Road Event Mapping Method for Mobile Devices with Cloud Computing based Technologies

Yoann dos Santos Resende

Versão final após defesa
Dissertação para obtenção do Grau de Mestre em Engenheira Informática
(2° Ciclo de Estudos)

Orientador: Prof. Doutor Nuno M. Garcia
Co-orientador: Dr. SusannaSpinsante

agosto de 2020

Acknowledgements

I would like to express my deepest gratitude to my friend, colleague and supervisor, Prof. Nuno M. Garcia for his guidance, useful critiques and invaluable expertise in the formulating of this dissertation.

I am particularly grateful for the assistance given by my friends and colleagues Sandeep Pirbhulal, Igor Matias and Henriques Zacarias for their valuable and constructive suggestions during the planning, development and writing phases. Their willingness to give their time so generously has been very much appreciated since day one.

Finally, and most importantly, I wish to thank my family and friends for all their encouragement, support and especially patience throughout my work and studies.

Thesis prepared at the Ambient Assisted Living Computing and Telecommunications Laboratory (ALLab), *Instituto de Telecomunicações, Universidade da Beira Interior*, and submitted to the *Universidade da Beira Interior* for defense in a public examination session.

The research has been partially funded by the FCT/MCTES through national funds, and when applicable, co-funded EU funds under the project UIDB/EEA/50008/2020 and Operação Centro-01-0145-FEDER-000019 –C4 –Centro de Competências em Cloud Computing, co-financed by the Programa Operacional Regional do Centro (CENTRO 2020), through the Sistema de Apoio à Investigação Científica e Tecnológica – Programas Integrados de IC&DT. I would also like to acknowledge the contribution of the COST Action IC1303: AAPELE—Architectures, Algorithms and Protocols for Enhanced Living Environments and COST Action CA16226; SHELD-ON—Indoor living space improvement: Smart Habitat for the Elderly, supported by COST (European Cooperation in Science and Technology).

Resumo

Esta dissertação descreve a investigação realizada para identificar problemas que podemos encontrar na estrada como travagens ou acelerações súbitas, buracos na estrada ou lombas, usando equipamentos.

Para a abordagem deste estudo foi necessário desenvolver uma aplicação para recolha de dados de acelerómetro e GPS recorrendo a um *smartphone* colocado dentro de um carro.

Apos a recolha, os dados foram tratados e analisados por um modelo de inteligência artificial que identifica os eventos recolhidos pela aplicação. Se os resultados obtidos forem bons, será possível o desenvolvimento de um sistema de controlo do estado das estradas que poderá ser utilizado de forma pervasiva.

Palavras-Chave

Acelerómetro;Global Positioning System;Eventos da estrada;Smartphones; Estado da Estrada; Estradas Seguras; Inteligência Artificial; Modelo de Classificação Supervisionado; Estilo de Condução; Computação na Cloud.

Resumo Alargado

Introdução

Esta secção apresenta o resumo alargado da dissertação intitulada "Método de mapeamento de eventos da estrada para dispositivos moveis com tecnologias baseadas em computação na Cloud" (*Road event mapping method for Mobile Devices with Cloud Computingbased technologies*).

Este resumo alargado tem a seguinte organização: em primeiro lugar, apresentam-se a descrição do problema e o principal objetivo da dissertação. Seguintemente, é feita uma breve descrição do trabalho desenvolvido bem como os principais resultados. Por último, são descritas as conclusões deste trabalho e propostas para trabalho futuro.

Descrição do Problema e Objetivos da Dissertação

A deteção do estado do pavimento das estradas é um problema comum à maioria dos países do mundo, independentemente do seu desenvolvimento, ter estradas seguras deve ser a prioridade par eles. Os métodos atuais para verificar o estado das estradas não são económicos e são custosos em termos de tempo e recursos humanos.

A principal motivação para a realização deste projeto é fornecer mais informação, resultados, dados fiáveis e um sistema funcional a fim de ser possível o desenvolvimento de uma futura aplicação móvel, que utilize esta informação. Tendo isto em consideração, o maior objetivo para este projeto é a deteção de eventos da estrada.

Foram analisados vários estudos onde verificamos que vários aspetos poderiam ser abordados com uma metodologia diferente.

Este trabalho fornece as seguintes contribuições:

1. Estabelece o estado da arte nesta matéria até aos dias atuais.
2. Desenvolveu uma aplicação móvel para recolha de dados para posterior treino de um modelo de inteligência artificial.

3. Desenvolveu uma parte que usa tecnologias de computação na Cloud.
4. Fornece uns testes e analise de resultados deste sistema e conclusões finais.

Estado da arte

Neste capítulo, apresentamos uma breve introdução sobre o trabalho que já foi feito nesta arte.

Alguns estudos já abordaram este tema, mas apenas focaram na deteção automática do estado do pavimento das estradas e não focaram no comportamento dos condutores que esta relacionado a esta arte. Os poucos que estudaram esta arte, usaram sempre métodos diferentes e obtiveram resultados diferentes, o que demostra que há necessidade de efetuar mais estudos com diferentes abordagens e métodos.

Com base nesta breve conclusão foi tomada a decisão de prosseguir com este estudo, utilizando uma abordagem diferente.

Desenvolvimento

Em resumo, neste capítulo apresentamos o desenvolvimento do projeto, onde descrevemos todas as fases do mesmo, desde o planeamento, materiais e métodos, o desenvolvimento de várias versões da aplicação de recolha de dados e a analise dos dados obtidos.

Primeiro, foi necessário criarvárias versões de uma aplicação móvel para recolher dados de acelerómetro e GPS e armazená-los. Será então desenvolvido um método para analise dos mesmos.

Para este projeto é necessário registar dados do mesmo período de tempo, para ter consistência entre as amostras. Foi também preciso registar muitos dados, de diferentes eventos na estrada (aceleração, travagem ou velocidade normal) sobre diferentes tipos de estrada (estradas de terra, alcatrão, calçada portuguesa e autoestradas) para poder ter uma grande variedade de dados. Esses dados são guardados no telemóvel num arquivo de texto (txt) que foi enviado para a base de dados de Microsoft Azure de onde foram posteriormente extraídos para serem transformados para poder treinar um modelo de inteligência artificial que poderá depois classificar os diferentes eventos da estrada recolhidos pelos smartphones. Os dados recolhidos contem dados sobre o acelerómetro, GPS e a hora da coleta, esses dados são todos de 100 linhas com 10 medidas por segundo o que equivale a 10 segundos.

A fim de obter a máxima precisão do nosso modelo de inteligência artificial, testamos diferentes métodos de compressão de dados usando formulas matemáticas porque para poder treinar um modelo e obter uma precisão boa é preciso milhares de dados recolhidos e não foi possível dado o tempo e com só uma pessoa a recolher dados.

A primeira abordagem foi a compressão de dados e escolhemos a linguagem Java para esta tarefa. Foi feita uma conversão de ficheiros txt em ficheiros Excel, e depois de fazer a compressão foram criados ficheiros csv para poder ser mais simples a treinar o modelo de inteligência artificial em Python. Depois de testar muitas formulas matemáticas, chegamos a ter uma fórmula matemática que nos desse um modelo com uma precisão melhor que as outras, os resultados podem ser vistos nas tabelas 1 e 3.

Depois foi preciso adaptar o sistema para dados mais compridos do que 10 segundos, foi feito uma modificação ao programa Java, para separar os ficheiros "grandes" em pequenos ficheiros de 100 linhas.Foi também feito uma função para detetar lombas e buracos na estrada.

Para acabar, foi desenvolvida uma aplicação que mapeai os eventos detetados com este sistema usando a API da Google Maps.

Conclusões e Proposta de Trabalho Futuro

Após uma análise de resultados, foi descoberto mais do que esperávamos. Apesar do facto deste projeto não ter a melhor precisão possível sobre todos os tipos de estradas, levou-nos a questionar e rever o projeto de formas diferentes em cada fase e sempre repensar as decisões tomadas previamente para poder ter o melhor sistema possível dado as circunstâncias.

Mais investigações serão necessárias para melhorar este sistema, mas conseguimos ver no futuro a possibilidade de desenvolver uma versão melhor da aplicação móvel proposta neste projeto, usando o método de "*crowdsourcing*" para obter muitos dados e ter uma precisão muito boa em todos os tipos de estradas ou usando um método de deteção de lombas e buracos na estrada mais fiável.

