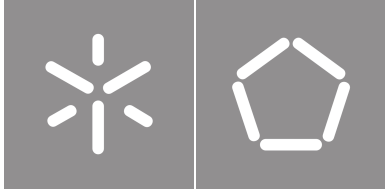
Universidade do Minho
Escola de Engenharia

Rita Alexandra Ferreira Pereira

Development of a classification algorithm for vehicle impacts: An Anomaly Detection approach

agosto de 2022



Universidade do Minho
Escola de Engenharia

Rita Alexandra Ferreira Pereira

Development of a classification algorithm for vehicle impacts: An Anomaly Detection approach

Dissertação de Mestrado
Mestrado em Engenharia Informática

Trabalho efetuado sob a orientação do:
Professor Doutor João Miguel Lobo Fernandes
Doutor André Leite Ferreira

agosto de 2022

Acknowledgements

I want to start by thanking my supervisor, Doctor Professor João Miguel Lobo Fernandes, and my mentor at Bosch Car Multimedia, S.A., Doctor Professor André Leite Ferreira, for all the support they provided and the availability they always showed to help me throughout the duration of my internship.

To Bruno Faria from the Small Damage Detection (SDD) team, thank you very much for all the time spent explaining to me the processes behind the developed concepts and for all the advice on the best way forward.

I would also like to also acknowledge and thank, in general, the SDD team who were always ready to give me suggestions and helped me integrate into BOSCH and understand the insights of a development team.

I want to thank my colleagues, Gabriel Santos and Bruno Nascimento, from the ALGO internship sub-team for sharing this experience with me and for all the sharing of ideas.

To my other colleagues, Carolina Gonçalves, João Sá, Ricardo Martins, and Luís Fontão, for making this experience much more enjoyable and fun.

To my boyfriend, thank you for your constant support and for the way you accompanied me in this academic journey. Without your support I would not be able to finish my master degree.

I want to thank my parents and sister for everything they taught me and for always believing in me.

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration.

I further declare that I have fully acknowledged the Code of Ethical Conduct of the Universidade do Minho.

Resumo

Development of a classification algorithm for vehicle impacts: An Anomaly Detection approach

Na última década, Machine Learning tem sido extensamente utilizado em soluções na indústria automóvel, o mais promissor sendo o desenvolvimento de veículos com condução autônoma.

Novos serviços de mobilidade estão disponíveis hoje como alternativas à posse de um carro, como *ride hailing* ou *car-sharing*. Os elevados custos associados à manutenção do veículo e a sua reduzida taxa de utilização ao longo do dia são alguns dos fatores que contribuem para a popularidade destes serviços.

Car-sharing é um modo de transporte *self-service* que fornece aos seus membros acesso a uma frota de veículos estacionados em vários locais duma cidade.

Danos são espectáveis de ocorrer quando os veículos são usados e a reparação necessária implica custos para os operadores da frota. Sistemas capazes de detectar esses danos irão promover um melhor aproveitamento desses veículos pelos utilizadores dos veículos.

Os danos de veículos resultam de impactos com outros objetos como, por exemplo, outros veículos ou estruturas e esses impactos provocam deformações na estrutura externa do veículo. A maioria desses impactos podem ser compreendidos ou detetados pelas forças envolvidas do resultado do impacto.

Anomaly Detection é uma técnica aplicável em uma variedade de domínios, como deteção de intrusões, deteção de fraude, deteção de eventos numa rede de sensores ou deteção de distúrbios no ecossistema.

O objetivo desta dissertação foi o estudo e desenvolvimento de um sistema inteligente semi-supervisionado para detecção e classificação de impactos de veículos a partir de uma abordagem de *Anomaly Detection*, utilizando os dados de acelerómetro, e seguindo uma estratégia que permitisse explorar um ciclo de Machine Learning.

Esta dissertação foi desenvolvida no âmbito de um estágio na empresa Bosch Car Multimedia S.A, situada em Braga.

Palavras-chave: Acelerómetro, Aprendizagem Semi-Supervisionada, Deteção de Anomalias, Impactos

Abstract

Development of a classification algorithm for vehicle impacts: An Anomaly Detection approach

In the past decade, Machine Learning has been heavily applied to automobile industry solutions, the most promising being development of autonomous vehicles.

New mobility services are available today as alternatives to owning a car, like ride hailing and car-sharing. High costs associated with the maintenance of the vehicle and the reduced rate of vehicle use throughout the day are some of the factors for the popularity of these services.

Car-sharing is self-service mode of transport that provides its members with access to a fleet of vehicles parked in various locations throughout a city.

Damages are expected to happen when vehicles are used and the required repair implies costs to fleet operators. Systems able to detect these damages will promote a better use of these vehicles by vehicle users.

Vehicle damages result from impacts with other objects, for instance, other vehicles or structures of any kind and these impacts inflict deformations to the vehicle exterior structure. Most of these impacts can be perceived or detected by the forces involved as result of the impact.

Anomaly Detection is a technique applicable in a variety of domains, such as intrusion detection, fraud detection, event detection in sensor network or detection ecosystem disturbances.

The objective of this thesis is the study and development of a semi-supervised intelligent system for detection and classification of vehicle impacts with an Anomaly Detection approach, using the accelerometer data, and following a strategy that would allow exploring a Machine Learning cycle.

This thesis was developed under an internship in the company Bosch Car Multimedia S.A, located in Braga.

Keywords: Accelerometer, Anomaly Detection, Impacts, Semi-Supervised Learning

Table of Contents

List of Figures

List of Tables

List of Algorithms

List of Acronyms

AI	Artificial Intelligence
AUC	Area Under the Curve
AW	After Window
BEV	Battery Electric Vehicle
BW	Before Window
CDF	Cumulative Distribution Function
CRISP-ML(Q)	Cross Industry Standard Process model for the development of Machine Learning applications with Quality assurance methodology
CRISP-DM	Cross Industry Standard Process for Data Mining
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DR	Detection Rate
DVA	Vertical Acceleration Impulse
DWT	Discrete Wavelet Transform
EDA	Exploratory Data Analysis
FAR	False Alarm Rate
FFT	Fast Fourier Transform
FN	False Negative
FP	False Positive
FPR	False Positive Rate
GMM	Gaussian Mixture Model
IMU	Inertial Measurement Unit
IVS	In-Vehicle Sensing
k-NN	k-Nearest Neighbors
LOF	Local Outlier Factor
LPG	Liquefied Petroleum Gas
LR	Logistic Regression
LRD	Local Reachability Density
MCC	Matthews Correlation Coefficient
ML	Machine Learning
MLP	Multi-layer Perceptron
NB	Naive Bayes
OCSVM	One-Class Support Vector Machine

PCA	Principal Component Analysis
PCB	Printed Circuit Board
PR	Precision-Recall
RF	Random Forest
RH	Relative Humidity
ROC	Receiver Operating Characteristic
RP	RankPower
SDD	Small Damage Detection
SMA	Signal Magnitude Area
SVC	Support Vector Classification
SVM	Support Vector Machine
TN	True Negative
TP	True Positive
TPR	True Positive Rate
UMAP	Uniform Manifold Approximation and Projection
WT	Wavelet Transformation

Chapter

1 Introduction

1.1 Context

The accelerated growth of population density in cities has a direct impact on the reduction in the capacity of urban mobility, which negatively affects the quality of life in cities, with significant direct and indirect costs to society.

In addition, there are several conditions that make owning vehicles in city contexts less and less interesting, such as mobility restrictions for vehicles, as well as the reduced rate of vehicle use throughout the day due to work schedules.

These factors lead to the appearance of new mobility services, such as ride hailing, for instance Uber or Lyft, and car-sharing, such as Sixt and ShareNow, which aim to increase the possibilities of choosing urban mobility without the hassle and expense of owning and maintaining a vehicle, because of all the requirements for it be allowed into the road, like insurance and annually government costs.

Car-sharing is a self-drive and self-service mode of transport that provides its members with access to a fleet of vehicles parked in various locations throughout a city. Car-sharing offers users access to a car that users typically pick up and return the car to fixed locations around the city, and, using an automated check-in and check-out system, are charged for the time used.

