Engineering Project

Road Quality Assessment

Team:
ALLEMAND Fabien
BALAKRISHNAN Sylvain
BONNAIL Julie

Alcatel·Lucent

École d'ingénieurs
Télécom Physique
Université de Strasbourg

Contents

List of Figures

Theme	ID	User	User Stories	Business Value	Complexity	Sprint Complexity	Action
Entraînement d'un premier modèle	1	Administrateur	En tant qu'administrateur, je souhaite obtenir suffisamment de données (de bonne qualité) pour entrainer un premier modèle (au moins 10 données pour chaque type de dégradation)		13		-Créer un système embarqué (Arduino, accéléromètre, GPS, carte SD) -Robot Innovlab -Construire un premier modèle
Collecte de données (accélérométriques)	6	Conducteur	En tant que conducteur, je souhaite que la collecte de données se fasse de façon simple (pas d'orientation précise pour le téléphone)	5	8	1,6	-Application pour smartphone permettant de collecter les données accélérométriques et GPS (Android Studio)
Détection	4	Responsable route	En tant que responsable des routes, je souhaite être informé au plus vite et de façon pertiente des détériorations dans les chaussées pour agir au plus vite (ex: une fissure a été détecté il y a 10 minutes à la position GPS [...], sur la carte cela correspond au rond point devant TPS, il faut donc que je m'équipe d'une pelle et de goudron pour pouvoir la réparer. Si ça avait été un trou j'aurais du aller chercher du sable et du gravier en plus.)	5	13	2,6	-Notifications -Application avec affichage sur carte -Application avec code couleur pour le type de dégradation et l'importance
Collecte de données (observation terrain)	7	Responsable route	En tant que responsable des routes, je souhaite participer à l'amélioration du modèle de façon simple (questionnaire rapide de vérification)	8	3	0,375	-Application pour smartphone permettant de recueillir les données observées sur terrain
Amélioration du modèle	5	Administrateur	En tant qu'administrateur, je souhaite pouvoir entrainer mon modèle en continu avec les données observées sur le terrain par le responsable route	13	5	0,3846153846	-Online learning
Fausse alerte	3	Responsable route	En tant que responsable route, je tolère au plus trois déplacements inutiles (fausses alertes) par mois	8	5	0,625	-Données d'entrainement -Modèle
Gestion des données	12	Conducteur	En tant que conducteur, je souhaite que ma position et mes déplacements restent privés	3	13	4,333333333	-Ne pas diffuser les données -Compatibilité avec la RGPD
Avertissement de danger	10	Conducteur	En tant que conducteur, je souhaite pouvoir être averti des dangers sur la route que j'emprunte par une notification	2	5	2,5	-Envoi d'une notification à un conducteur qui arrive au niveau d'une détérioration -Envoi d'une notification à un conucteur lorsqu'une dégradation est détéctée sur un trajet fréquent
Récompense	11	Conducteur	En tant que conducteur, je souhaite être récompensé pour participer à ce programme	5	1	0,2	-Envoyer "bravo" sur l'application
Détection	2	Responsable route	En tant que responsable route, je souhaite pouvoir détecter l'apparition de différents types de dégradations dans les chaussées (trous, fissures, affaissements, bosses, surface rugueuse...)	21	13	0,619047619	-Données d'entrainement contenant des trous, des fissures, des affaissements... -Modèle
Amélioration du modèle	9	Administrateur	En tant qu'admistrateur, je souhaite pouvoir suivre l'évolution de mon modèle (visualisation du taux d'erreur, etc.)	13	3	0,2307692308	-Dashboard

Figure 1: User Stories

1 Introduction

Road infrastructures are key when it comes to traveling. Whether it is for daily commuting or one-time journeys, millions of people drive their vehicle on the road in order to go from a point A to a point B.
There is no denying that the state of deterioration of roads has a huge impact on the security of the drivers and passengers. An unexpected hole on a road can lead a conductor to change direction abruptly or loose control of the vehicle.
The effect of a poorly maintained road on vehicle is usualy overlooked but it seems logical that holes and bumps on a road are likely to cause damage on cars reducing security and increasing maintenance costs on the vehicle.
At a larger scale, transportations can be slowed down by deteriorated roads meaning the entiere process of economical exchanges is running at a slower pace, hurting the economy of cities or even countries.
Finally, the military force of a country can be evaluated by the state of roads networks. In case of emergency, military forces need to move quickly. Once again, the state of the roads is a key factor.