Abstract

Among other features, the road condition influences safety and driving comfort. Driving styles are somewhat related to the road profile, which leads to the need for a monitoring model to detect and signal damaged road segments and danger zones. Some methods were developed to collect and format the data of road infrastructures, in order to estimate the quality of the roads, based on the data collection from a mobile phone's sensors placed inside a vehicle and on the feeding of artificial intelligence classification models. However, these methods do not provide instant feedback to the driver in the vehicle nor checks driving styles and current road condition.

This work aimed to develop a system capable of detecting events on the road from the driver's point of view (this research being limited to breaking and acceleration events), on different types of road pavements, using a machine learning supervised classification method that can be used in the future in a connected architecture to alert drivers of the dangers of the road ahead and therefore help increase road safety.

Keywords

Accelerometer, Global Positioning System, Road Events, Smartphones, Road Condition, Safe roads, Artificial Intelligence, Supervised Classification Method,Driving Style, Cloud Computing.

Contents

List of Figures

List of Tables

List of Acronyms

UBI	Universidade da Beira Interior
AI	Artificial Intelligence
GPS	Global Positioning System
NN	Neural Network
kNN	k-Nearest Neighbors
SVM	Support Vector Machine
PCA	Principal Component Analysis
DTW	Dynamic Time Warping
RF	Random Forest
GB	Gradient Boosting
RSA	Range Search Algorithm

Chapter 1

1 Introduction

According to the European Commission, the total cost of the entire road network for the year 2016 of all the countries in the European Union was about equal to 184 billion euros [1]. The condition of the road infrastructure is related to rolling resistance and, therefore, to the amount of CO2 emissions of combustion engines. Not only that, but also road safety has been severely jeopardized due to the presence of potholes and speed-breakers [2]. Many roads are regularly inspected by qualified staff to decrease the risk of accidents, but budgetary restrictions make it increasingly harder to frequently and closely monitor the state of usability and safety of the roads. Furthermore, road condition is not the only danger when you are driving on the road; that is, unexpected sharp curves are the cause of many car accidents [3].
These are some of the reasons why there is a need for a system that pinpoints some of the events that may have a relation with danger zones, such as systematic zones where drivers suddenly brake or accelerate, being this the first goal of this research.

1.1 Objective

The main objective of this project is to detect specific events of a car, such as acceleration and breaking events, using Artificial Intelligence models and combine it to a method that detects potholes and road bumps in order to identify warning situations. And, to use those two methods to create a mapping of the events detected by the system so that it is easy to recognize and pinpoint warning situations for road maintenance workers.

1.2 Motivation

Road conditions is a real important matter nowadays and monitoring it by road workers is time and resource consuming. Having an automatic road monitoring system would be a low-cost and a more efficient way to identify dangerous zones at a larger scale.
As a car driver myself, I think the problem of having to drive on a damaged road is an importantone and should be taken care of because nobody wants to drive on these types of roads.

My motivation to carry out this investigation is to provide technology developers with credible results and data to assist them in making this system possible all over the world.

1.3 Contribution

This work focuses on the study, analysis and implementation of a much-needed road event monitoring system. The main contribution of this work is to define a monitoring solution that can eventually be integrated in taxi's companies or, in the future, on a automated car that drives alone and detects danger zones automatically.

1.4 Organization

This dissertation is organized and structured in five chapters that demonstrate the process used to accomplish the work's objectives. The chapters are the following:

- **Chapter 1 - Introduction:** The first chapter consists of a general introduction of the dissertation theme presenting the objectives, motivations and contributors of the project's implementation.
- **Chapter 2 –State of Art:**This second chapter focuses on the study of all the relevant parameters of this project.
- **Chapter 3 – Implementation:** In this third chapter, we will see all the steps taken to carry out this project, its planning, the materials and methods used, as well as the target audience and the development of different mobile applications necessary to collect the necessary data.
- **Chapter 4 - Results:** In this second-to-last chapter, the data collected will be treated and analyzed in order to obtain clear and quite objective results.
- **Chapter 5 – Conclusion and Future Developments:** The last chapter deals with the conclusions obtained and possible developments, improvements and analyzes to be carried out.

Chapter 2

2 State of art

2.1 Introduction

In this chapter, we present a brief introduction about monitoring driver habits and road pavement conditions using the smartphone accelerometer.

In [4], a review was done on different studies about this theme to create a method to implement in the Center region of Portugal.
The smartphone must be installed in a fixed position in the car so that the X axis retrieves the curves data, the Y axis retrieves accelerations and breakings and the Z axis is used to detect road bumps [5]. This use of the smartphone is also used to have information about the road pavement state [6].

Different studies have already addressed this topic, but only a few have studied the relationship between the road pavement condition and the speed changes in drivers. The few who approached this issue did not obtain a functional system, which shows that there is a need to further study this problem and come up with different approaches and methods. Based on this brief discussion, the decision was made to proceed with this study, using a different way to identify this relationship.

In this chapter, we will discuss about the discoveries on this theme, in addition to its potential and limitations.

This chapter includes the following sections:

- Section 2 – Data collection
- Section 3 – Methods used
- Section 4 – Measured metrics
- Section 5 – Conclusion

2.2 Data collection

In the study made by [7], a data collection approach using a smartphone perched on the vehicle's center of gravity is presented. The recovered data were as follows: GPS signal, accelerometer and data from the angular rate sensor. The accelerometer and rotation rate data are retrieved on three axes and on a 200 Hertz sample, the GPS data is retrieved at 10 Hertz. To retrieve the speed data, calculations were made using GPS data.

In another study, made by [8], a data collection method using crowdsourcing is presented. They used the same data collection method as in the study by [7], but to collect as much data as possible, they created an android application called RoADApp. Users have to register with their mobile phone numbers to create a unique identifier for everyone. This application can be used in 4 different vehicles (motorcycles, tuk-tuks, cars and buses) and the smartphone can be positioned where the user wants (in the pocket or in a fixed position in the car). The data was retrieved with 15 different vehicles and 9 different smartphones. The data is then saved locally until the users have access to the internet, then the files are sent on a server.

The study made by [9] also used a phone placed inside the car, gathering with it accelerometer and GPS values of the vehicle. They reoriented their accelerometer values using Euler Angles so that it was easier to analyze and detect road conditioning. They also send the data in real time to the central server.

One study by [10] used a phone fixed on a motorcycle to gather accelerometer and GPS data. The data was gathered on four different roads and it was split into three datasets. The system also used the microphone of the device to label the data collected, if the experimenter collecting the data came across a road anomaly, the system collected the voice which is then converted to anomaly class labels.

2.3 Methods used

In [7], the road pavement anomaly detection methods used were the following:

- Support-vector machines (SVMs): Support vector machines are models of supervised machine learning with associated learning algorithms that analyzes data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one of two categories.An SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier.
- K-nearest neighbors (k-NN) algorithm: the nearest k-neighbors algorithm is a non-parametric method used for classification and regression. The algorithm stores all available cases and classifies new cases based on a similarity measure

(most of the time distance functions).

In order to classify events using these methods, they had to test differentmethods of calculating distances between points:

- Euclidian distance
- PCA distance
- Hausdorff distance
- DTW distance.

Road bumps and potholes detection method used in the project made by [8] are:

- Decision tree classification: this classification method uses a decision tree (as a predictive model) to go from observations on an object (represented as branches) to conclusions about the target value of the object (represented as leaves).
- K-Fold Cross Validation: this method is a static method used to estimate machine learning hypothesis performance. The available data are randomly divided into K groups, the model is adjusted K times, each time using one group as a test set and the remaining K-1 groups as learning data. In this study, 10 groups were used (K = 10).

In [6], the road bumps and potholes detection methods used were:

- Random Forest: The Random Forest method is a joint learning method for classification, regression and other tasks that operate by building an infinite number of decision trees when training. This method corrects the habit of overfitting decision trees for your training sets.
- Gradient Boosting: This method is a machine learning technique for regression and classification, which produces a prevision model in the form of a set of weak prevision models, usually decision trees. It builds the model in steps, like other reinforcement methods, and generalize them, allowing the optimization of an arbitrary differentiable loss function.
- Neural Network: Neural networks are multilayer networks of neurons that are used to classify objects or make predictions.

In [9], the method used to detect road anomalies was based on the analysis of the vertical acceleration impulse.First, they removed the “background noise” of the data using a processing algorithm. Secondly, they selected only the high-impulse events and discarded the rest. And finally, if the algorithm they developed detects a road anomaly, it sends its location directly to the central server and shares the information with the other users of the application.