Car-sharing is a type of car-rental. However, in traditional car-rental schemes there is a contact between a providers agent and the client to find a car that best suits the requirements of the client and the whole process of reservation, pickup and return is made with physical contact. In car-sharing the process is completely self-service.

The client logs into his account in the car-sharing providers app, searches for the car and define the rental schedule. When it is time to pickup the car, the client goes to the location of the car and then just needs to be close to the car and is given access via a virtual key that needs to be configured on the car system or the car unlocks via remote action. The return of the car is also made with no contact with an agent, in fixed locations around the city.

1.2 Problem

The problem with car-sharing lies in the fact that typically there is not a regular supervision of the vehicle, which results in scratches, for instance, that are impossible to pinpoint to a single client.

Given that the vehicles are spread throughout the cities and the service provider is not required in the process because of it being completely automated, the identification of damages can sometimes be made too late.

There is no control on what the vehicle condition status is after every customer use when there is a requirement to certify if a vehicle is in a condition that allows it to be lend to another customer in a legal and safe state, specially addressing the exterior condition of the vehicle: external hood damage, scratches, etc. In the normal car-rental process, after every use, an inspection is made by a professional of the company to check the vehicle condition.

The safety and confidence of the clients is the biggest worry for the providers. A client driving through a bump in high speed can result in damage on the vehicle that could compromise the client and the following users of the vehicle.

1.3 Objectives

Given the context, it was relevant to equip the vehicle with a setup of acquisition, which consists of a system of sensors, that enabled the service providers to better detect when impacts occur.

In this thesis there is a study of an Anomaly Detection system with a semi-supervised approach for detection and classification of vehicle impacts using accelerometer data.

Taking into account that impacts are abnormal events on the roadway, the analysis of an Anomaly Detection approach to the problem was contemplated to be a possibility that could lead to good results, specially when considering the application of the Anomaly Detection algorithm as a filtering mechanism.

Additionally, considering the importance of a well structured project of a Machine Learning (ML) system, the understanding and the application of a ML cycle is also appointed as an objective of this thesis. To better conduct the process of implementing an Anomaly Detection project, a well carried out method on all phases is fundamental to reach the best possible results.

1.4 Document Structure

This thesis document is divided into the following chapters:

- State of the Art has the theoretical analysis of anomaly detection, describing the definition of concepts necessary to understand the problematic of anomaly detection and a review of literature.
- Machine Learning Process where the model process behind a generalized and industrialized Machine Learning project is explained.
- Use Case: EasyRide there is a definition of additional work that was performed on another project in order to fully comprehend and execute the ML cycle.
- Use Case: SlimScaley - Business and Data Understanding where the phases corresponding to the Data Collection and Exploratory Data Analysis (EDA) are performed to understand the data.

- Use Case: SlimScaley - Data Preparation and Modeling phases is the final chapter where from the development of the dataset to the implementation of the Anomaly Detection solution is outlined.
- Conclusion and Future Work is the chapter where an interpretation of the results is performed and follow-up strategies are outlined in order to possibly improve the outcome of the application of the Anomaly Detection strategy.

Chapter

2 State of the Art

This chapter presents the theoretical foundations about the main fields covered in this Master thesis. First, a brief description of concepts, like anomaly and outlier, and their differences are presented and, subsequently, a specification of important aspects to consider before choosing the Anomaly Detection technique, such as the different types of anomalies and the data used.

The main objective of this chapter is to give a complete review of the base theoretical background of Anomaly Detection which is fundamental to understand the application of Anomaly Detection in the problem introduced. A review of literature is also presented.

2.1 Theoretical Concepts

The relevant terms and concepts applicable to understand the problematic of Anomaly Detection are explained in this section.

2.1.1 Anomaly Detection

An important notion is the distinction of anomalies and outliers. Hawkins (1980) defines outlier as an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism.

Aggarwal (2016) presents in his book the terms that are also used to reference outliers and a citation from that paragraph is mostly referenced to prove the equality of both concepts: "Outliers are also referred to as abnormalities, discordants, deviants, or anomalies in the data mining and statistics literature."

However, in the same book, Aggarwals explains that "the term "outlier" refers to a data point that could either be considered an abnormality or noise, whereas an "anomaly" refers to a special kind of outlier that is of interest to an analyst". So Aggarwal regard outliers as a broader concept which also includes noise in addition to anomalies, as in the article of Salgado et al. (2016).

The terms anomaly and outlier are used interchangeably in a lot of research about Anomaly Detection but, when used, the term outlier is always not inclusive of noise. The difference between noise and anomalies are explained later in this section.

In this thesis the terms anomalies and outliers are not used interchangeably, with outlier being the combination of anomalies and noise data.

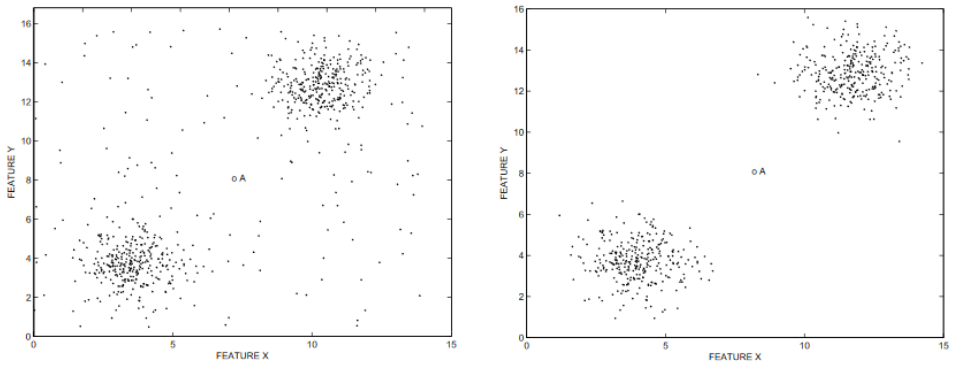
One of the most common definitions of anomalies is the following: "Anomalies are patterns in data

that do not conform to a well defined notion of normal behavior"(Chandola et al., 2009).

Some important characteristics of anomalies are:

- The distribution deviates remarkably from the general distribution of the data.
- Anomalies are rare data points of the dataset.

Another important notion that has already been presented is the distinction between anomalies and noise.



a With noise b No noise

Figure 1: Noise and anomalies: the difference.

Noise can be a mislabeled example (class noise) or errors in the attributes of the data (attribute noise) (Salgado et al., 2016).

As seen in Figure 1, the distribution of the main two clusters is the same. In Figure 1a the anomaly, marked with an A, is difficult to distinguish from the rest of the noisy points. But in Figure 1b the anomalous point seems to be obvious as it deviates significantly from both clusters.

Another important concept to distinguish from anomaly is novelty. Novelty patterns are data points that haven't been previously observed. The distinction between novel patterns and anomalies is that the novel patterns are typically incorporated into the normal model after being detected (Miljković, 2010). Most methods used in Anomaly Detection are also used in Novelty Detection.

A definition of Anomaly Detection can be given: Anomaly detection refers to the problem of finding patterns in data that do not conform to expected behavior (Chandola et al., 2009).

The concept of anomalies and how to detected them has been studied for many years in the statistics community. In fact, in the 19^{th} century, Edgeworth (1887) studied and researched the aspect of anomalies, naming it "discordant observations"and how to proceed with the "treatment of such observations".

It is important to differentiate between Anomaly Detection and Forecasting. Forecasting focuses on the established pattern of data distribution (Lazzeri, 2020). In contrast, anomaly detection focuses on the data points that deviate from what is expected.

Chandola et al. (2009) has listed the main challenges about Anomaly Detection:

- Defining a normal region that encompasses every possible normal behavior is very difficult. In addition, the boundary between normal and anomalous behavior is often not precise. Thus, an

anomalous observation that lies close to the boundary can actually be normal, and vice versa.