The goal of this project is to develop an AI-based solution in order to facilitate road maintenance. By training an AI to recognize degradation on a road, road-wrokers could more easily service roads and thus improve security and user experience.
In this study, the AI will be mostly trained on acceleration data measured on vehicles. In order to work properly, the model must be able to detect various degradations (bumps and obstacles, holes and cracks as well as gravel) regardless of the type of vehicle the data is coming from.
For the purpose of the study, two methods will be used in order to collect acceleration data. First, using an Arduino and an *Inertial Measurement Unit* (IMU). In a second time, by using smartphones accelerometers. As a proof of concept, a smartphone application will be created and will demonstrate the effectiveness of this road quality assessment method.

2 R1

Just like every projects, the first steps consisted in understanding the problematic, the scope and the challenges. We made a few researches regarding road quality assessment and related topics in order to learn about already existing solutions or relevant methods.

We were then able to define sprint objectives and organise them by using *user stories* and *sprint complexity* (Figure 1). The sprints were then used to plan the project progress.

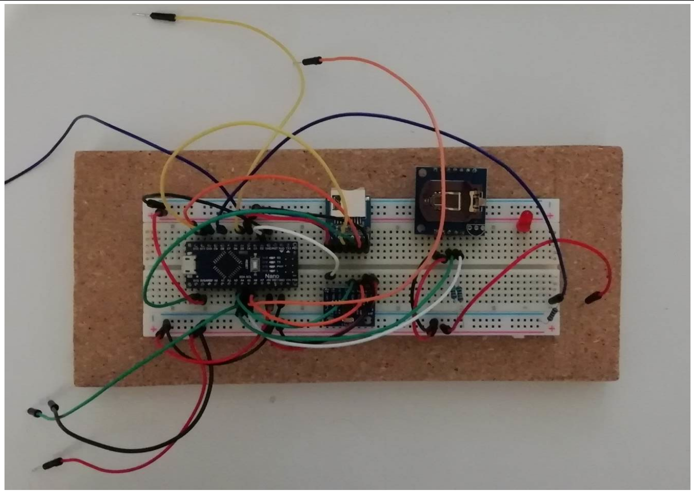

Figure 2: Prototype of the Arduino device

The project was devided in three distinct parts:

- Collect data
- Train an AI
- Develop an application

For the most part, these three aspects of the project can be developped independantly meaning we will be able to divide tasks in the future.

However, as the entiere project is based on acceleration data, we decided to focus on data collection at the beginning. We thought it would help us understanding the project furthermore (what kind of data we are working with, how to collect acceleration data, what does it look like...) and avoid any misleading assumption and useless work.
We designed a basic device around an Arduino Nano and an IMU that would provide sample of acceleration data to visualise and analyse. At the time of the last review the device was still in developement (Figure 2) due to unexpected behavior of the Arduino controler when connecting multiple modules on the I^{2}C bus.

3 State of the Art

Artificial intelligence is being used more and more and is becoming more and more powerful. Current AI models are capable of responding to complex tasks with solid repercussions. The applications are diverse: audio signal processing, medical image classification, or defect detection on production lines.
There are already many applications regarding roads notably: drunk driving detection [8][2] or vehicles control and monitoring [13] without forgetting autonomous vehicles which do not cease evolving [9].
Road condition analysis and damage detection have been the subject of much research. Some researchers use external accelerometers in a series of filters to identify potholes or railway crossings via time-domain analysis only [12][6]. Data are generally used in the frequency domain [5][1].
The use of smartphone accelerometric sensors has already proven to be a relevant alternative to external sensors [7]. Systems that performs rich sensing using smartphones used by people on regular day-to-day driving, where several sensing modalities, such as accelerometers, microphones, GSM radio and GPS are used to monitor both road and traffic conditions [11].
Other machine learning applications use accelerometric data. The rise of connected watches has revealed the usefulness of this data type for fall detection [3] [4] using edge artificial intelligence architecture [10].

