figure 3.6a).

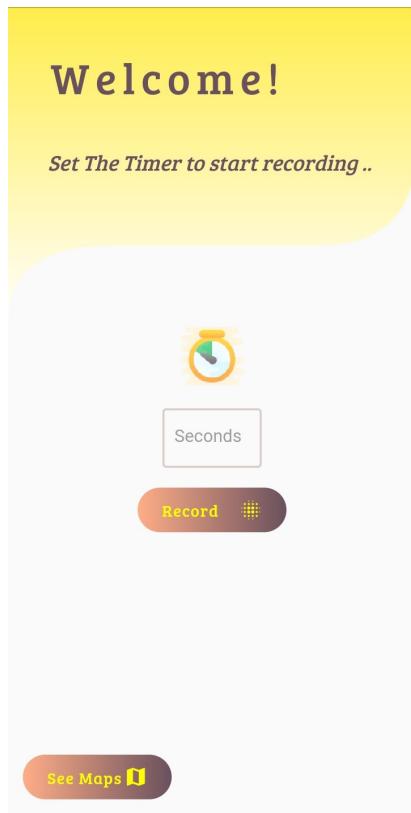
3.3. APPLICATION DE COLLECTE

- Deuxièmement, il lui demande d'activer le GPS (Figure 3.6b). Sinon, il ne démarrera pas l'enregistrement (Figure 3.6c).
- Une fois le GPS activé et l'heure d'enregistrement entrée, l'application amène l'utilisateur à l'écran d'enregistrement, lui montrant une carte en temps réel et un graphique accéléromètre / gyroscope. Il peut annuler l'enregistrement quand il le souhaite (Figure 3.6d).
- Lorsque le temps est écoulé, l'application envoie des données au serveur une fois que l'utilisateur clique sur «Envoyer les données» - elles seront envoyées automatiquement dans les travaux futurs - (Figure 3.7).
L'application les enregistre également sous forme de fichier CSV contenant (Figure 3.8). :
 - La durée d'enregistrement (Time).
 - Données de l'accéléromètre (Accel X, Accel Y, Accel Z).
 - Données du gyroscope (Gyro X, Gyro Y, Gyro Z).
 - Données GPS (Lat, Long).
 - Données de la vitesse du véhicule (Speed).

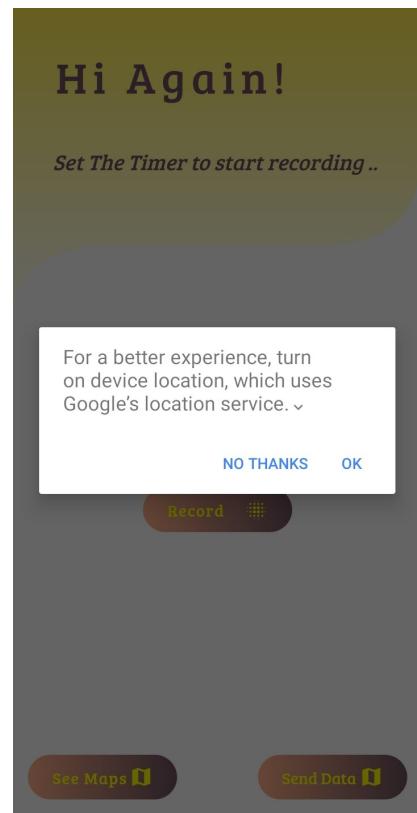
3.4 Server

3.5 Conclusion

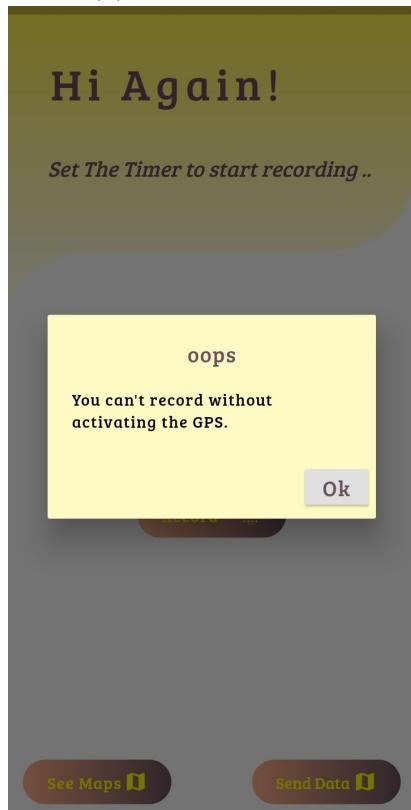
3.5. CONCLUSION



(a) Welcome Screen.