In [10], two methods were tested, one method used a SVM and the other used unsupervised learning.First, the system is fed with data that was labelled using the voice of the experimenter when he came across a road anomaly, then the data is reprocessed and relabeled using a simple heuristic functionto further improve the accuracy of the labelling. The unsupervised learning method is a clustering method that uses data which has been divided into multiple datasets according to speed since speed influences the vibration range.

2.4 Measured metrics

Among the results that were obtained by the different references mentioned above, the [6] study had the weakest results. They used a confusion matrix to evaluate their model. In the three methods they used, they wanted to classify five different events which are “pothole”, “small road bump”, “large road bump”, “no anomaly” and “others”.
Using the Gradient Boosting method, they got the following accuracies:

- No anomaly: 92.8%
- Pothole: 72.0%
- Small road bump: 77.8%
- Large road bump: 70.5%
- Others: 46.2%.

The Random method got them the following accuracies:

- No anomaly: 92.3%
- Pothole: 48.0%

- Small road bump: 63.9%
- Large road bump: 56.8%
- Others: 15.4%.

Using the Neural Network (Linear SVC), they got the following accuracies:

- No anomaly: 97.5%
- Pothole: 12.0%
- Small road bump: 58.3%
- Large road bump: 75.0%
- Others: 0.0%.

In the [7] study, they compared the computing speed of the different distance calculation algorithms between two points. They made tests on a thousand trajectories with data from different distances (50 meters and 200 meters).

The following results are for the comparison of the computing speed of a thousand trajectories of 50 meters:

- Euclidean distance: 0.0050 seconds
- PCA distance: 1.0170 seconds
- Hausdorff distance: 1.1612 seconds
- DTW distance: 0.0598 seconds.

The results that follows are for the comparison of the computing speed of a thousand trajectories of 200 meters:

- Euclidean distance: 0.0105 seconds
- PCA distance: 1.1600 seconds
- Hausdorff distance: 4.8872 seconds
- DTW distance: 0.0708 seconds.

With those results, they obviously chose the Euclidean distance algorithm since it was the fastest computing algorithm. Then, they added another method called the range search algorithm (RSA), that is used to pre-process the data to ultimately speed up the calculation speed of the Euclidean distance algorithm.
In terms of accuracy of the system, they did not share any information, they only did a test to verify if the system could determine if the event they had given to it was classified as "railroad crossing" event, which it did.

The study made by [8] opted for a different approach than the other two.
First, they had to modify the data because the application they conceived allowed users to use the phone in their pocket or in any other position. This was made possible by a method that reorients the data using the gyroscope and a Euler-based approach that determines the orientation of an object in a space. They also realized that speed was a factor to consider, because when the speed is higher the changes in values on the Z axis are much shorter than when the vehicle is at a lower speed. Having that in mind, they defined limits to maximum and minimum values according to the speed of the vehicle. Using only the Z axis, if the axis values were greater than the maximum or lower than the minimum, it detects the events that later transfer to the decision tree that classifies the events between road bumps and potholes.
Their model has a 98.0% for road bumps and 92.0% for potholes on smooth roads and a 91.4% accuracy for road bumps and an 87.7% for potholes on irregular roads.

In the [10], the SVM method had a 78.5% precision in detecting road anomalies correctly, it also ranked the roads from the least damaged to the most damaged. The second method gave a rank and a grade to the data (road segments), we don't have information about the accuracy of this method.

The [9] used three different phones, running the app, inside of a single car on two different test sites to evaluate the performance of their project, so that they could adjust the values of the algorithm used and improve its accuracy.They then proceeded to test the system on four test sites and checked the number of true and false positives events detected by the system. The mean value of this validation phase for true positives was 82.0% and the mean value for false positives was 13.0%.

2.5 Conclusion

We were able to see three different studies on this art with three distinct methods and approaches to this problem, which is the state of the road pavement and how to automatically detect these problems that we encounter daily. These three studies show us that there are several answers to this problem and there is not one that is better than the others.

Many studies on this topic focus on the state of the road pavement and do not study the other part of the problem, which is the habits and trends in people's driving styles. Because, we can probably deduce from these driving styles, for example, if a lot of people suddenly brake in a portion of the same road it may be because of sudden curves, badly placed light signals or even unexpected pothole/road bump. That's what we are trying to bring,with this project, to the already existing work and studies made before.

Chapter 3

3 Design and Implementation

In this chapter, the development of the project is presented. And, all phases of the project are described, from planning to materials and methods used for the development of the applications and analysis of the data obtained.

3.1 Project Development Planning

First, it is necessary to create an application to collect the accelerometer, GPS and the current time data from the smartphone and store it in an appropriate format, in this case it will be a text format, then develop a method to analyze the data collected so that we can have, in the end, a system that detects road events.

3.2 Materials and Methods

For this project, it is necessary to collect recordings of the same events (accelerations, breakings and normal speeds) multiple times on different types of roads so that similarities could be identifiedbetween the same events and understand the data collected.

In order to collect the necessary data, it was decided to develop an Android application, so the materials needed were an Android smartphone and a vehicle.

3.3 Mobile Application

The application to collect accelerometer, GPS and time data was developed in the beginning of this project and was done, using Java, in AndroidStudio. The required version of Android to install it is the 3.6.2 version or above.

Figures 1 and 2 show the initial versions of the prototype app interface.

There were madethree versions of the application:

- The first version was an application that gathered accelerometer, GPS values and time records. When launching it, it starts to gather the data and after pressing the "Save" button in the application, those values are saved into a generated file that was then stored in the device's memory. Ten measurements where made every second for increased accuracy.

- The second one was the same application, but the "Save" button was replaced with the "Timer" button that only recorded the following ten seconds worth of data after pressing it, data that was later used for training purposes.
- The third and final version of the application had two different buttons, the "Start" button for starting the recording of the data and the "Upload" button for uploading the file generated by the application to a Cloud Computing Server (Microsoft Azure).

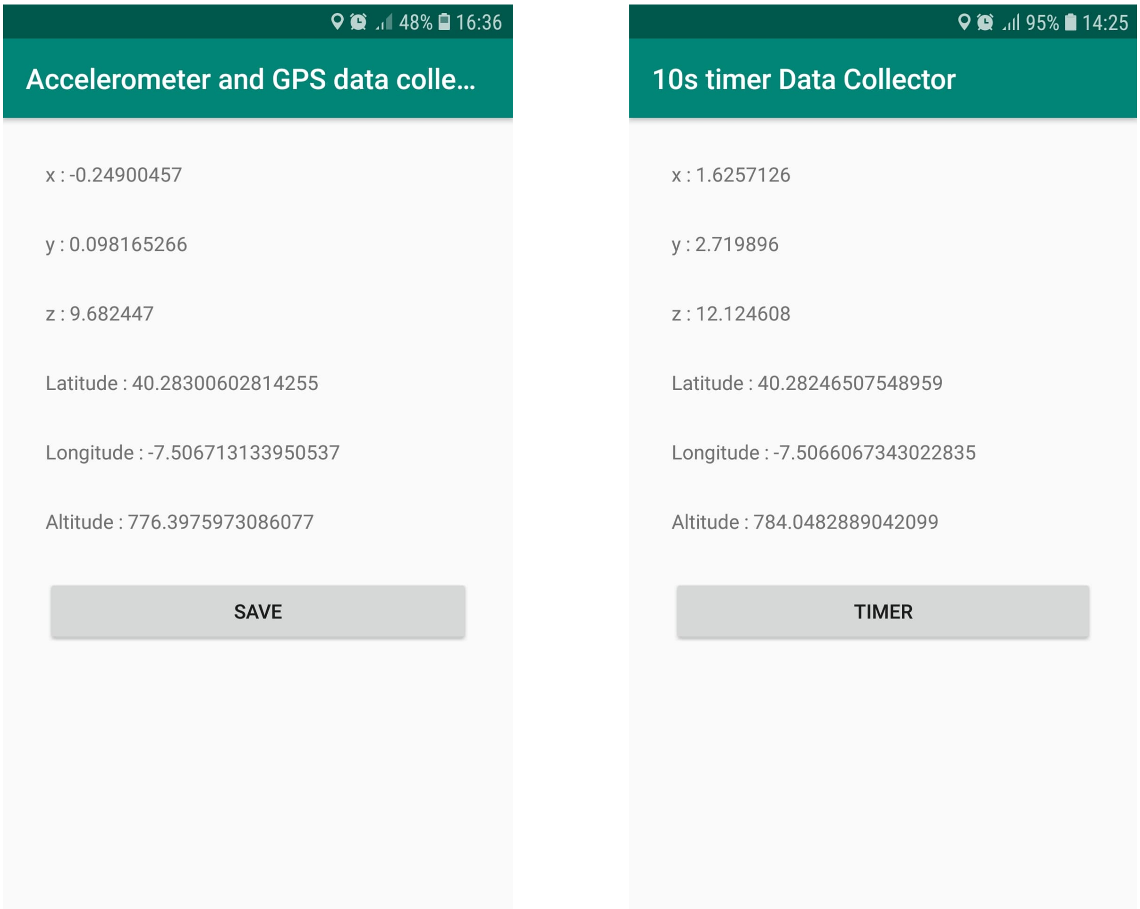

Fig.1. First two versions of the application developed.