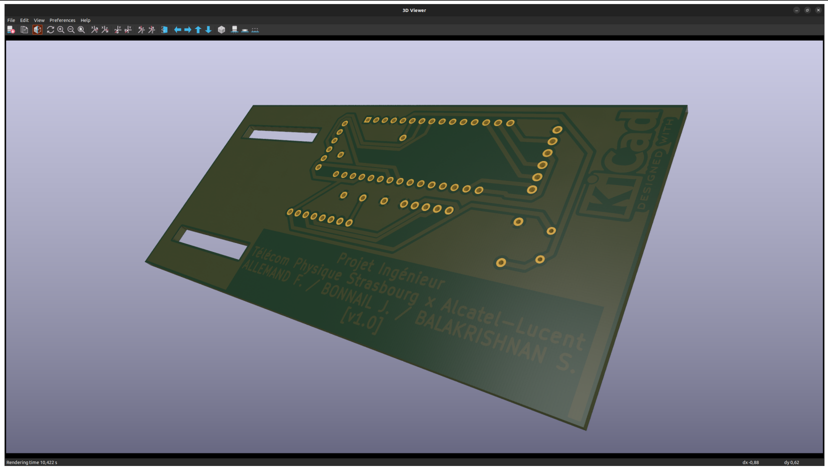

Figure 3: Preview of the Arduino device PCB

4 Data Collection

In order to start working, we obviously needed some data. This section presents the three methods we used in order to get samples of acceleration data. The resulting datasets will be analysed (Section 5) and tested in order to find the best way to collect such data on a vehicle.

4.1 Robot

Being able to collect our own data is a huge advantage because we can also record metadata that can become handy for better understanding the raw accelerometric data but also for future experiments (for instance, comparing two captors or their placement on the vehicle).

It was possible thanks to the acquisition of a brand new radio-controled robot for the Télécom Physique Strasbourg Innov'Lab. This vehicle robot, an Agile-X Scout 2.0, is built for outdoor operations meaning it is perfectly suited to roll on bumps and cracks we had already spotted in the school parking lot.

Instead of waiting for our application, we decided to speed-up the process by using an Arduino to collect data. We paired an Arduino Nano with a fairly affordable but reliable IMU: an MPU-6050, which is a common module for DIY drones hobbyists and a micro-SD card module in order to record acceleration data. At the beginning, we also wanted to connect a *Real Time Clock* (RTC) module and a GPS module (respectively a DS1307 and a BN-880) in order to track both time and position but we had to settle back for only acceleration data due to Arduino and I^2C bus limitations.

After a lot of prototyping and debugging (Figure 2), we were finally ready to build a *Printed Circuit Board* (PCB) to conveniently hold the components together, removing any risk of disconnection during data collection. We designed the PCB with Ki-Cad (Figure 3) and quickly made two of them with the school FabLab milling machine (Figure **??**).

In order to properly mount this little device on the robot, we designed a mounting plate in Solidworks that would acomodate us with many mounting holes for both attaching the plate on the robot and securing our devices on the plate. A FabLab manager helped us to laser-cut it in a large piece of acrylic (Figure 4).

As using the robot takes a lot of time and effort, we prepared a detailed protocol beforehand describing all manipulations and data or metadata to collect. We also proposed some safety measures because the robot is both big, heavy and rather quick.