- In many domains, normal behavior keeps evolving and a current notion of normal behavior might not be sufficiently representative in the future.
- The exact notion of an anomaly is different for different application domains. For example, in the medical domain, a small deviation from normal (for example, fluctuations in body temperature) might be an anomaly, while similar deviation in the stock market domain (for instance, fluctuations in the value of a stock) might be considered as normal. Thus, applying a technique developed in one domain to another, is not straightforward.
- Availability of labeled data for training/validation of models used by anomaly detection techniques is usually a major issue.
- Often the data contains noise that tends to be similar to the actual anomalies and hence is difficult to distinguish and remove.

Due to these challenges, an Anomaly Detection problem is not easy to solve. To select the appropriate approach to resolve the problem of anomaly detection, some properties are important like the type of data, the availability of labeled data and types of anomalies. These aspects are described in the following sections.

2.1.2 Structure of anomalies

The nature of the anomalies can be classified into three categories: Point Anomalies, Contextual Anomalies and Collective Anomalies (Chandola et al., 2009; Prado-Romero et al., 2016; Song et al., 2007; Zhao et al., 2020).

An instance is described as a point anomaly when it is considered as anomalous with respect to the rest of data. This is the simplest type of anomaly and is the focus of the majority of research on anomaly detection.

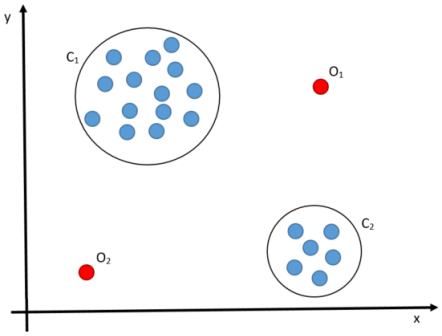


Figure 2: Example of two points anomalies.

Figure 2 gives a simple representation of point anomalies: O_1 and O_2. The points O_1 and O_2 are far from the rest of the data, grouped on C_1 and C_2.

An instance is described as a contextual anomaly if it is anomalous in a specific context (but not

otherwise), then it is termed as a contextual anomaly, also referred to as conditional anomaly. Each instance is composed of contextual attributes and behavioral attributes.

1. Contextual attributes are used to determine the context for that instance.
2. Behavioral attributes define the non-contextual characteristics of an instance.

Contextual anomalies are most common on time series data or spatial data.

A normal behavior on a context can be anomalous when in another context. The temperature of 30°C in Portugal, without any context, cannot be seen as an anomaly. For instance, in Portugal a temperature of 30°C in the summer is normal. A temperature of 30°C in the winter is anomalous. The spatial-temporal aspect can also be used in this example. Snow is normal on the highest point of Continental Portugal, the Serra da Estrela, in the winter. However, snow in Beja in the winter is anomalous.

Is important to state that anomalies are not necessarily impossible events, but unlikely/rare under normal conditions.

If a collection of related data instances is anomalous with respect to the entire data set, it is termed as a collective anomaly. The individual data instances in a collective anomaly may not be anomalies by themselves, but their occurrence together as a collection is anomalous.

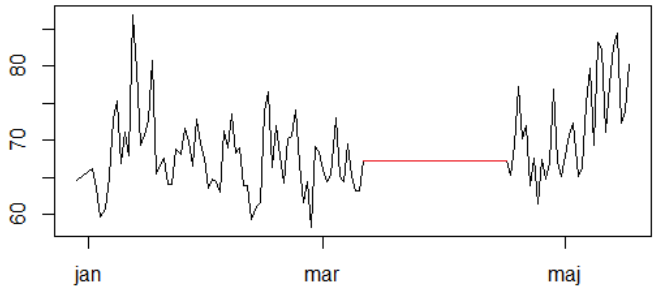


Figure 3: Example of a Collective Anomaly.

Looking at figure 3, one can conclude that the data represented by the red line is anomalous. The respective value itself is not anomalous as there's other instances that record the same value. The set of values is what causes that group of records to be anomalous.

If considering a context, a point anomaly detection problem or a collective anomaly detection problem can become a contextual anomaly detection problem, as described on the Table 1.

		Data Grouping	
		No	Yes
Data Context	No	Point Anomaly	Collective Anomaly
	Yes	Contextual Anomaly	Contextual Anomaly

Table 1: Taxonomy of the types of anomalies.

2.1.3 Performance Metrics

A confusion matrix (Fawcett, 2006), also called a contingency table, forms the basis for many common metrics. The concept of contingency table was first introduced by Pearson (1904).

		Real	
		Anomaly	Normal
Predicted	Anomaly	True Positive	False Positive
	Normal	False Negative	True Negative

Table 2: Confusion Matrix.

The Confusion Matrix presented on Table 2 represents a generalized application of an Anomaly Detection algorithm, where the positive class is the one represented by anomalous data, and the negative class is the normal data.

When an anomaly detection technique is applied three things can happen (Mehrotra et al., 2017):

- Correct Detection: detected anomalies in data correspond exactly to real anomalies. In most of real live systems, a 100% correct detection is impossible. On the Confusion Matrix represented on Table 2, the corrected detection is equivalent to combination of the True Positive (TP) and True Negative (TN) cases.
- Presence of False Positive (FP) and none False Negative (FN): the process continues to be normal, but unexpected data values are observed.
- Presence of FN: the process becomes abnormal, but the consequences are not registered in the abnormal data.

The objective is to minimize both FPs and FNs. Depending on the problem, it could be more beneficial to allow a higher value of FPs than FNs, and vice-versa. An analysis should be carried to make this decisions. For example, in a medical disease diagnosis it is more beneficial to have more FPs than FNs, however not a big imbalance.

The simpler and widely used performance metric that allows to evaluate the performance of an algorithm is the accuracy.

$$Acc = \frac{\text{True Negatives} + \text{True Positives}}{\text{True Positives} + \text{True Negatives} + \text{False Positives} + \text{False Negatives}} = \frac{\text{Correct Prediction}}{\text{Total Data}}$$

Accuracy is not a metric which can be applied to evaluate an anomaly detection technique.

Given that anomalies are few and rare, an example of an anomaly detection problem sometimes consists on 99.9% of normal data and only 0.1% of anomalous data. A simple technique that evaluated all the data on this problem as normal would have 99.9% of accuracy, an impressive value for an evaluation of a technique but a fact that is not desired on the anomaly detection problem.

The metrics Precision, Recall, RankPower (Tang et al., 2007; Tharwat, 2020), Receiver Operating Characteristic (ROC) Curve and Precision-Recall (PR) Curve defined below, are often used to evaluate the performance of the anomaly detection algorithm.

Given a dataset $\mathscr{D}$, suppose an anomaly detection algorithm identifies s>0 potential anomalies, of which $s_t(\leqslant s)$ are known to be true anomalies and having n true normal data. Then Precision, which measures the proportion of true anomalies in top s suspicious instances is:

$$P_r = \frac{s_t}{s}$$

Precision equals 1.0 when all the instances identified by the algorithm are true anomalies.

Taking into consideration the information on Table 2, Precision can be defined as:

$$P_r = \frac{\text{True Positive}}{\text{True Positive + False Positive}} = \frac{\text{True Positive}}{\text{Total Predicted Positive}}$$

Precision is a good metric when the cost of FP is high.

If $\mathscr{D}$ contains $d_t(\geqslant s_t)$ true anomalies, then Recall is defined as:

$$R_e = \frac{s_t}{d_t}$$

Recall equals 1.0 when all true anomalies are identified by the algorithm.

$$R_e = \frac{\text{True Positives}}{\text{True Positives + False Negatives}} = \frac{\text{True Positives}}{\text{Real Anomaly}}$$

Recall, also known as True Positive Rate (TPR) and sensitivity, is recommended to be the model metric when there is a high cost associated with a False Negative.