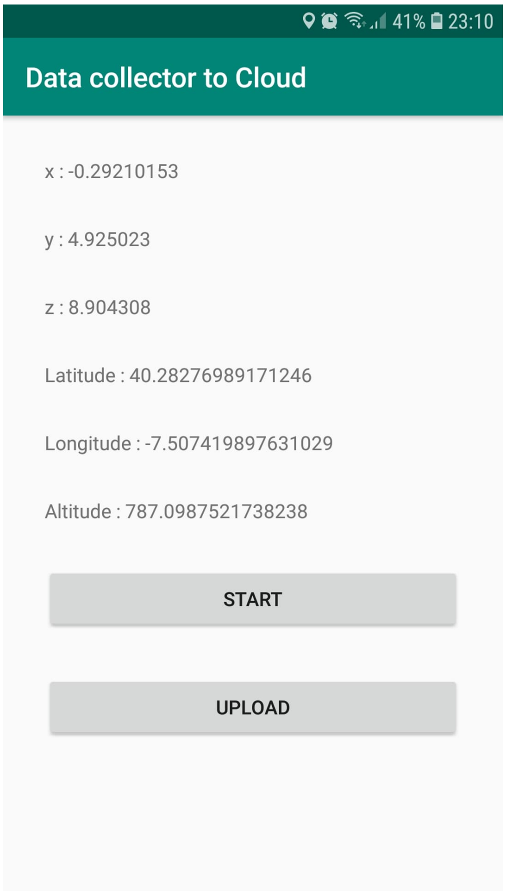

Fig.2. Last version of the application developed.

3.4 Data collection Method

In this section, it is presented an overview on haw we gathered all the data used to train the proposed supervised machine learning system.

The multiple versions of the applications created were installed into a smartphone (Samsung galaxy S7 edge) that was put inside a car (Renault Megane IV). Figure 3 shows the scheme for the placement of the smartphone inside the car.

Using the first version of the application, the phone was placed on the passenger's seat and the data gathered was of bad quality and unusable. One of the variables to be controlled is the positioning of the smartphone inside the vehicle, including finding a way to put the phone on a fixed position inside the car so it would not move and compromise all the data retrieved beforehand, so that the data is consistent.

The data collected was gathered on different scenarios: on wet and on dry roads and on four different types of roads: Cobblestone, Dirt, Tarmac and Highway.

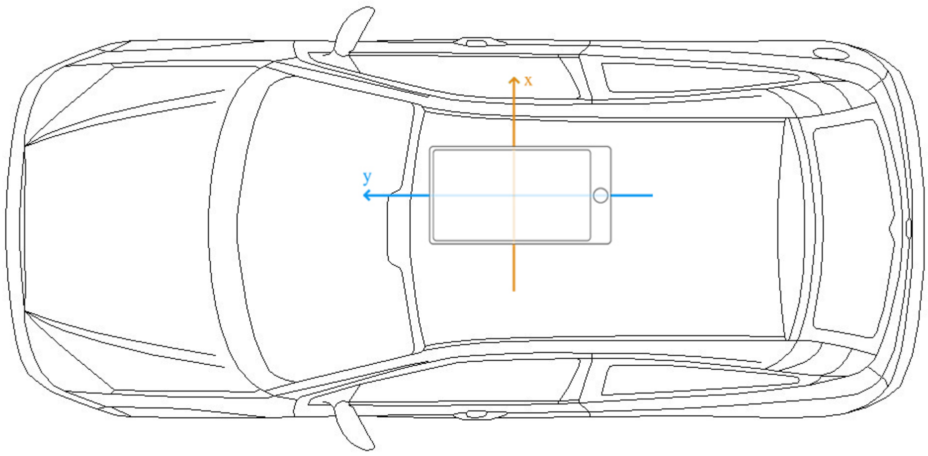

Fig. 3. Phone placement location in the vehicle (The phone is not to scale).

That was the first limitation of this research, another limitation is that the collected data is strongly dependent of the experiment's variables, such as for example, the brand, model and characteristics of the vehicle: the limitation will be discussed later in this dissertation.

After this step, it was necessary to prune the data files as to allow the data to have the same size to make it suitable for the feeding process of a machine learning system. This was the source of motivation on making the second version of the initial application.

With this second version of the proposed application, data files of a hundred lines (since it gathers ten seconds of data and the application collects ten measurements in a second) were collected. Figure4, shown below, is an example of the data that is collected by the application.

X: -0.36632404
Y: -2.901861
Z: 9.778218
Latitude: 40.28241948558828
Longitude: -7.502337137832754
Altitude: 679.9195704936343
Time: Mon Jun 08 23:32:39 GMT+01:00 2020

Fig. 4. Example of one line in the data files created by the application.

3.5 System design and methodology

This project is separated in three parts which will be discussed in this section.

The first one is about condensing and treating the data to make it easier for the Artificial Intelligence method to learn. This part is probably one of the most important parts because in order to make an AI method learn from the data acquired, it needs an immense amount of data and there was no public dataset that we could use to help us in this task. Since we had only one man gathering all the data, we couldn't just give the raw data to the AI, so we needed to treat the data before feeding it to the AI.

The second one is about which AI model we chose and how we trained it, in order to have a suitable model for the problem.

And, the third one is about which cloud computing service we chose and how we used it.

3.5.1 Data condensation

The first task is to condense the data that we collected so that it is easier to feed it to the machine learning model that we designed.

3.5.1.1 Collecting values

In this step, the goal was to extract the values of the accelerometer (X, Y, Z axis) and the GPS values (Latitude, Longitude, Altitude) and store them into two new excel files which have the same name as the original file but with an extension “-Accel” or “-GPS” to distinguish them from the original file as shown in Figure5. This treatment needed to be done on all files of course.

03-09-2020-2-3945PM.txt
03-09-2020-2-3945PM-Accel.xlsx
03-09-2020-2-3945PM-GPS.xlsx

Fig. 5. Example of a file that was treated and split into two different files.

3.5.1.2 Reduce the hundred lines files to only one line

For this step, the goal was to condense all the accelerometer values into one singular line, to make it easier for the machine learning model to classify the following events: acceleration, breaking and normal speed. This method was chosen because only one person alone was gathering all the data and gathered near one hundred and fifty files of data with the second version of the application.

There were applied different mathematical formulas to the data, on the distinguishing process between the three events, all the following are used on all the X, Y and Z axis:

- The first one was the average formula:

$$averageX = \frac{1}{n}\sum_{i=1}^{n} X_i \quad (1)$$

X_i represents the index i of the list of X axis values of length n.

- Then, I used the extend formula:

$$extendY = Max_Y - Min_Y \quad (2)$$

$Max_Y$and Min_Y represent respectively, the maximum and minimum of the Y axis values.

- Variance was used as well:

$$varianceZ = \frac{1}{n}\sum_{i=1}^{n} (Z_i - \overline{Z})^2 \quad (3)$$

Z_i represents the index i of the list of Z axis values of length n. $\bar{Z}$ represents the average of all the values on the Z axis.

- A modified version of the extend formula was also used:

$$extendY = Min_Y - Max_Y \tag{4}$$

- Finally, a modified version of the variance was used:

$$varianceZ = \frac{1}{n}\sum_{i=1}^{n}(Z_i - \overline{Z}) \tag{5}$$

After reflection on the problem, we felt that the formulas mentioned above were the one that were most likely going to distinguish the three different events mentioned beforehand.