(b) Gps activating alert.



(c) Notification to activate GPS.



(d) Recording Screen.

Figure 3.6 – Application de récolte

3.5. CONCLUSION

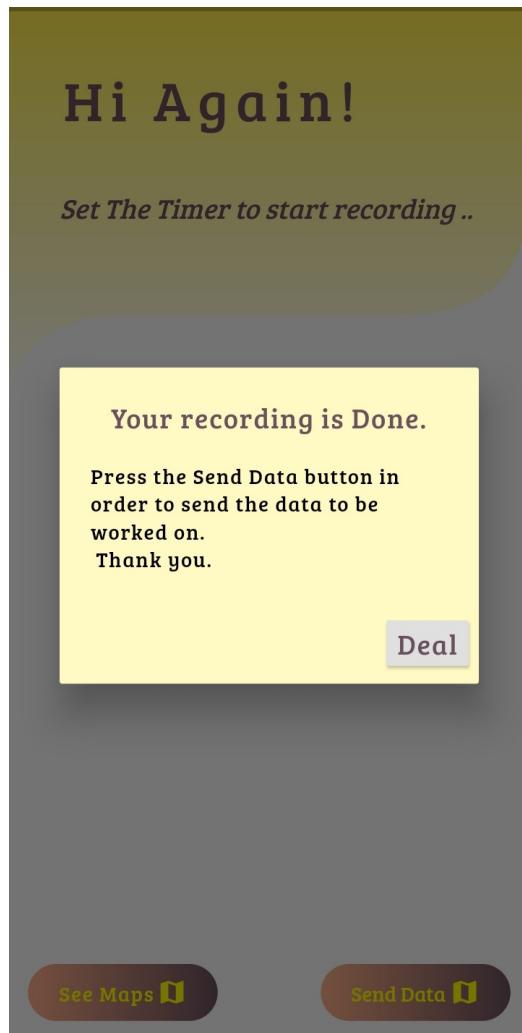


Figure 3.7 – Done Recording.