Precision and recall can be combined into a single score that seeks to balance both concerns, called the F-score or the F-measure.

$$F - Measure = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision + Recall}}$$

The Fbeta-Measure is an abstraction of the F-measure where the balance of precision and recall in the calculation of the harmonic mean is controlled by a coefficient β. F-score is a specific case of Fbeta-Measure where β = 1.

$$Fbeta - Measure = \frac{(1+\beta^2) * Precision * Recall}{\beta^2 * Precision + Recall}$$

Another really import metric is called RankPower (RP). RP was proposed by Tang et al. (2007) and evaluates the ratio of the number of known anomalies and anomalies returned by an algorithm, along with their rankings. The difference between the previous metrics and Rank-Power is that the preceding do not give any preference to the ranks, that is, how anomalous is a particular sample.

Formally, if R_i denotes the rank of the ith true anomaly in the sorted list of most suspicious objects, then the RP is given by:

$$RP = \frac{s_t(s_t+1)}{2\sum_{i=1}^{s_t} R_i}$$

RP takes maximum value 1 when all d_t true anomalies are in top d_t positions.

False Positive Rate (FPR), also known as specificity, can be defined as follows:

$$FPR = \frac{s - s_t}{n}$$

The metric RankPower only takes into consideration the abnormal class, leaving the normal class with no evaluation consideration.

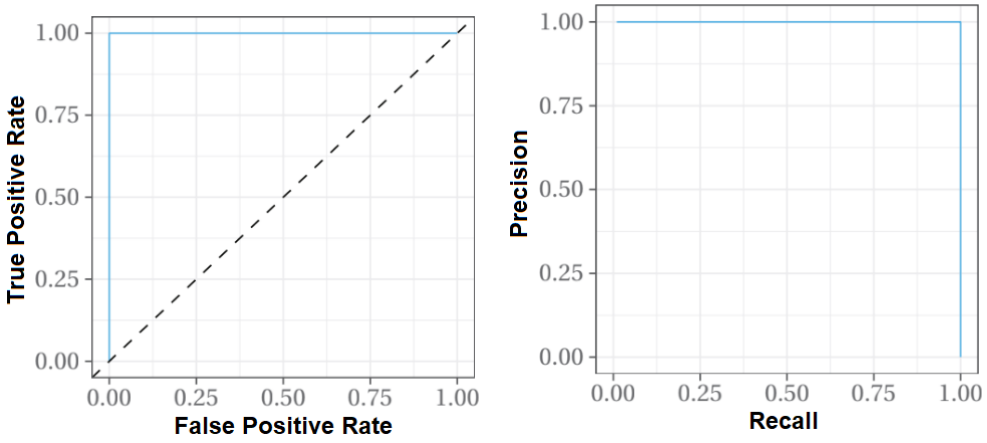
A popular graphical plot that characterizes binary classifiers is the ROC Curve. ROC graphs are two-dimensional graphs in which TPR is plotted on the Y axis and FPR is plotted on the X axis (Fawcett, 2006).

Area Under the Curve (AUC) is computed using a trapezoid rule and is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one. The score is a value between 0.0 and 1.0, the latter meaning a perfect classifier.

Although generally effective, the ROC Curve and AUC can be optimistic under a severe class imbalance, especially when the number of examples in the minority (anomaly) class is small.

An alternative to the ROC Curve is the PR Curve that can be used in a similar way, although focuses on the performance on the minority (anomaly) class.

PR Curve is a plot of the precision, in the Y axis, and recall, in the X axis, for different probability thresholds.



(a) ROC Curve (b) PR Curve

Figure 4: (Schlegel, 2019): Example of a perfect evaluation by a ROC Curve (4a) and PR Curve (4b)

Either in the ROC Curve or the PR Curve (Figure 4), the best model bows towards the coordinate (1,1).

Matthews Correlation Coefficient (MCC) is a metric introduced by Matthews (1975), extensively used on the Machine Learning (ML) field.

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

Considering the equation above that represents the MCC calculation, this metric returns a value between -1 and 1, where 0 is a random estimate.

MCC is a metric that weigh all the outcomes of the algorithms, lowering the evaluation score from FP, as well as FN. F-score, a measure explained before, equals zero when the TP of the result of applying the classifier is zero, being independent from the value of TN and is not symmetric if a class swapping occurs.

The key advantage of the MCC is that it generates a high quality score only if the prediction correctly classified a high percentage of negative data instances and a high percentage of positive, being considered a balanced measure (Chicco et al., 2020). This aspect of the MCC metric, as well as the limitations of other metrics explained formerly, are arguments for applying this technique, over others, in an Anomaly Detection approach.

2.1.4 Supervised, Unsupervised and Semi-Supervised Learning

The Anomaly Detection techniques can be one of the following three approaches, based on the the availability of labels in the data:

- Supervised. Labels available for both normal data and anomalies. The algorithm distinguish between normal and known anomalous instances.
- Semi-supervised. Labels available only for normal data. The training data has labels on non anomalous (normal) instances that are then trained to be able to find the anomalies. The algorithm learns the normal behavior and then detect any deviations from normal behavior as anomalous.
- Unsupervised. No labels assumed. This approach is based on the assumption that anomalies are very rare compared to normal data. If this aspect is not found on the data, a unsupervised approach cannot be applied successfully.

	Supervised	Semi-Supervised	Unsupervised
Require Prior-Knowledge	Yes	Yes	No
Environment	Static	Dynamic	Dynamic
Detection Speed	Fast	Fast/Moderate	Moderate/Slow
Detection Generality	No	Yes	Yes

Table 3: Difference between Supervised, Semi-Supervised and Unsupervised Learning.

The Table 3 gives a brief resume of the main differences between the different approaches. Because of the difficulty that manually labeling data implies, semi-supervised and unsupervised learning are two of the most used approaches, besides the possibility of finding new anomalies because of its ability to generalize.

2.1.5 Output of Anomaly Detection

The output of anomaly detection techniques is an important aspect of it, influencing a lot of decisions in terms of which technique to use, depending on the desired output type. Typically, the outputs produced by anomaly detection techniques (Gao et al., 2006; Kriegel et al., 2011; Mehrotra et al., 2017) are one of the following two types :

- Scores. Scoring based anomaly detection techniques assign an anomaly score to each instance

in the test data depending on the degree to which that instance is considered an anomaly, that is, the test data more likely to be an anomaly has a higher associated value (score) than a likely normal instance. There are many advantages to transforming the output scores into well-calibrated probability estimates. The analyst can analyze the top anomalies or use a domain-specific threshold to select the most relevant anomalies.

- Binary Labels. Techniques in this category assign a label (normal or anomalous) to each test instance. Although some algorithms might directly return binary labels, anomaly scores can also be converted into binary labels. This is the typical output of classification-based approaches.

2.2 Literature Review

This section provides a review of articles about Anomaly Detection applied to vehicle data. The articles mention road condition, driver behaviour and accident detection based methods using Anomaly Detection. Besides the analysis of Anomaly Detection algorithms applied to the context mentioned, firstly there is gonna study of some of the preprocessing techniques applied.

Raw data is sometimes difficult to analyze and understand so it needs to be preprocessed, adding new features or even transforming the data, allowing for an easier to understand data and increasing its usability in algorithms.

A ri-orientation procedure using the Euler Angles in a stationary condition in which the only acting force is the gravity (acceleration along the z axis), was carried out by Astarita et al. (2012). Using the XYZ sequence, the equations for the calculation of the roll and pitch angles have the following expressions:

$$\alpha = \arctan\left(\frac{a_y}{a_z}\right)$$

$$\beta = \arctan\left(\frac{-a_x}{\sqrt{{a_y}^2 + {a_z}^2}}\right)$$

The article (Vittorio et al., 2014) also performed this calculation. In this article, the evaluation of the road surface quality was based on the analysis of the Vertical Acceleration Impulse (DVA): the temporal derivative of the acceleration absolute amplitude in one second $d(a_{z-max} - a_{z-min})$, that was calculated as the difference between the maximum and the minimum values of the vertical acceleration in the defined unit of time. The DVA corresponds to a high-energy event, that is labelled as an “anomaly” on the road surface. The ri-oriented accelerometer data were also filtered by means of a post-processing algorithm in order to remove low frequency components in the signal due to the background noise.