Additionally, a formula was created to furthermore differentiate the values by multiplying the values by a coefficient that is related to the speed of the car that is calculated using the GPS files. We did it so that an acceleration or a breaking event would be easier to detect while driving at low speed. The formula used was:

$$Values \times (1 + \frac{SpeedThreshold - Speed}{SpeedThreshold}). \tag{6}$$

The speed threshold was set to 80 Km/h, so any speed below that value would result in a multiplication of all the values of the X, Y and Z axis by that formula. So, the lower the speed the greater the coefficient will be. We chose 80 Km/h as the threshold because we felt it was the value were there was no issues in detecting the different events.

Due to several learning problems of the AI model created by the lack of a larger dataset, we thought about using data augmentation or using only the axis that we need to focus on. We decided, after some tests, to only use the Y axis (since it is the one monitoring the acceleration or braking of the car) to train the AI model. Figure 6, shown below, is an example of all the subfiles created by the data condensation techniques talked above, for one file in the dataset.

03-09-2020-2-3945.txt
03-09-2020-2-3945-Accel-Average.csv
03-09-2020-2-3945-Accel-Average-Y.csv
03-09-2020-2-3945-Accel-Extend.csv
03-09-2020-2-3945-Accel-ExtendModified.csv
03-09-2020-2-3945-Accel-Extend-Y.csv
03-09-2020-2-3945-Accel-ExtendModified-Y.csv
03-09-2020-2-3945-Accel-Variance.csv
03-09-2020-2-3945-Accel-VarianceModified.csv
03-09-2020-2-3945-Accel-Variance-Y.csv
03-09-2020-2-3945-Accel-VarianceModified-Y.csv

Fig. 6. Example of a file that has been split into subfiles that were applied with different mathematical formulas.

3.5.2 Supervised machine learning model

The supervised machine learning model used at first was the K-Fold Cross Validation with 10 folds. To load the data, it was created a file, like the one in Figure 7, with the names of all the data files on each line and specified their class in the same line (1 -> Acceleration, 2->Breaking, 3 -> Normal Speed).

03-12-2020-12-3812PM-Accel-Extend.csv,1
02-23-2020-1-5952PM-Accel-Extend.csv,2
03-12-2020-12-5343PM-Accel-Extend.csv,3
02-06-2020-4-1419PM-Accel-Extend.csv,3
03-12-2020-12-3521PM-Accel-Extend.csv,1
02-05-2020-9-2549PM-Accel-Extend.csv,3
02-23-2020-2-4949PM-Accel-Extend.csv,2
01-24-2020-12-1338PM-Accel-Extend.csv,1

Fig. 7. Example of the files that were given to the supervised machine learning.

Due to a lack of good results, discussed later in the dissertation, we opted for a new approach where we fused the events "acceleration" and "breaking" into one event. Using this method, we used a Support Vector Machine (SVM) that called a binary classifier (OneVSRestClassifier) which we gave it 80% of the data for training and 20% of the data for testing.

To load the data, the file given to the model needed to change (1 -> Acceleration/Breaking, 2 -> Normal Speed), like the one in Figure 8.

03-12-2020-12-3812PM-Accel-Extend.csv,1
02-23-2020-1-5952PM-Accel-Extend.csv,1
03-12-2020-12-5343PM-Accel-Extend.csv,2
02-06-2020-4-1419PM-Accel-Extend.csv,2
03-12-2020-12-3521PM-Accel-Extend.csv,1
02-05-2020-9-2549PM-Accel-Extend.csv,2
02-23-2020-2-4949PM-Accel-Extend.csv,1
01-24-2020-12-1338PM-Accel-Extend.csv,1

Fig.8. Modified version of the Fig.7 applying the change previously mentioned.

3.5.3 Cloud computing services

Most of the work done in this dissertation was done on desktop, the cloud computing part is focused on storing the files created by the application created and on using the Google Maps API used in the event mapping application.
The cloud computing services that was used was Microsoft Azure due to the quality of the service and its cost, we did not have to pay to have an account (unlike Amazon Web Services). The service used is the "Storage account" that stores, into a container, the data files sent by the final version of the application, displayed in the Figure 9.

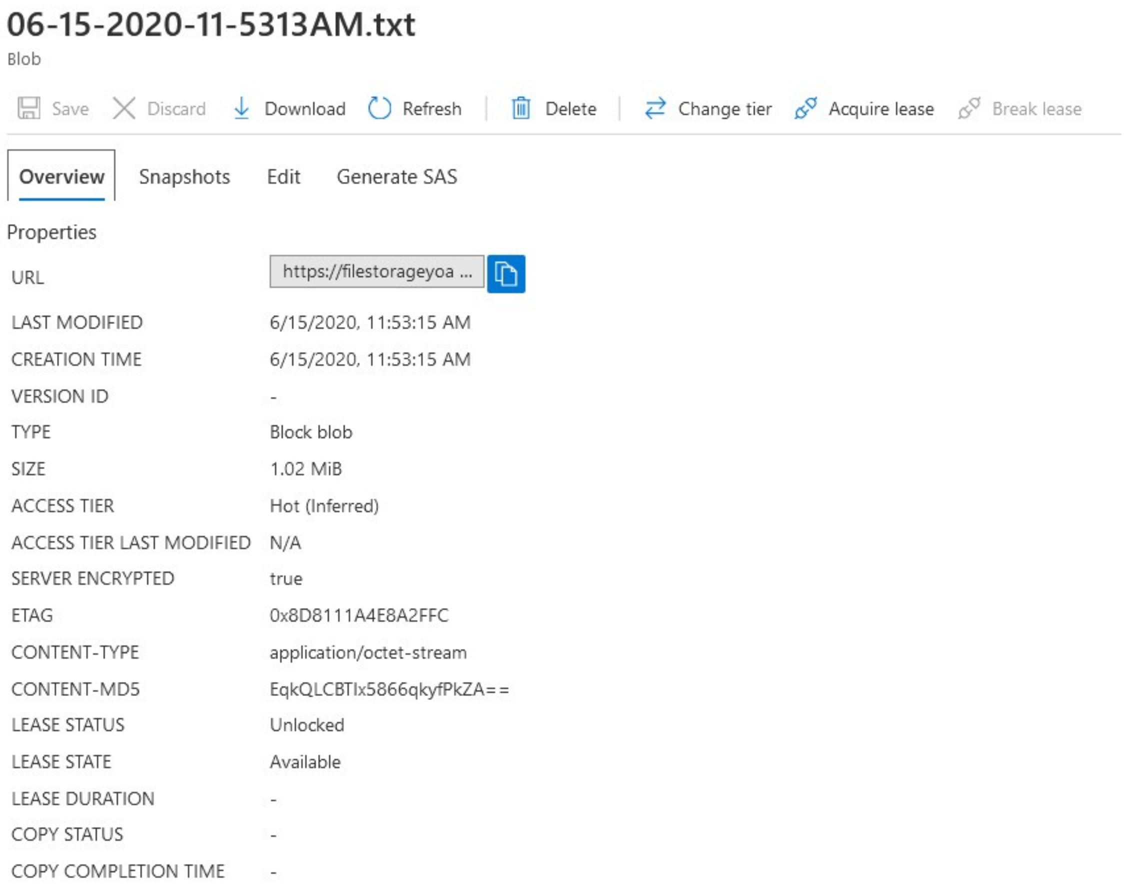

Fig. 9. Example of a file uploaded by the application to the Cloud.

We also used the Google Cloud services to allow us to use the Google Maps API in order to create the application that mapped the events detected by our system using heatmaps.

3.5.4 Data Processing and Analysis

At this state of the project, we are able to take a file of a hundred lines, treat it, and give it to the trained supervised machine learning model so that it can predict if it is an acceleration, a breaking or a normal speed related file. But the files collected by the final version of the application can produce files of thousands of lines. To address this problem, it was necessary to make a Java program that splits a file with over 100 lines into several files. For example, a file of 2565 lines will be split into 25 files of 100 lines and one file of 65 lines.

Of course, these files need to be condensed and must be reduced into one line using mathematical formulas as the other files mentioned above.

After applying the changes on the training of the supervised machine learning model, distinguishing the difference between an acceleration and a breaking event was necessary. A Python function that does this distinction was created. To explain the reasoning for this, we need to see graphs of the variations of the Y axis of the different speed events, displayed below with Figures 10, 11 and 12.