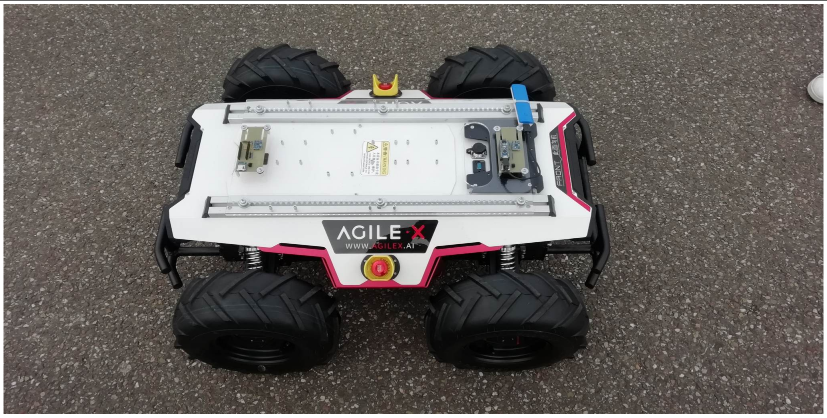

Figure 4: First assembly of the Arduino devices on the robot vehicle

Figure 5: Final assembly of the Arduino devices on the robot vehicle

Unfortunately, the first use of the teleguided vehicle did not go as planned.
We had planned to use two Arduino devices by powering them with smartphones power banks because the robot was not yet ready to deliver 5v to the Arduino. Before any measurement, we noticed one of the battery was broken, leaving us with a unique recording device. Then, after a couple rides we tried to unload data from the micro SD card but there was none. Because the Arduino consumes so little power, the power bank was not detecting any charge and turned itself off... It was decided to use an old laptop straped with two wide elastics in order to provide power (Figure 5) allowing us to reuse our second IMU. But either due to some damage inflicted to the recording device during a driving test or due to some micro SD card incompatibility (formatting format or too large capacity), the second Arduino device actually never recorded any data.

4.2 Car

The data collected from inside a car is quite different from our measurements on the robot.

First of all, the captor used to acquire accelerometric data is a smartphone accelerometer (utilised with an Android application named Physics Toolbox Sensor Suite). According to the person who performed the measurements, it was put on the passenger seat in the longitudinal direction for the first two recordings and sideways to the car for the last four collections.

Moreover there is GPS data associated with every recording. The position was not recorded by the smartphone but by a smartwatch.

In theroy, this new way of collecting data brings diversity to our dataset both in terms of vehicle and regarding degradations. The differents tracks contains new obstacles we cannot yet study with the robot such as: big speed-bumps, subsidences, manholes, bridge joints, rough strips, pavers and damaged pavement. But it remains to be seen if we can easily labelise the data using the driver's knowledge of the road.

Remark As the mobile orientation is neither the same as the car's nor the same as real world coordinate system, a small program must be implemented in order to reproject acceleration data along relevant axis.

4.3 Online Dataset

A machine learning model needs an important amount of data to be trained. Collecting data ourselves is very laborious and labelising it would be very time consuming. So using online datasets would allow us to work with a larger and more diverse database.
A dataset containing a large amount of files was found online. These files mainly contains acceleration along the X, Y and Z axis of an unknown vehicle driving on top of some sort of obstacle (*Metal bumps*) at two different speeds (unknown unit). Although these files looks to be suitable for AI training (labels and training/testing sets) it cannot be used unless we find more information about the data in contains.

5 Data Analysis

Following Section 4, there are now three different datasets. It is now time to visualise and analyse the data in order to gain insights before using it in order to train an AI. This section describes the method used and the results for each dataset.

5.1 Method

In the frist place, we worked in a Python Notebook which allows us to execute code cells independantly making the analysis process more user-friendly.
Acceleration and GPS (if available) data are loaded in Pandas dataframes giving access to many usefull functions to understand the data. The most important methods are `describe` and `isna`. The first one provides general statistic values regarding the data (count, mean, standard deviation...) and the second one can be used with `sum` in order to compute the amount of missing values for each attribute.
Finally, different methods allowing to visualise accelerometric and/or GPS data were implemented. Eventhough Matplotlib figures are great (Figure 6), HTML figures created with Plotly have the huge advantage of being interactive on all devices and operating systems with a internet browser. This means it is possible to zoom in and out on a specific part of a graph which is very important when working with high frequencies or long recordings. Two types of graphs look interesting depending of the need:

- Every attributes on a single graph (Figure 7)
- Each attribute on a separate graph (Figure 8)

Working with HTML figures also allows to tie acceleration and GPS data on a single page (Figure 9).