In Douangphachanh et al. (2014) the data consisted of only acceleration data (x, y, z) from an accelerometer and location data, including speed, from GPS, collected from smartphones, and it is checked and matched with referenced data before dividing into 100 meter sections. A high pass filter is used to remove unrelated low frequency signal, which is contributed by the effect of vehicle maneuver such as braking and turning as well as the contribution of the force of gravity, from all axes of the acceleration data. After sectioning, road sections that have incomplete data will be excluded from the analysis. Road sections where experiment vehicles have stopped are also excluded.

Bello Salau et al. (2015) extracted the mean, standard deviation and variance of accelerometer data over a sliding window. The energy content (E) of the measured time series signal $x(n)$ was also computed using the following equation:

$$E = \sum_{i=0}^{N-1} \left(|x(n)|\right)^2$$

The authors noted that areas with road defects will have higher energy content than other anomaly free portions of the dataset, where the threshold values were manually set to identify portions of higher energy content, and hence, indicating of the presence of possible road defects within the dataset.

Bello Salau et al. (2018) used a Wavelet Transformation (WT) theory and a noise filtering technique to characterize road anomalies into either potholes or bumps. The filtering technique was used to filter noise from acceleration signals noting that noisy samples have lower correlation values across an increasing WT scale.

Some articles that used Anomaly Detection techniques are next reviewed in order to best define the possible algorithms to solve the problematic.

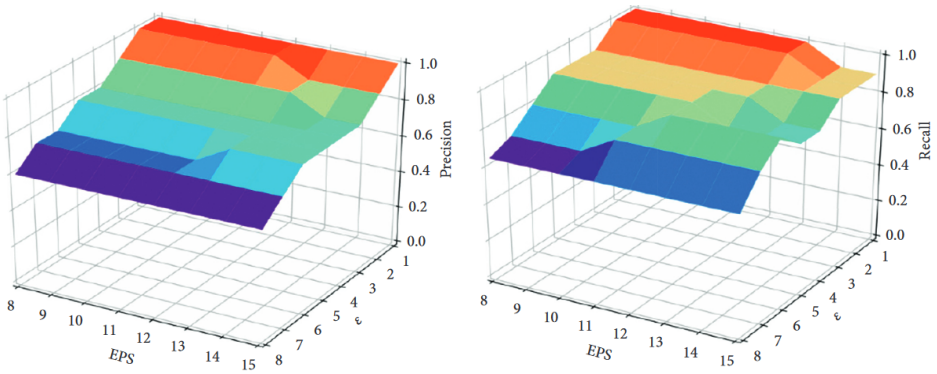
In the first article reviewed, Yoon et al. (2007) developed a threshold-based clustering algorithm customized for their data, called threshold-based quadrant clustering since general clustering algorithms do not work with unlabeled GPS data. The data was divided into 4 quadrants with each one indicating a different kind of condition. The first quadrant indicates a both spatially and temporally good traffic—steady travel at good speed. The second quadrant represents a spatially good but temporally bad one. Similarly, the fourth quadrant depicts good average speed, but with some slow-and-go periods. The last quadrant shows poor traffic conditions.

Eriksson et al. (2008) analyzed the abnormal behaviors of taxis, including detour behavior, speed anomaly, and local shape anomaly. The dataset used contained the GPS trajectory data of 10357 taxis in Beijing for the period from February 2 to February 8, 2008. According to the different anomalies causing abnormal taxi trajectory, different solutions were proposed:

- Global router anomaly detection algorithm: the similarity between trajectories in the same deme is used as the input to the Isolation Forest (iForest) algorithm that trains a suitable model to determine global router anomaly trajectories.
- Local speed anomaly detection algorithm: the instantaneous velocities of trajectory points are clustered by Density-Based Spatial Clustering of Applications with Noise (DBSCAN) for each road section. A trajectory having a sufficient number of speed anomaly points will be marked as a local speed anomaly trajectory. Because of the large number of clusters in this dataset, they selected clustering results of three road sections between 13:00 and 14:00 on February 2, 2008.
- Local shape anomaly detection algorithm: the direction deflection angle of each trajectory point is calculated. The deflection angle of the trajectory point on the same road section is used as the input of the Local Outlier Factor (LOF) algorithm to determine the abnormal trajectory of the lane change. The parameters used were k = 5, d = 0.5 and f = 0.15.

The metrics used were precision and recall rates. The number of abnormal trajectories in the precision

and recall rate calculation were obtained by the union of the results of the three algorithms. The results are presented in Figure 5.



(a) (b)

Figure 5: Precision (a) and Recall (b) of the algorithm (Eriksson et al., 2008).

Zhu et al. (2009) proposed a distance-based Anomaly Detection method that used the Euclidean Distance to decide if a specific point is an anomaly or not. Through analysis of road condition on urban arterial road, traffic abnormality can be recognized as anomaly. Given that an incident causes a reduction of roadway capacity, it will possibly be detected as an anomaly. After the Euclidean distance is calculated, Zhu et al. proposed two strategies:

- Assign a threshold parameter $\widetilde{\varphi}$ and calculate the average distance:

$$\overline{\varphi} = \frac{1}{n} \sum_{i=1}^{n} \varphi_{k,i}$$

  where $\varphi_{k,i}$ is the Euclidean-distance calculation and n is the amount of vectors in the data. If $\overline{\varphi} > \widetilde{\varphi}$, then there is probably an incident.

- Assign two threshold parameters which are $\widetilde{\varphi}$ and $\widetilde{\rho}$. Calculate the amount of outliers:

$$\rho = \sum_{i=1}^{n} h(i)$$

  where the function $h(i)$ is defined as

$$h(i) = \begin{cases} 1, & \text{if } \varphi_{k,i} > \widetilde{\varphi} \\ 0, & \text{if } \varphi_{k,i} \leq \widetilde{\varphi} \end{cases}$$

  and if $\rho > \widetilde{\rho}$, then there is probably an incident.

The first strategy was chosen to allow an easier analysis of the evaluation. The performance metrics used were Detection Rate (DR) and False Alarm Rate (FAR). With an $\widetilde{\varphi}$ = 2.97, the DR = 81.5% and FAR = 1.83%.

Dogru et al. (2012) consider incidents in normal traffic flow as an anomaly. The principle used was that when accident happens, following cars will slow down or stop and many cars are affected from accident

which could be considered an anomaly, depending on the technique and attributes. The Simulation of Urban MObility (SUMO) traffic simulator was used to enable mobility of vehicles and all the information needed. The algorithm used was DBSCAN to cluster the anomalies as they occurred. The evaluation metric used was not specified but the authors shared an example of the output and the number of anomaly cluster increases after the accident.

Ren et al. (2012) used K-means as a feature learning algorithm and a L1 regularized Support Vector Machine (SVM) with RBF kernel for incident classification.

Alonso et al. (2014) used SVM for wet road surface identification using tyre/road noise, detecting road wetness from audio of the tire-surface interaction and discriminating between wet and dry classes.

Chen et al. (2015) used an onboard accelerometer to sense vehicle vibrations by examining the z-axis acceleration. Normally, the vibration on abnormal road sections is greater than that on smooth sections, so an abrupt increase of z-axis acceleration often signifies a pothole. The Gaussian Mixture Model (GMM) algorithm was used for the event detection.

Silva et al. (2017) implemented and deployed a cloud-based road condition/anomaly information management service based on vehicle context data, collected during driving activities using smartphones that collect inertial data, that is, accelerometer data, for x, y, and z axes, GPS coordinates, speed, and bearing during driving activity. The algorithms used were Gaussian Naive Bayes (NB), Linear Support Vector Classification (SVC), Decision Tree, Gradient Boosting and Multi-layer Perceptron (MLP) Classifier. The algorithms were evaluated on using all the attributes and the attributes that resulted from the preprocessing process. Linear SVC had the best improvement going from both experiences. Then in Soares et al. (2018), the authors developed a cloud-based road condition/anomaly information management service based on vehicle context data. Collected data is then processed, transformed and classified using a ML model, producing the road anomaly information managed by the service, described on the previous mentioned article.