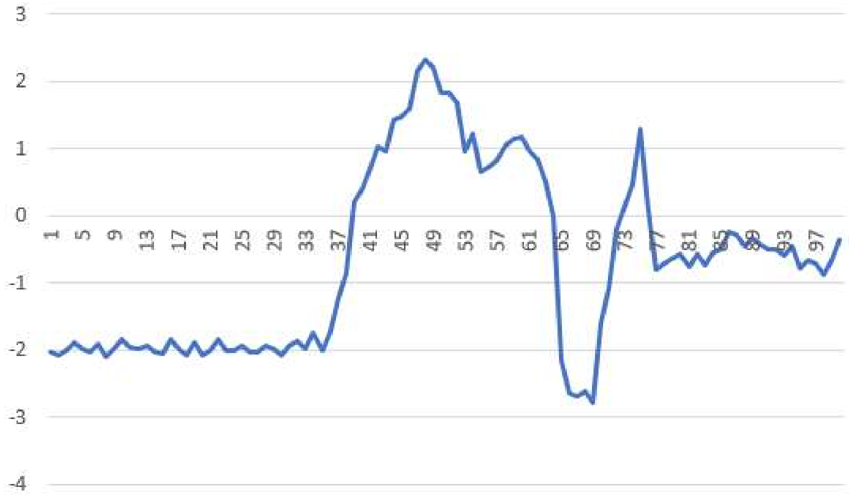

Fig.10. Example of graph of variation of Y axis for acceleration event.

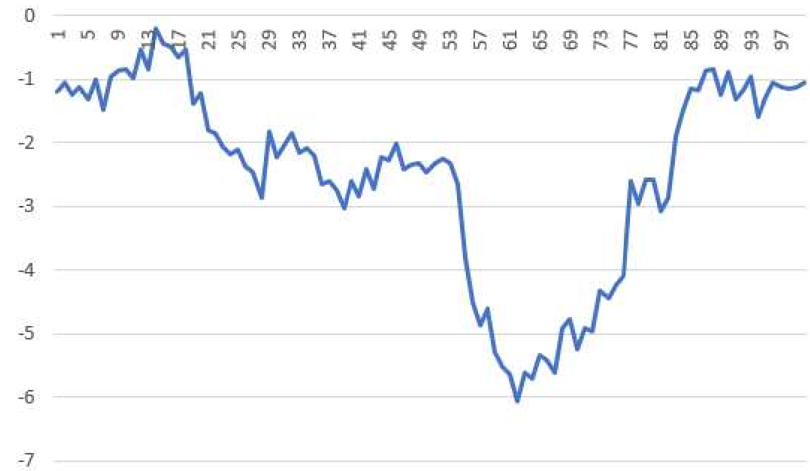

Fig.11. Example of graph of variation of Y axis for breaking event.

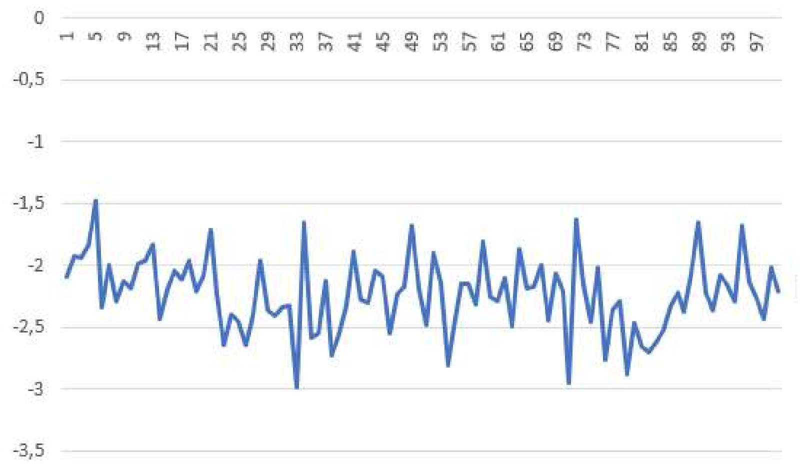

Fig.12. Example of graph of variation of Y axis for normal speed event.

The function, previously mentioned, is only called on events that were labeled as breaking or acceleration.
First, the function checks for obvious breakings or accelerations like if the minimum of the Y axis is lower than -3 or the maximum of the Y axis is greater than 1.
We decided to define these limit values when we looked at all of the data collected beforehand.
We noticed that, in most of our breaking data, we had the lowest value on the Y axis around -3.5 or even lower. The same goes for acceleration data, the highest value was around 1 or more. That's why we opted to go with those limit values. On the normal speed data, the values fluctuated between -0.5 and -2 most of the time.
Then, if the minimum is greater than zero, the event is obviously an acceleration since the average value of a normal speed and breaking is lower than at least -0.5. If the event is not obvious, we check the distance between the maximum value and 1, and the distance between the minimum value and -3. If the distance between the maximum value and 1 is the lowest then, it's an acceleration or else it's a breaking.

A Python function that identifies the presence of a road bump or pothole was also created. It takes all of the Z axis values that represent the elevation or the lowering of the smartphone and treats the data in the following way.

After the prediction of one event, thisfunction takes the index of the lowest value of Y and takes thethree values of the Z axis before the index of the lowest value of the Y axis and makes an average of those values.

It then sums upthe difference between the three following Z axis values and the average calculated before like in the formulas below.

$$averageBefore = (Z[indexMin_Y - 1] + Z[indexMin_Y - 2] + Z[indexMin_Y - 3])/3 \quad (7)$$

$indexMin_Y$ represents the index of the minimal value of the Y axis and Z[i] is the value of the Z axis in the index i.

$$total = (Z[index] - averageBefore) + (Z[index + 1] - averageBefore) + (Z[index + 2] - averageBefore) \quad (8)$$

If total is greater than 2 then we assume it is a road bump, or if total is lower than -2 then we assume it is a pothole. The accuracy of the prediction of road bumps and potholes will be discussed in the next chapter.

After giving the files to the trained model, it will create a text file with a listing of all the files and all of the events predicted by the model giving the event detected, GPS values for events that are not normal speeds and if it detects a road bump or a pothole. Such as the one presented in the Figure 13 below.