In a second time, we created a Python script which automaticaly generates figures depending of the type of data and user preferences (output format) specified in arguments.

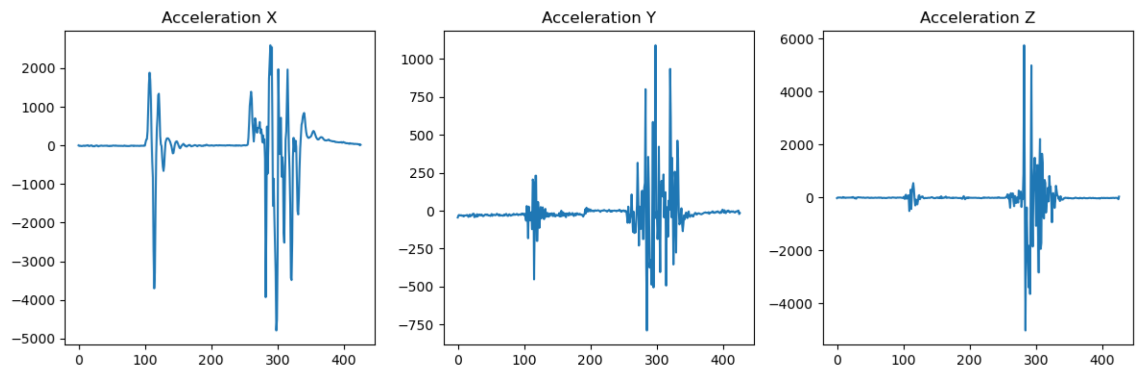

Figure 6: Acceleration data recorded with the Arduino device on the robot while rolling straight on a small speed-bump (plotted with Matplotlib)

5.2 Robot

Reminder

The force of gravity has been removed from the acceleration measurements and the Arduino device was mounted on the robot taking care of the orientation of the IMU:

- Positive X = Froward
- Positive Y = Left
- Positive Z = Up

As expected, there is no missing values in the data recorded with the Arduino device and, on the most simple samples, it is really easy to understand the acceleration data. On Figure 6, we can clearly see huge variations along the Z axis when passing the bump (meaning the vehicle is effectively changing speed along verticaly due to the bump) and smaller variations along the X axis (because the vehicle is slowed down on the rising section and accelerates when going down). We can also see some even smaller variations form side to side (Y axis) which are mostlikely due to the robot damping system and vibrations of the mounting plate.

Remark

The sampling frequency (i.e. horizontal granularity) is defined by the Arduino program as: $f = \frac{1}{10 \times 10^{-3}}$ which correspond to a delay of 10 ms between two measures.
It is important to pay attention to the vertical scale of each graph.

Plotting data on a single graph is usefull in order to check the timing of shocks along the three axis. Figure 7 shows there are acceleration variations during two brief periods regardless of the axis. These periods correspond to the start of the vehicle (when it begins to move, ranging from 100 to 150) and the ride over the bump (starting at 250).

However looking at longer recordings, it is already difficult to recognize obstacles at a glance. Figure 8 correspond to the robot rolling in a hole, then over a small crack and finally on top of some little bumps. This recording only lasts a couple of seconds so it is way shorter than the data we expect to get when collecting data on real journeys.

5.3 Car

The data collected in a car was also analysed using the same method and fortunatly there was no missing values to report. However we noticed small fluctuations in the sampling rate. The timestamps on the acceleration data are separated by values ranging from 20 to 30 milliseconds. This issue has not been adressed yet but it is only a minor correction and it does not prevents us from visualising GPS data and making sure we can access accelerometric data.