Chowdhury et al. (2020) used an autonomous device anomaly detection in a surveillance setting, which contained data from IMU sensor and images of respective time frames. The IMU data was divided on two groups: IMU/data which contains the orientation and velocity information of the drone along 3 axis and IMU/mag that contains magnetic field data read by magnetometer.

The methodology proposed was a autoencoder based anomaly detection system for IMU data and an AngleNet to estimate the angle of an input image with respect to a normal reference image sample and later ensembles the 3 outputs to estimate the degree of abnormality. The AngleNet is a Convolutional Neural Network (CNN) based regression architecture which estimates angular displacement between two images. An ensemble mechanism is utilized to estimate the degree of abnormality using predictions both from the given image and IMU data samples at a particular timestamp.

Even through the authors claim that they used an unsupervised Anomaly Detection approach, in reality their approach was semi-supervised because only the data labeled as normal was used on training the algorithms.

Figure 4 has the information about the performance of the autoenconder on IMU data.

Data	Accuracy	F-1 Score	False Negative
IMU/data	96.8%	0.9812	2
IMU/mag	100%	0.95	0

Table 4: (Chowdhury et al., 2020): Performance on IMU Data.

C. Wu et al. (2020) used Logistic Regression (LR), SVM and Random Forest (RF) for pothole-detection. The SVM classifier had as parameter the radial basis function kernel and RF with n_estimators=100. The metrics used for evaluating the models were Precision, Recall and F1-Score and the results can be observed in Table 5.

Classifiers	**Accuracy for Training Set**	**Accuracy for Testing Set**	**Window Type**	**Precision**	**Recall**	**F1-Score**
LR	0.961	0.952	normal	0.965	0.984	0.974
			pothole	0.851	0.734	0.788
SVM	0.951	0.948	normal	0.952	0.992	0.971
			pothole	0.908	0.642	0.752
RF	0.999	0.957	normal	0.965	0.988	0.976
			pothole	0.885	0.750	0.812

Table 5: Precision, Recall and F1-Score (C. Wu et al., 2020).

Chapter

3 Machine Learning Process

A well defined process for a Machine Learning (ML) project allows for a structured and organized development which contributes for the best results.

Two model processes will be introduced in this chapter in order to define the structure of the proposed work and a better organization of the development process: Cross Industry Standard Process for Data Mining (CRISP-DM) and the Cross Industry Standard Process model for the development of Machine Learning applications with Quality assurance methodology (CRISP-ML(Q)).

3.1 Introduction to the CRISP-DM

The Cross Industry Standard Process for Data Mining (CRISP-DM) is a process model originally developed to standardize the Data Mining process by Shearer (2000), currently widely adopted for ML as well (Lukyanenko et al., 2019).

The CRISP-DM process model has six major phases (Shafique et al., 2014; Wirth, 2000):

- Business Understanding: definition of important factors including success criteria, business and data mining objectives and requirements as well as business terminologies and technical terms.
- Data Understanding: executing data collection, checking of quality and exploring of data to get insight in order to form hypotheses for hidden information.
- Data Preparation: selection and preparation of the data into the state needed for analysis.
- Modeling: process selection and application of various modeling techniques to support business decisions; different parameters are set and different models are built for same data mining problem.
- Evaluation: evaluation of obtained models and interpretation of the results.
- Deployment: determining the use of the knowledge and results obtained and organizing, reporting and presenting the gained knowledge to a customer, for example.

Annex I there is a more detailed structure of the phases involved on the CRISP-DM model process. Taking into consideration, that the CRISP-DM process was designed for Data Mining, the process lacks some important aspects of a ML project.

The first aspect to consider is the fact that Data Mining is different from Machine Learning.

Data mining is the process of discovering interesting patterns and knowledge from large amounts of data. The data sources can include databases, data warehouses, the Web, other information repositories

or data that are streamed into the system dynamically (Han et al., 2012). Machine Learning, on the other hand, is the study of algorithms that allow computer programs to automatically improve through experience (Mitchell, 1997).

Considering that Machine Learning and Data Mining are not the same, the process involved in applying each of this methodologies is also different. This reasons led to researchers Studer et al. (2021) to propose a guide for ML practitioners though the development life cycle.

3.2 The CRISP-ML(Q) methodology

Studer et al. (2021) released a new methodology called Cross Industry Standard Process model for the development of Machine Learning applications with Quality assurance methodology (CRISP-ML(Q)), which is an adapted CRISP-DM applied to the development of ML projects.

The differential aspect of CRISP-ML(Q) and other process model is the focus on quality assurance throughout the development of the project.

The CRISP-ML(Q) model is divided into six stages:

1. Business and Data Understanding
2. Data Preparation
3. Modeling
4. Evaluation
5. Deployment
6. Monitoring and Maintenance

Taking the CRISP-ML(Q) process as a basis for the definition of all the stages, an explanation of its contents is next made.

3.2.1 Business and Data Understanding

The first step in any ML project is to identify the scope of its application, the success criteria, and a data quality verification. The goal of the first phase is to ensure the feasibility of the project.

Regarding the scope of the project, is imperative that the objectives and requirements from a business perspective are collected. The application of ML tasks take this information into consideration, so the continuous communication between business stakeholders is instrumental for a successful understanding and application of the objectives and requirements.

The feasibility confirmation before setting up the ML project is a best practice in industry. The application of the ML Canvas Framework is an example of a structured document to perform this confirmation (Zhou et al., 2020). The ML Canvas guides through the initial phases of the ML application. The non-functional requirements include robustness, scalability, explainability and resource demand that are used for the development and verification of later phases (Hamon et al., 2020; Studer et al., 2021).

The principal non-functional requirements can be explained as being (Fernandes et al., 2016; Ghezzi et al., 1991):

- Robustness is the ability of a computer system to cope with errors during execution and cope with erroneous input.
- Scalability is a property of a system to handle a growing amount of work by adding resources to the system.
- Explainability is the extent to which the internal mechanics of a ML system can be explained in human terms.

Taking into account that the data is the guide along the process, data collection and data quality verification are essential to achieve business goals. A descriptive documentation of the process involved behind the data collection exercises is crucial for evaluating and assurance of the data quality. An well defined data exploration and of its statistical properties, resulted from a Exploratory Data Analysis (EDA) (Tukey, 1977), can assure that the requirements are being followed (Morgenthaler, 2009).

3.2.2 Data Preparation

The second phase of the CRISP-ML(Q) process model aims to prepare data for the modeling phase, considering that in this phase Feature Engineering tasks are analyzed.

While for Studer et al. (2021) the tasks of Feature Selection and Feature Engineering are considered as being independent, the identification of valuable and necessary features for the future model training is a phase of Feature Engineering named Feature Selection.

Feature Engineering can be divided into three phases: Feature Selection, Feature Extraction and Feature Reduction. According to some articles, e.g., (Gherabi et al., 2021; Qin, 2020), Feature Normalization is also described as being a task of Feature Engineering.

Feature Selection is related to Feature Reduction in that both methods seek to reduce the number of features, however use different approaches to do so. Feature Selection uses filter methods, wrapper methods or embedded methods for the selection of features, being examples of techniques for Feature Selection: Correlation, Forward Selection and Information Gain (H. Liu et al., 2012).

Feature Reduction is mostly known as dimensionality reduction. Therefore, the number of features can be reduced in two ways: using Feature Reduction or Feature Selection. Principal Component Analysis (PCA), for instance, is a well known Feature Reduction approach that allows the original variables to be combined into a smaller number of principal components (Hoffmann, 2007).

Feature Selection then can be described as: from m variables one selects a subset of variables that seem to be the most discriminating. The features obtained therefore, correspond to some of the given measurements, by simply selecting and excluding given features without changing them, while in the display methods the dimensionality reduction is obtained by using all the variables but combining these into a lower dimension. Feature selection therefore constitutes a means of choosing sets of optimally discriminating variables, filtering irrelevant or redundant features from the data set and, if these variables

are the results of analytical tests, this consists, in fact, of the selection of an optimal combination of analytical tests or procedures (Deming et al., 1988; Theodoridis et al., 2009).