Result of Split.txt

```
06-08-2020-11-3520PM-1.txt: Normal speed
06-08-2020-11-3520PM-10.txt: Breaking Latitude: 40.27908146703283 Longitude: -7.501774768314352 Altitude: 650.2473251336096
06-08-2020-11-3520PM-11.txt: Breaking Latitude: 40.28031596330083 Longitude: -7.5011455073564814 Altitude: 668.5408008269573
06-08-2020-11-3520PM-12.txt: Breaking Latitude: 40.28127811389221 Longitude: -7.500952811332892 Altitude: 677.7470723388149
06-08-2020-11-3520PM-13.txt: Breaking Latitude: 40.2818500592639l Longitude: -7.501552599624629 Altitude: 677.9535231963863
06-08-2020-11-3520PM-14.txt: Breaking Latitude: 40.281881904423926 Longitude: -7.5018994903142815 Altitude: 677.986725151333
06-08-2020-11-3520PM-15.txt: Acceleration Latitude: 40.28265950733544 Longitude: -7.502341873925929 Altitude: 680.4037247852052
06-08-2020-11-3520PM-16.txt: Breaking Latitude: 40.28353147335941 Longitude: -7.50336663606234 Altitude: 670.9673598682264
06-08-2020-11-3520PM-17.txt: Acceleration Latitude: 40.28395525298493 Longitude: -7.504077683614671 Altitude: 671.5266578552382
06-08-2020-11-3520PM-18.txt: Breaking Latitude: 40.28394806129595 Longitude: -7.504617912339939 Altitude: 675.1346866619294
06-08-2020-11-3520PM-19.txt: Breaking Latitude: 40.28412778827089 Longitude: -7.505390283287307 Altitude: 689.369185695514
06-08-2020-11-3520PM-2.txt: Acceleration Latitude: 40.27702431910331 Longitude: -7.498836662400072 Altitude: 580.1868349981278
06-08-2020-11-3520PM-20.txt: Normal speed
06-08-2020-11-3520PM-21.txt: Acceleration Latitude: 40.284265085298145 Longitude: -7.505509849258193 Altitude: 690.679022457284
06-08-2020-11-3520PM-22.txt: Breaking Latitude: 40.28453438130632 Longitude: -7.50591122687376 2 Altitude: 700.0803302343206
06-08-2020-11-3520PM-23.txt: Breaking Latitude: 40.28474716864103 Longitude: -7.506310444016724 Altitude: 701.927939124186
06-08-2020-11-3520PM-24.txt: Breaking Latitude: 40.28544388822817 Longitude: -7.508793561790508 Altitude: 701.4813703489841
06-08-2020-11-3520PM-25.txt: Breaking Latitude: 40.285546010199155 Longitude: -7.5092997872396055 Altitude: 701.3304845624393
06-08-2020-11-3520PM-26.txt: Normal speed
06-08-2020-11-3520PM-27.txt: Breaking Latitude: 40.28362943463985 Longitude: -7.506318177434045 Altitude: 710.8093032405961
06-08-2020-11-3520PM-28.txt: Breaking Latitude: 40.28362943463985 Longitude: -7.506318177434045 Altitude: 710.8093032405961
06-08-2020-11-3520PM-29.txt: Breaking Latitude: 40.283772547126006 Longitude: -7.506905907608184 Altitude: 725.2629578090207 Road state: Road Bump
06-08-2020-11-3520PM-3.txt: Breaking Latitude: 40.27735179048072 Longitude: -7.499687561467375 Altitude: 597.8021864357975
06-08-2020-11-3520PM-30.txt: Breaking Latitude: 40.28368941691483 Longitude: -7.507068850206831 Altitude: 733.0598326927975
06-08-2020-11-3520PM-4.txt: Breaking Latitude: 40.27779887174264 Longitude: -7.499780333106245 Altitude: 612.6251480928163
06-08-2020-11-3520PM-5.txt: Breaking Latitude: 40.27898515872457 Longitude: -7.499582313922152 Altitude: 634.6591526376731
06-08-2020-11-3520PM-6.txt: Acceleration Latitude: 40.27912221949603 Longitude: -7.50010219097881 8 Altitude: 634.2528992945265
06-08-2020-11-3520PM-7.txt: Acceleration Latitude: 40.27820183683911 6 Longitude: -7.501542005210072 Altitude: 637.7347899062746
06-08-2020-11-3520PM-8.txt: Breaking Latitude: 40.27752913226097 Longitude: -7.502061055922978 Altitude: 640.5798648967645
06-08-2020-11-3520PM-9.txt: Breaking Latitude: 40.27720398311034 Longitude: -7.502927356052765 Altitude: 644.1914091615461
```

Fig.13. Example of a file returned by the program using the trained AI model

Figure 14shows a flow chart of the data collection.

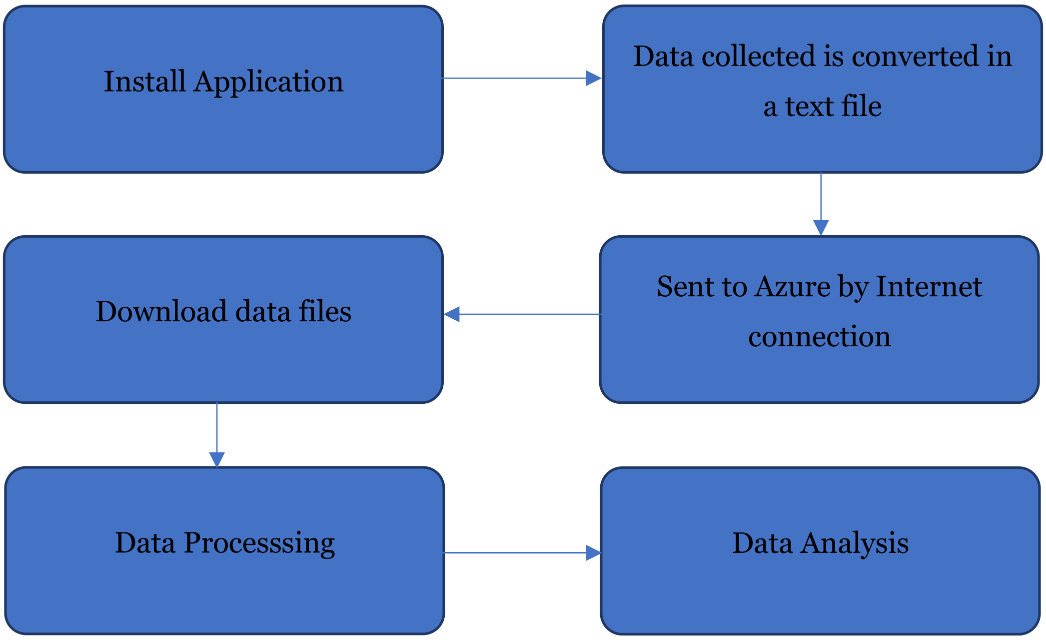

Fig. 14. Data collection Flow Chart.

After the data analysis, we upload then the result files to another container in the Azure Cloud. An Android application was created to display heatmaps of the events predicted by the AI model. This application retrieves the result files in the Azure Cloud and accesses the Google Maps API, through the Google Cloud Services, to display heatmaps of the events which we can select a specific one. This map allows you to zoom in and out and choose between the following events:

- Accelerations
- Breakings
- Accelerations and Breakings
- Potholes
- Road bumps
- Potholes and Road Bumps
- All of the events above.

The figures 15 and 16 below show the heatmaps for accelerations and breakings on car trips near the city of Covilhã, Portugal.

Fig.15. Heatmap of Acceleration events pinpointed by the Event Mapping Application

Fig.16. Heatmap of Breaking events pinpointed by the Event Mapping Application

Chapter4

4 Results and Discussion

To test the efficiency of the AI model trained, we used confusion matrices, and we also separated the files in two folders, one with all the data and the other with all the data except the data retrieved on cobblestone and on dirt roads. We made this decision because some events were hard to detect on those two types of road, moment in which we started training with all the formulas detailed earlier and got a model that predicted always only acceleration or normal speed, and never once a breaking event, depending on which formulas were being used. The accuracy of all the models trained are in the Table 1 below.
These values were trained using the K-Fold Cross Validation model which tried to classify 3 different events (acceleration, breaking and normal speed).

Table 1 – Comparative results between the accuracies of all the formulas using all the dataset and using only dataset of Highway and Tarmac only.

Formulas used	*Model trained with all of the data*	*Model trained with only data from Highway and Tarmac*
Average	34.5%	10.0%
Average Y axis	34.5%	20.0%
Extend	34.5%	46.7%
Extend Y axis	34.5%	06.7%
Extend Modified	34.5%	06.7%
Extend Modified Y axis	34.5%	20.0%
Variance	24.1%	40.0%
Variance Y axis	26.7%	31.0%
Variance Modified	34.5%	06.7%
Variance Modified Y axis	34.5%	06.7%

The models were almost always only predicting accelerations and sometimes normal speeds. After those results, we collected another fiftyish set of data, on a tarmac road only, that had mainly breaking events since it was not predicted by the model trained beforehand.
Adding this data and implementing the speed threshold formula when the vehicle's speed was lower than 80km/h allowed the supervised machine learning model to learn and it finally started to predict multiple events instead of just one. Still the results weren't good enough, we decided to merge the acceleration and breaking events into one single event and use a SVM model with a binary classifier.
And, this is where, we also decided to only focus on the Y axis since it is the only one useful to the model. As we saw in the last chapter, the Y axis is the one we need to monitor to see the variationsof speed of a car, that's why we dropped the otheraxis for this model.
After that, we started retraining models using all the formulas previously mentioned only using the Y axis values this time around. The results were a lot better in consequence of those changes, as you can see in Figure 17 below the example of the confusion matrix of the model trained using the variance formula on the Y axis values using only the set of data recorded on tarmac and on highway.