3.5. CONCLUSION

	A	B	C	D	E	F	G	H	I	J	Q
1	Time	Accel X	Accel Y	Accel Z	Gyro X	Gyro Y	Gyro Z	Lat	Long	Speed	
2	1644	0.37469	3.48955	5.07212	-0.494	-0.2626	-0.249	35.6631	-0.6886	0.99306	
3	1649	0.37469	3.48955	5.07212	0.47421	-0.2211	0.04723	35.6631	-0.6886	0.99306	
4	1650	0.37469	3.48955	5.07212	-0.7182	-0.0476	-0.282	35.6631	-0.6886	0.99306	
5	1650	0.37469	3.48955	5.07212	0.52247	-0.0965	0.10282	35.6631	-0.6886	0.99306	
6	1651	0.37469	3.48955	5.07212	-0.8508	-0.2889	-0.0853	35.6631	-0.6886	0.99306	
7	1652	0.37469	3.48955	5.07212	-0.1996	0.08863	-0.0829	35.6631	-0.6886	0.99306	
8	1652	0.37469	3.48955	5.07212	-0.1086	-0.163	0.01363	35.6631	-0.6886	0.99306	
9	1972	-0.7015	1.03789	10.6554	-1.2435	0.1375	-0.282	35.6631	-0.6886	0.99306	
10	2027	-0.1173	2.62764	7.65187	0.31417	-0.1698	0.04418	35.6631	-0.6886	0.99306	
11	2102	-0.5255	-2.2649	12.3289	0.18894	0.00922	0.03318	35.6631	-0.6886	0.99306	
12	2259	-0.0838	1.62087	9.56364	0.61594	-0.0067	0.46873	35.6631	-0.6886	0.99306	
13	2508	-1.2905	1.5287	11.9183	-0.5643	-0.7147	-0.0951	35.6631	-0.6886	0.99306	
14	2658	-0.674	1.07859	9.54927	-0.2344	0.76913	-0.0566	35.6631	-0.6886	0.99306	
15	2871	-1.2869	2.83234	9.09198	0.26224	0.15888	0.2873	35.6631	-0.6886	0.99306	
16	3153	-1.1337	2.7665	9.17817	-0.1018	0.04098	0.02585	35.6631	-0.6886	0.99306	
17	3309	-1.166	2.50553	9.17698	-0.1428	-0.0726	-0.0016	35.6631	-0.6886	0.99306	
18	3487	-1.2881	2.48159	9.48942	0.0014	0.13261	0.15352	35.6631	-0.6886	0.99306	
19	3722	-1.2186	2.39061	9.55765	0.00751	-0.0513	-0.0756	35.6631	-0.6886	0.99306	
20	3891	-1.0175	2.29963	9.48224	0.06371	0.04098	-0.009	35.6631	-0.6886	0.99306	
21	4086	-1.3443	2.47321	9.35055	-0.0469	-0.0311	-0.0035	35.6631	-0.6886	0.99306	
22	4274	-0.9912	2.44568	9.6199	-0.0395	-0.072	0.0344	35.6631	-0.6886	0.99306	
23	4460	-0.7961	2.17872	9.4164	-0.0487	-0.025	-0.0047	35.6631	-0.6886	0.99306	
24	4703	-1.0606	2.40737	9.42358	-0.0023	-0.0232	-0.1336	35.6631	-0.6886	0.99306	
25	4858	-0.5986	1.78009	9.41879	-0.0279	0.02266	0.06067	35.6631	-0.6886	0.99306	
26	5072	-1.0953	2.13683	9.67736	0.02462	-0.0061	0.02096	35.6631	-0.6886	0.99306	
27	5280	-0.826	2.27449	9.43794	0.07227	-0.0116	-0.1513	35.6631	-0.6886	0.99306	
28	5484	-0.9433	2.33794	8.80707	0.22743	0.15216	-0.0988	35.6631	-0.6886	0.99306	
29	5673	-0.8691	2.37984	9.18775	0.08082	0.03365	0.01241	35.6631	-0.6886	0.99306	
30	5871	-1.0032	2.38702	9.66779	0.03073	0.00739	-0.0536	35.6631	-0.6886	0.99306	
31	6053	-1.1013	2.39779	9.62708	-0.0414	-0.0855	0.00997	35.6631	-0.6886	0.99306	
32	6257	-0.5435	1.98838	9.4559	-0.158	-0.0213	0.05151	35.6631	-0.6886	0.99306	
33	6481	-0.0096	2.29604	8.15106	-0.2442	-0.0494	-0.0939	35.6631	-0.6886	0.99306	
34	6680	0.2909	3.38301	5.59525	0.39419	-0.4813	0.26165	35.6631	-0.6886	0.99306	
35	6875	-1.2474	1.37188	10.5752	1.31904	-0.6078	0.06067	35.6631	-0.6886	0.99306	
36	7062	-2.8515	-0.6704	17.6249	0.20727	-0.2211	-0.0682	35.6631	-0.6886	0.99306	
37	7272	-1.8974	0.26097	11.9997	-0.5704	0.72149	-0.1623	35.6631	-0.6886	0.99306	
38	7471	0.87388	3.26809	6.23091	-0.6589	0.43743	-0.0022	35.6631	-0.6886	0.99306	
39	7652	0.24421	3.3447	6.98269	0.11319	0.05625	0.09732	35.6631	-0.6886	0.99306	
40	7865	-0.8691	2.11049	9.80545	0.11319	-0.0678	-0.1256	35.6631	-0.6886	0.99306	
41	8051	-0.4106	2.02669	9.7875	-0.0145	0.08008	0.0283	35.6631	-0.6886	0.99306	
42	8280	-0.7266	2.35709	9.47984	0.00751	0.04282	0.01974	35.6631	-0.6886	0.99306	
43	8456	-1.1157	2.40737	9.52174	0.04355	-0.0225	0.05517	35.6631	-0.6886	0.99306	
44	8696	-0.7254	2.15358	9.6618	0.03012	0.19981	-0.0554	35.6631	-0.6886	0.99306	
45	8857	-0.5842	2.33554	9.17458	-0.0291	0.02571	0.17429	35.6631	-0.6886	0.99306	
46	9060	-0.7123	2.31519	9.5804	-0.045	-0.0433	-0.0242	35.6631	-0.6886	0.99306	
47	9258	-0.6045	2.32118	9.35175	0.00385	-0.0048	-0.0358	35.6631	-0.6886	0.99306	
48	9460	-0.6285	2.33554	9.56603	0.03256	0.03487	0.00447	35.6631	-0.6886	0.99306	
49	9710	-1.1456	2.68989	7.92241	0.04355	0.02815	-0.034	35.663	-0.6886	0.02047	
50	9865	-0.7242	2.20147	9.63068	0.02706	0.01044	-0.042	35.663	-0.6886	0.02047	
51	10030	-0.759	2.19668	9.47146	-0.0084	0.0361	0.0118	35.663	-0.6886	0.02047	