Feature Extraction allows for the creation of new features based on mathematical computations applied to the data. Some examples applied to data sets are (Nahid et al., 2019):

- Mean: arithmetic mean of the dataset.
- Standard Deviation (σ): measure of the amount of variation or dispersion of a set of values.
- Variance (σ^2): squared deviation of a random variable from its population mean or sample mean.
- Minimum: it is the minimum value in the given data set. The lowest possible value of the function at minimum point in a given data set is called Least significant value.
- Maximum: the highest possible value, also known as the most significant value, in a given data set.
- Kurtosis: defines how heavily the tails of a distribution differ from the tails of a normal distribution.
- Skewness: defined as the measure of the similarity or asymmetric distribution. It can be positive as well as negative.

The standardize data step on the CRISP-ML(Q) model process denotes the process of normalization of the data and the establishing of the file format needed for the specific problem, a more analytical perspective of the standardize data task. Normalization is an important task on the application of some ML algorithms because features defined on different scales could lead to bias towards the larger scaled features (Sola et al., 1997; Trebuna et al., 2014). Normalization implies the rescaling of the features values into a range, commonly on the range [0,1]. On the scientific field, standardization frequently means rescaling of the data to have a mean of 0 and a standard deviation of 1 (Sabri, 2021).

CRISP-ML(Q) process also mentions the problem of unbalanced classes as being an important aspect to tackle by applying over-sampling or under-sampling strategies. Likewise, data augmentation tasks can be important to perform, depending on the ML task objective. Data augmentation utilizes known invariances in the data to perform a label preserving transformation to construct new data (Engelhardt et al., 2021).

3.2.3 Modeling

The choice of modeling techniques depends on the ML and the business objectives, the data and the boundary conditions of the project the ML application is contributing to. The requirements and constraints that have been defined in the Subsection 3.2.1 are used as inputs to guide the model selection to a subset of appropriate models. The goal of the modeling phase is to craft one or multiple models that satisfy the given constraints and requirements (Studer et al., 2021).

The literature research on similar problems is also an important source to select a subset of models and tasks to establish a baseline for the modeling strategy.

The definition of quality measures of the model can be weighed differently depending on the application. On the CRISP-ML(Q) development lifecycle model, besides the performance metric of the model, measures such as robustness, explainability, scalability, resource demand and model complexity should

also be evaluated.

Generally, the modeling phase includes model selection, model specialization and model training tasks. Additionally, depending on the application, there might be a need to use a pre-trained model, compress the model or apply ensemble learning methods to get to the final ML model.

Another important principle of scientific methods and the characteristics of robust ML application is reproducibility: the method itself needs to be reproducible and also its results. Keeping track of the changes to the model and its implications on the performance through experimental documentation is crucial to prove reproducibility. The documentation should contain the listed properties in the method reproducibility task.

3.2.4 Evaluation

Model training is followed by a model evaluation phase. During this phase, the trained model is validated against a test set and model robustness is assessed.

The explainability of the ML model is important to provide trust, meet regulatory requirements and comprehend the ML-assisted decisions.

Finally, the model deployment decision should be met automatically based on success criteria or manually by domain and ML specialists. Similar to the modeling phase, all outcomes of the evaluation phase need to be documented (Visengeriyeva et al., 2021).

3.2.5 Deployment

The ML model deployment denotes a process of the ML model integration into the existing software system. After succeeding in the evaluation step in the ML development life cycle, the ML model is graduated to be deployed in a production environment (Visengeriyeva et al., 2021).

The ML model deployment includes the following tasks: inference hardware definition, model evaluation under production environment, providing user acceptance and usability testing, providing a fall-back plan for model outages and setting up the deployment strategy.

3.2.6 Monitoring and Maintenance

Once the ML model has been put into production, it is essential to monitor its performance and maintain it.

When an ML model performs on real-world data, the main risk is that the characteristics of the data distribution are represented incorrectly by the training data. The unseen data can degrade the performance of the model over time. Furthermore, model performance is affected by the degradation of hardware and software and/or hardware updates. Therefore, the best practice is to perform monitoring tasks in order to best analyze the strategy for fixing the model performance drop when it occurs, allowing for an update on

the ML model or even an adjust of the ML process.

3.3 Conclusion

In this chapter, a description of the CRISP-DM and CRISP-ML(Q) model processes was presented in order to better define the next phases.

The importance of a well defined process for a ML project is crucial for a continuous development where all stakeholders are in sync with the state of the project. The pursuit for a ML model process has been more prioritized in the last few years, inasmuch as the market demand for Artificial Intelligence (AI) related products is at an all time high.

Considering that the CRISP-DM was developed with an objective of being applied to a Data Mining industrial project, the application of this model to a ML project would lead to weaknesses in the process, where ML specific tasks would not be considered.

The CRISP-ML(Q) is, as of today, the most complete and reliable process model for the development of ML project in a generalized industrial scenario. Nevertheless, it still lacks a more scientific background to the ML process, with it being, in some stages, too industrial and business focused.

It is important to emphasize that the ML process is not a linear procedure. Contrarily, it is a cycle where the tasks can go back and forth among them. This contributes to an enrichment of all phases as more knowledge is acquired during the development.

Chapter

4 Use Case: EasyRide

The project Detection of anomalies from fusion of multiple sensors is one of the 18 Projects of the Easy Ride R&D Program. This program results from the University-Industry Collaborative Partnership engaged between Bosch Car Multimedia Portugal, S.A. and University of Minho, with the collaboration of Centro de Computação Gráfica.

The objective of this project is to detect anomalies that happens inside the vehicle, with the use of multiple sensors fused together for better detection and results understanding.

This chapter presents the first phase of the Cross Industry Standard Process model for the development of Machine Learning applications with Quality assurance methodology (CRISP-ML(Q)) model process: Business and Data Understanding. The importance of this chapter lies on the implementation of the data collection task.

4.1 Business Understanding and Data Understanding

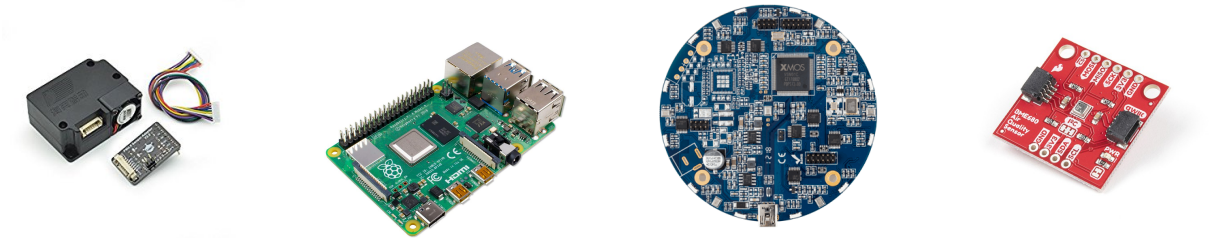
The project Detection of anomalies from fusion of multiple sensors is part of the defined vision for the Intelligent Cockpit subprogram and aims to develop solutions to enable anomalies detection based on the fusion of data from several sensors that monitor the vehicle interior (In-Vehicle Sensing (IVS)).

A major advantage of sensory fusion systems is the capability of crossing information generated by several sensors. Data coming from several sensors describes normal processes or behaviors occurring inside the vehicle. Anomaly Detection aims to understand when these systems behave in abnormal states by considering single or combined sources of sensory data.

An overview of the sensors used is described in the next section.

4.1.1 Data Collection

Considering that no data was available for this project, data acquisition exercises where pivotal and necessary. The data acquisition of the Easy Ride data was performed using the components presented in Figure 6.



(a) PM2.5 (b) Raspberry 4 (c) UMA-8 (d) BME680

Figure 6: Components of the data acquisition setup

The required components are a particle sensor and microphone, connected though a Raspberry PI 4. The addition of a gas sensor to the array of sensors was thought to complement the particle sensor in its recordings outputs.