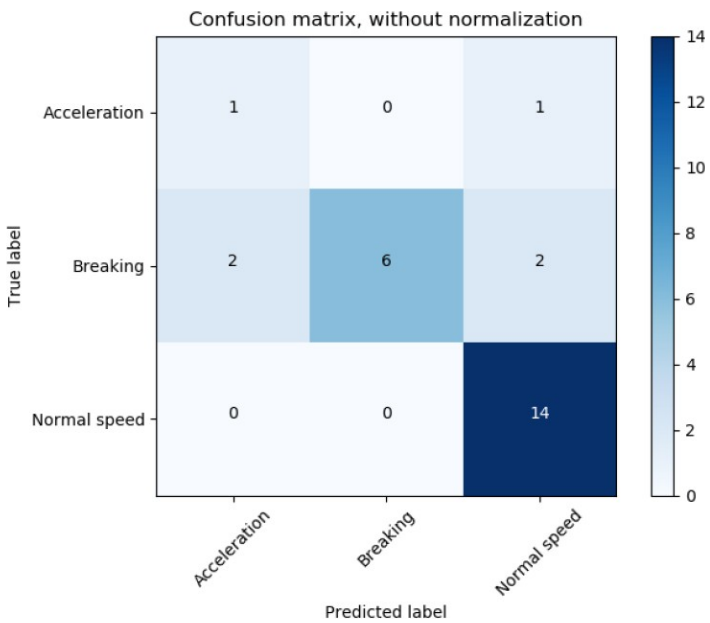

Fig.17. Confusion matrix of one of the models trained using only Y axis values and with 3 events to detect.

After checking this model's limitation to only 80.0% of accuracy, we rethought the problem and wanted to see if merging the acceleration and breaking events would help the model having a higher accuracy.
We also tried to test this new model using all the collected dataset to see if this change would allow us to use cobblestone and dirt road retrieved data without hurting the accuracy of the model.

Table 2 – Comparative results between Extend and Variance formulas using all the dataset.

Formulas used	*Road events*	*Accuracy*
Extend	Acceleration/Breaking	76.9%
	Normal speed	78.5%
	All	77.5%
Variance	Acceleration/Breaking	68.0%
	Normal speed	77.0%
	All	72.5%

Table 3 – Comparative results between Extend and Variance formulas using the data with only tarmac and highway roads.

Formulas used	*Road events*	*Accuracy*
Extend	Acceleration/Breaking	86.6%
	Normal speed	100.0%
	All	92.3%
Variance	Acceleration/Breaking	80.0%
	Normal speed	100.0%
	All	88.0%

Among the formulas used, the extend and variance formulas were the only ones with an acceptable accuracy. As we can see in the tables 2 and 3 above, this change did not allow us to use all the retrieved data without hurting the accuracy of the model (lower than 80.0%). Based on these results, we chose to use the extend formula applied to the Y axis model since it was the one with the highest accuracy.

The accuracy achieved by the model is really good due to the circumstances of the research (only one person gathering the data, limited time to retrieve the data, a lot of data was lost during the collection process due to smartphone sometimes moving out of place and mess the data retrieved).

After obtaining these results, we needed to test the program that splits the data and predicts the events on it by using the model trained and the code mentioned in the previous chapter. To test the accuracy of this program, we used some of the files we used to train the model (the table below shows the accuracy of it). Those files contained files of all of the events.

Road Events	*Predictions of Acceleration*	*Predictions of Breaking*	*Predictions of Normal Speed*
Acceleration	70.0%	00.0%	30.0%
Breaking	00.0%	84.4%	15.6%
Normal speed	00.0%	23.1%	76.9%

Table 4 - Comparative results between predictions of events of the system using a small amount of data.

As we can see in Table 4, the function used to differentiate the events “acceleration” and “breaking” is working well in predicting which event is which between the two (100% accurate in this example). The other predictions are the result of the model, this accuracy is lower than the 92.3% mentioned above because of the few files that we gave it.

Testing the pothole and road bumps detecting part was a bit more complicated due to the lack of data on it, we had 15 files of road bumps and 4 of potholes. The poor amount of data on potholes were due to the fact that finding potholes is complicated.

For the detection of road bumps, they were 11 out of the 15 that were detected as road bumps the other 4 did not detect any event.As for potholes 2 out of 4 were detected as potholes, the rest were not detected as potholes nor road bumps. For the potholes, the 2 files that were not predicted as potholes had totals just below the -2 threshold.
We cannot conclude if this method of detecting potholes and road bumps is reliable due to the lack of training and testing data.

We also tested out the system on very long files and noticed that the predictions on the events were accurate most of the time. There was a problem that we noticed concerning the prediction of acceleration events, it was related to the file splitting since these long data were retrieved on city roads with a lot of traffic and when there was an acceleration there was almost always also a breaking event inside the same file and we defined in this project that breaking was the priority between the two. If a file only contained an acceleration of three seconds and normal speed the rest of the time, the system would predict an acceleration event though.

Chapter 5

5 Conclusions and future work

The study of road pavement state and driving behavior is not new. In fact, several authors have published results that were reviewed in [4].

This work proposed a method to pinpoint a road event such as an acceleration or a breaking using a smartphone, having obtained a model that is 92.3% accurate in predicting those kinds of events. One of the limitations of this model is that it is 92.3% accurate on tarmac and highway roads and not on cobblestone nor dirt roads. Another is the method that detects the road bumps or potholes with non-conclusive results.

After discussing some results, more than we expected was discovered and raised further questions than the initial ones.Despite the fact that this project did not establish a model that can predict breaking or acceleration events of any type of car that is driving on any types of road pavement, it can still be refined and upgraded.

The results of this project showed that we can make systems that would work automatically and precisely detect road pavement status or even driving behaviors in certain driving areas, so that it can alleviate the work of road workers and help them make safer roads for everyone.

Regarding the title "Road Event Mapping Method for Mobile Devices with Cloud Computing based technologies", we can conclude that this method needs to be implemented everywhere so that we can drive on safe roads.

Further investigation will be needed to improve this system but it is clear that it is possible, in the future, to develop a better version of the mobile application created in this project that takes into account automatic reorientation of the data so the phone using the application can be placed wherever inside the car or even inside the pocket of someone in the car. It can also be improved by using a crowd sourcing method to gather a massive amount of data to strengthen the model build and make it so it can accurately detect events on every type of road pavement whilst using different types of cars.

Making a better detection method of road bumps and potholes like the one of [8] or [7] could be another upgrade to the system overall.

References

[1] A. Schroten *et al.*, *Overview of transport infrastructure expenditures and costs*. 2019.

[2] P. Nowak, "Intelligent Systems'2014," *Adv. Intell. Syst. Comput.*, vol. 323, no. September, pp. 781–792, 2015, doi: 10.1007/978-3-319-11310-4.

[3] Y. yuan Ren, X. sheng Li, Y. Ren, R. Guan, and W. wei Guo, "Study on Driving Dangerous Area in Road Curved Section Based on Vehicle Track Characteristics," *Int. J. Comput. Intell. Syst.*, vol. 4, no. 6, pp. 1237–1245, 2011, doi: 10.1080/18756891.2011.9727872.

[4] I. M. Pires and N. M. Garcia, "Identification of warning situations in road using cloud computing technologies and sensors available in mobile devices: A systematic review," *Electron.*, vol. 9, no. 3, 2020, doi: 10.3390/electronics9030416.

[5] N. Kalra, G. Chugh, and D. Bansal, "Analyzing Driving and Road Events via Smartphone," *Int. J. Comput. Appl.*, vol. 98, no. 12, pp. 5–9, 2014, doi: 10.5120/17233-7561.

[6] N. Silva, J. Soares, V. Shah, M. Y. Santos, and H. Rodrigues, "Anomaly Detection in Roads with a Data Mining Approach," *Procedia Comput. Sci.*, vol. 121, pp. 415–422, 2017, doi: 10.1016/j.procs.2017.11.056.

[7] J. Masino, J. Thumm, M. Frey, and F. Gauterin, "Learning from the crowd: Road infrastructure monitoring system," *J. Traffic Transp. Eng. (English Ed.*, vol. 4, no. 5, pp. 451–463, 2017, doi: 10.1016/j.jtte.2017.06.003.

[8] M. Y. Alam *et al.*, "Crowdsourcing from the True crowd: Device, vehicle, road-surface and driving independent road profiling from smartphone sensors," *Pervasive Mob. Comput.*, vol. 61, p. 101103, 2020, doi: 10.1016/j.pmcj.2019.101103.

[9] A. Vittorio, V. Rosolino, I. Teresa, C. M. Vittoria, P. G. Vincenzo, and D. M. Francesco, "Automated Sensing System for Monitoring of Road Surface Quality by Mobile Devices," *Procedia - Soc. Behav. Sci.*, vol. 111, pp. 242–251, 2014, doi: 10.1016/j.sbspro.2014.01.057.

[10] Y. Tai, C. Chan, and J. Y. Hsu, "Automatic Road Anomaly Detection Using Smart Mobile Device," *Proc. 2010 Conf. Technol. Appl. Artif. Intell.*, 2010.