Figure 3.8 – CSV File.

Bibliographie

- [1] Fissuration et orniérage : Problématiques des chaussées.
<https://www.texum.swiss/fr/renforcement-de-chaussees/la-problematique>.
- [2] Flutter - Beautiful native apps in record time. <https://flutter.dev/>.
- [3] Identifiez les avantages d'une API REST.
<https://openclassrooms.com/fr/courses/6573181-adoptez-les-api-rest-pour-vos-projets-web/6817216-identifiez-les-avantages-d'une-api-rest>.
- [4] Pandas - Python Data Analysis Library. <https://pandas.pydata.org/>.
- [5] (PDF) LynxNet : Wild Animal Monitoring Using Sensor Networks.
https://www.researchgate.net/publication/221462069_LynxNet_Wild_Animal_Monitoring.
- [6] (PDF) The Pothole Patrol : Using a mobile sensor network for road surface monitoring.
https://www.researchgate.net/publication/221234378_The_Pothole_Patrol_Using_a_mobile_sensor_network_for_road_surface_monitoring.
- [7] The pothole patrol | Proceedings of the 6th international conference on Mobile systems, applications, and services.
<https://dl.acm.org/doi/10.1145/1378600.1378605>.
- [8] Python (langage) — Wikipédia. [https://fr.wikipedia.org/wiki/Python_\(langage\)](https://fr.wikipedia.org/wiki/Python_(langage)).
- [9] Welcome to Flask — Flask Documentation (1.1.x).
<https://flask.palletsprojects.com/en/1.1.x/>.
- [10] Welcome to Python.org. <https://www.python.org/>.
- [11] What's Revolutionary about Flutter | Hacker Noon.
<https://hackernoon.com/whats-revolutionary-about-flutter-946915b09514>.
- [12] Flask (framework). *Wikipédia*, Sept. 2020.
- [13] Pandas. *Wikipédia*, June 2020.
- [14] Speed bump. *Wikipedia*, Aug. 2020.
- [15] R. Bhoraskar, N. Vankadhara, B. Raman, and P. Kulkarni. Wolverine : Traffic and road condition estimation using smartphone sensors. In *2012*

BIBLIOGRAPHIE

- Fourth International Conference on Communication Systems and Networks (COMSNETS 2012)*, pages 1–6, Jan. 2012.
- [16] P. Mohan, V. N. Padmanabhan, and R. Ramjee. Nericell : Using mobile smartphones for rich monitoring of road and traffic conditions. In *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*, SenSys '08, pages 357–358, New York, NY, USA, Nov. 2008. Association for Computing Machinery.
 - [17] D. Nassima. Accidents de la route : Alger en tête du classement au niveau national en 2019. <http://www.aps.dz/societe/99373-accidents-de-la-route-alger-en-tete-du-classement-au-niveau-national-en-2019>.
 - [18] T. Nomura and Y. Shiraishi. A method for estimating road surface conditions with a smartphone. *International Journal of Informatics Society*, 7(1) :29–36, 2015.
 - [19] A. Rahmouni. Bindex. <https://github.com/Noisy96/pfe-report>.
 - [20] J. TILLU. Mobile sensors : The Components that make our smartphones smarter. <https://medium.com/jay-tillu/mobile-sensors-the-components-that-make-our-smartphones-smarter-4174a7a2bfc3>, June 2019.