It seems that the recording method needs to be improved. On the one hand, GPS data are correct and

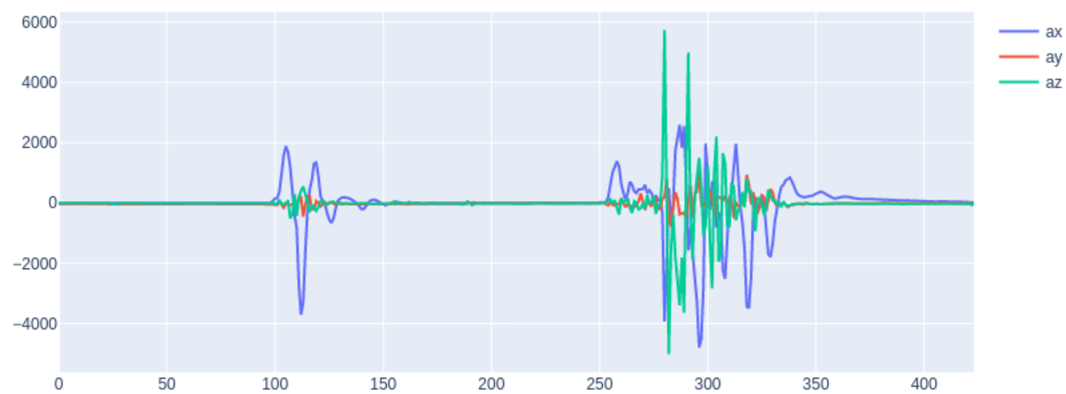

Figure 7: Acceleration data recorded with the Arduino device on the robot while rolling straight on a small speed-bump (plotted with Plotly)

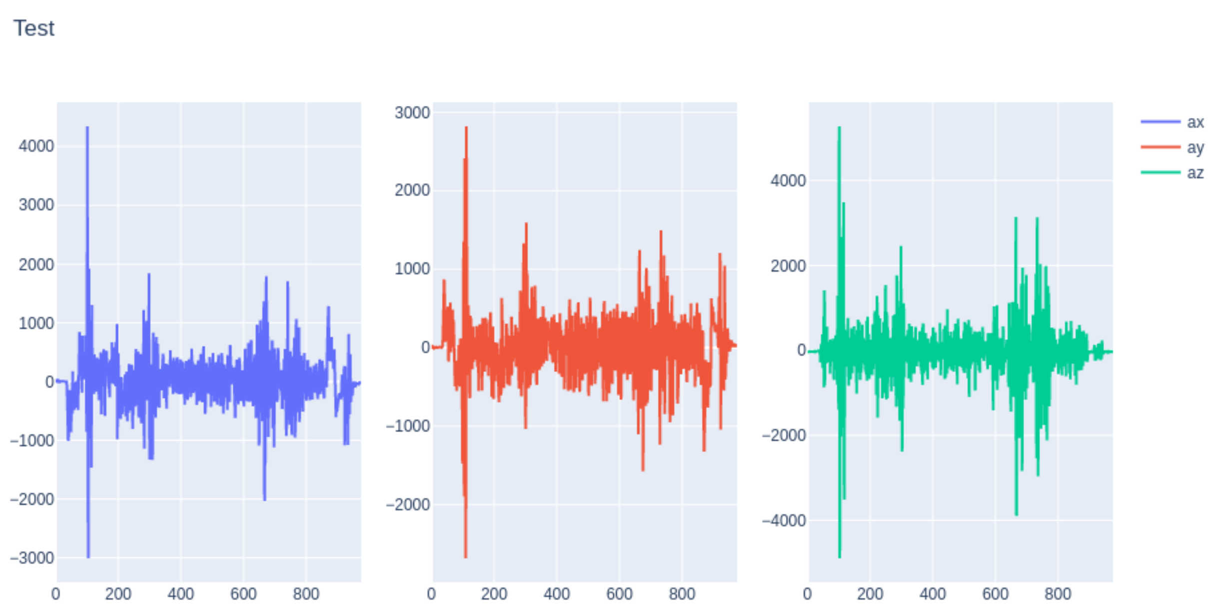

Figure 8: Acceleration data recorded with the Arduino device on the robot while rolling on various obstacles (plotted with Plotly)

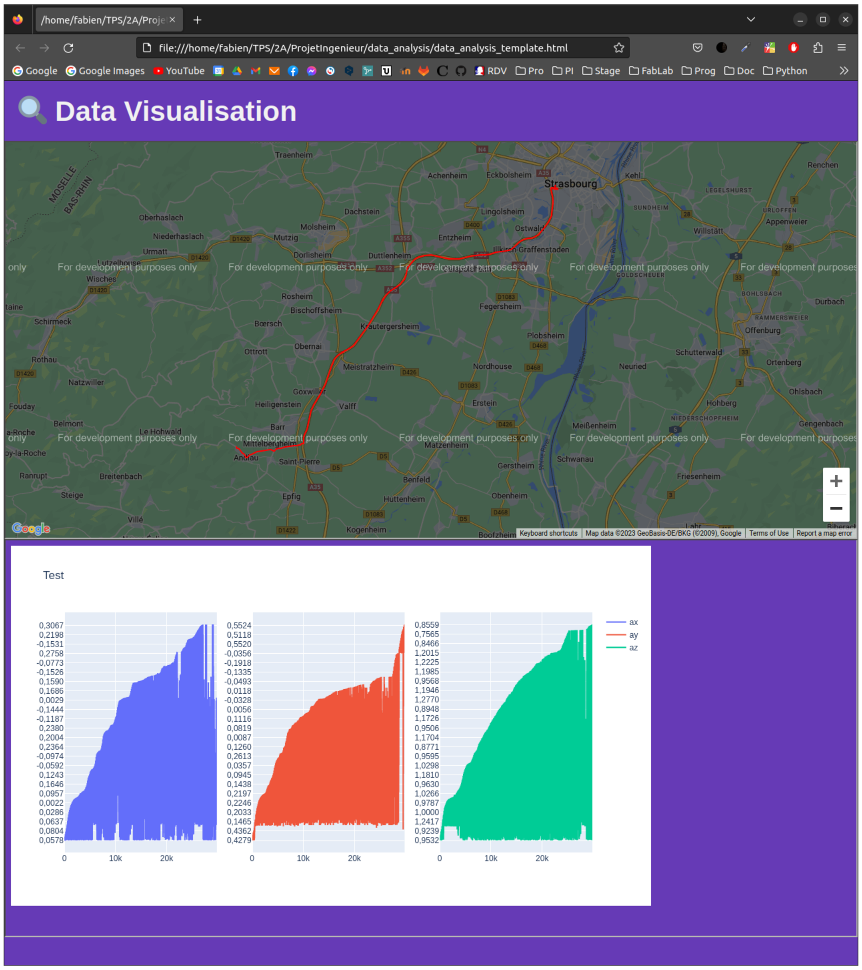

Figure 9: Acceleration and GPS data recorded with a smartphone and a smartwatch in a car (plotted with Plotly and Gmplot)

quite precise, but on the other hand, acceleration data has an unexpected and incoherent shape (Figure 9).

5.4 Online Dataset

The data contained in this last dataset requires a different processing. Eventhough its origin and the methodology used to build it are still unknown, we managed to get a few informations from the files.
First and formost these files contain labelised data: most files are json files containing acceleration along the X, Y and Z axis as well as some metadata at the beginning. This metadata contains among other things the time period of the vehicle riding on the obstacle and the type of obstalce (for the most part *Metal bumps*). It is also worth mentioning that the files are separated in folders according to the speed (either 30 or 50 of an undetermined unit) and their use (training or testing).

An included Python script allowed us to visualise some recordings (Figure 10). According to a couple of previews, metadata looks correct but we will have to investigate and find more information about this dataset.

Remark || This dataset does not reduce the force of gravity from the acceleration which explains the close-to-ten offset of the Z component.

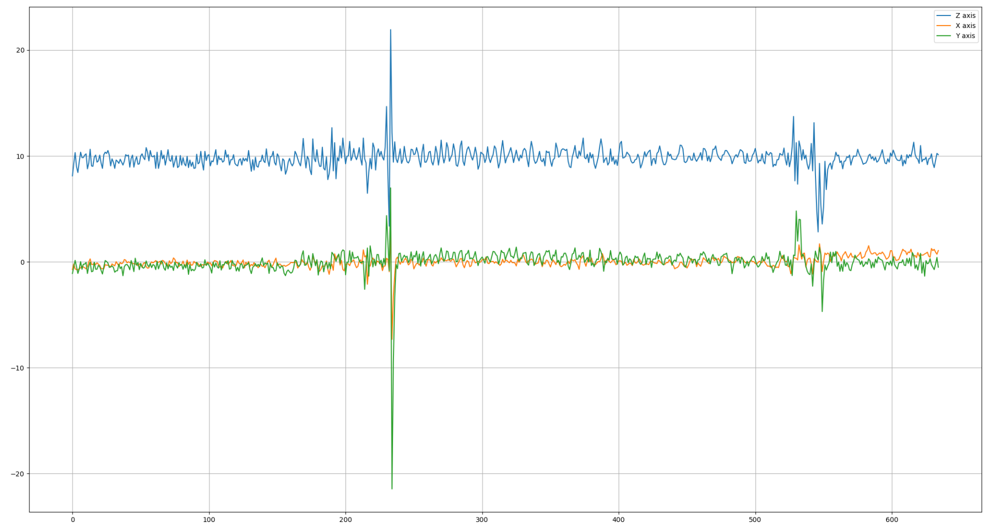

Figure 10: Acceleration data from a mysterious dataset

6 Smartphone Application

The developement of the application will have to start during the next sprint and continue throughout the rest of the project in parallel with the other tasks.This process will take a lot of time because none of us has any experience what so ever regarding developing smartphone application.
We made a few research on that subject and settled for developing an Android application using Android Studio as we do not have any Apple device and Android Studio combines all the tools required for developing a proper application. By following a couple of tutorials on the Android Developers website we managed to build a rather nice looking (but useless) application (Figure 11).

7 Conclusion

Having understood the subject, its issues and constraints, having created supports to use the sensors and having started to collect our first accelerometric data, we can move on to the next stage of the project. Between now and the next review, we plan to analyze the data collected. This will give us an idea of the artificial intelligence model to use. Once this is done, we will have to collect more data so that the selected models are trainable and as accurate as possible. We will then be able to compare the models and choose which one(s) we will select for the rest of the project. Finally, we will also start to develop the mobile application because it can be done in parallel with the artificial intelligence part.

Figure 11: First Android application made using Android Studio

References

[1] Alessio Martinelli et al. Road surface anomaly assessment using low-cost accelerometers: A machine learning approach.

[2] Juan Liu et al. Drunk driving detection.

[3] Luca Palmerini et al. Accelerometer-based fall detection using machine learning: Training and testing on real-world falls.

[4] Osama Zaid Salah et al. Accelerometer-based elderly fall detection system using edge artificial intelligence architecture.

[5] Thanuka Wickramarathne et al. On the use of 3-d accelerometers for road quality assessment.

[6] B. Hull R. Newton S. Madden J. Eriksson, L. Girod and H. Bal-akrishnan. The pothole patrol: Using a mobile sensor network for road surface monitoring (in proc. int. conf. on mobile systems, applications and services (mobisys), breckenridge, usa, 2008).

[7] Madis Kariler. Road surface anomaly assessment using low-cost accelerometers: A machine learning approach.

[8] Juan Liu. An intelligent online drunk driving detection system based on multi-sensor fusion technology.

[9] Nvidia. Solutions pour les véhicules autonomes.

[10] Raed Abdulla Osama Zaid Salah, Sathish Kumar Selvaperumal. Accelerometer based elderly fall detection system using edge artificial intelligence architecture (in international journal of electrical and computer engineering).

[11] V. Padmanabhan P. Mohan and R. Ramjee. Nericell: Rich monitoring of road and traffic conditions using mobile smartphones (in acm conf on embedded network sensor systems (sensys)).

[12] Wikipedia. Détection et localisation des défauts de la chaussée.

[13] XXII. Xxii : Valorisons le temps humain.