The Mic Array UMA-8-SP (Figure 6c) is a high-performance multichannel USB microphone paired with a digital audio amplifier. This sensor has seven high-performance MEMS microphone array configured in a circular disposition to ensure the capture of high-quality voice for heterogeneous kinds of applications. The UMA-8-SP (miniDSP, 2018) supports several voice algorithms including beamforming, noise reduction, acoustic echo cancellation and de-reverb, which are important algorithms to filter irrelevant and noisy data (Table 6).

Specification	Description
USB audio capabilities	2 possible configurations for audio recording: • 8-channel mode (7 x MEMS installed + 1 x spare) • Stereo recording with DSP processing enabled USB audio playback: 2ch fed to stereo amplifier
DSP processing	• Beamforming with configurable beam width (up to 20dB attenuation) • Perceptual acoustic echo cancellation (up to 80dB attenuation) • Noise suppression (up to 20dB attenuation) • De-reverb (up to 20dB attenuation)
Sample rate	11/16/32/44.1/48 kHz
Resolution	24bit
Amplifier output	Two output power amplifier >90% efficiency at full power
MEMS microphones	7 x microphones with low noise buffer high performance modulator • Low distortion: 1.6% @ 120 dB SPL • High Signal-to-noise ratio: 65 dB and flat frequency response • Radio Frequency shielded against mobile interference • Omnidirectional pick-up pattern

Table 6: UMA-8 specifications

The PM2.5 laser dust sensor SKU SEN0177 (Figure 6a) represents a digital universal particle concentration sensor that can be used to analyze the concentration of particulate matter, that is, the amount of suspended particulate matter (mixture of liquid droplets and solid particles) in a unit volume of air that has 0.3 to 10 microns and the quality data of per particle (Beijing Hike IoT, 2017). The specifications from the

PM2.5 sensor is described on the Table 7.

Specification	Description
Measuring pm diameter	0.3-1.0, 1.0-2.5, 2.5-10 (um)
Measuring pm range	0 ~500 ug/m3
Standby current	≤ 200 uA
Response time	≤ 10 s
Operating temperature range	-20 ~50C
Operating humidity range	0 ~99% RH
Minimum size of micron resolution	0.3

Table 7: PM2.5 specifications

The SparkFun Environmental Sensor - BME680 (Qwiic) (Figure 6d) is a module for the BME680 gas sensor from Bosch (2017). The BME680 combines a gas sensor with temperature, humidity and barometric pressure sensing (Table 8).

Specification	Description
RH	• Operating range: 0% to 100% • Absolute Accuracy: ±3%RH • Resolution: ±0.008%RH
Temperature	• Operating range: -40°C to +85°C • Absolute Accuracy: ±0.5°C to ±1.0°C • Resolution: 0.01°C
Pressure	• Operating range: 300hPa - 1100hPa • Relative accuracy: ±12Pa (25°C to 40°C @ constant RH) • Absolute accuracy: ±60Pa (0°C to 65°C) • Resolution: 0.18PA, highest oversampling
Gas	• Resolution of gas sensor resistance: 0.05% to 0.11% • Typical current consumption (varies based on mode and active sensor) ◦ 2.1µA to 18mA ◦ 0.15µA (sleep mode)

Table 8: BME680 specifications

4.1.1.1 Planning of the Data Collection

The Data Acquisition exercises were based on the requirements for the project objective, it being audio anomaly detection and the measurement of air quality inside the vehicle.

The regular use of a vehicle comprise background noise done by the proper use of the vehicle (e.g. engine sound) and all the surroundings of the moving or stationary vehicle. Additionally the noise made by the occupants of the vehicle is a factor that also needs to be evaluated and analyzed. Considering these aspects, a wide selection of use cases were defined and their corresponding labels of normal and abnormal behavior.

Abnormal behavior does not strictly imply abnormal use of vehicle or of its occupants. It just implies

some aspects of not normally occurring events.

Use Case ID	Use Case Description	Behaviour Label
0101	Normal Wake state	Normal
0102	Talking	Normal
0103	Texting and talking	Normal
0104	Using the Smartphone	Normal
0105	Reading	Normal
0106	Singing	Normal
0204	Coughing	Anomaly
0205	Breaking a Window	Anomaly
0206	Argumenting	Anomaly

Table 9: Use Case Description of Audio Anomaly Detection

Considering logistical and organizational barriers to perform all the use cases presented on the Table 9, the use cases collected were the following:

- Normal Wake State: only the sound of the functioning of the vehicle and exterior sounds is considered for this use case; the objective is to then be able to filter the sounds of a normal running vehicle and its surroundings.
- Talking: occupants talking inside the vehicle.
- Texting and Talking: occupants talking on the phone or via Bluetooth; the variant of texting, because of security constraints, was not included.
- Singing: occupants singing inside the vehicle.
- Coughing: occupants coughing inside the vehicle.
- Argumenting: occupants arguing inside the vehicle.

Regarding the audio collection requirements, each use case had to be collected from 3 to 5 hours following a certain distribution of variants. These variants were the combination of windows closed/open and radio off/on. The most significant variant was windows closed and radio off, which had to be 70% of the total time, where the remaining 30% split between the other variants (10% to each).

The technicalities for this collection were that the microphone had to be recording at least at 44100 Hz with a single channel and, if the recordings were being done in segments, each segment had to be more than one minute. The importance of these specifications lies in the fact that these were the basis for a audio analysis that was able to differentiate between the different use cases. The requirements for the air quality inside vehicle only defined that a total of 6 hours should be collected, 3 hours for each type of use case (normal and abnormal), independently of the use case.

Sensor	Sample Rate
UMA-8	48kHz
BME680	0.5Hz
PM2.5	0.5Hz

Table 10: Sensors Sample Rate

The final setup configurations are presented in Table 10. On can conclude that while the microphone recorders, every second, 48.000 instances, the gas and particle sensors only record every two seconds, which means that for each 96.000 instances of the microphone there is 1 recording of gas and of particle, like specified on Table 10.

In order to simplify and streamline the data collection process, a representative nomenclature of window and radio variants was created and put into practice during the planning, execution and labeling of the exercises.

Structure	Variant
W0	Windows closed
W1	Windows open
R0	Radio off
R1	Radio on

Table 11: Nomenclature of window and radio variants

An exercise of Normal Wake State with windows open and radio on would be portrayed as 0101W1R1: 0101 corresponding to the use case ID of the event (0101) and the respective nomenclature of the state of the windows (W1) and radio (R1). This format was applied to all use cases and variants of windows and radio.

The recording of data for the Air Quality Analysis was performed during the recording of the audio information, where the analysis of anomalous or normal events was not considered. An example of anomalous event was smoking, a particularity that no person involved in this first data collection does.

4.1.1.2 Execution of the Data Collection

The execution of the data acquisition plan was done considering all the use cases and variants already described above. A stationary and moving data acquisition were conducted using a series of different vehicles ranging from internal combustion engines (petrol, diesel, Liquefied Petroleum Gas (LPG)) to a Battery Electric Vehicle (BEV) (Table 12).

On the first data acquisition, all vehicles were stationary and it allowed for some conclusions to be made and configurations to be changed. The microphone was recording dynamically on two channels of all 8 channels, meaning that there was no control over which channels were recording and even in which side it was being activated. If a noise was made on the right side, for instance, the microphone would dynamically choose the two nearest channels to activate and record. The impossibility to know where the events occurred lead to a change on the configuration of the microphone. What started out as 8 channels

recording dynamically, moved to only 2 fixed channels (mic0 and mic1), with mic0, corresponding to the channel 1, recorded the driver side and mic1 recorded the opposite direction.

Both data collections were done at the same time by having the particle and gas sensors recording simultaneously as the microphone, which allowed to collect more data for the air quality exercise. The exercises were carried out sequential, with the sensors data being saved every 80 seconds in order to preserve the data if any error on the data collection occurred.

The moving data acquisition plan was conducted on a wider spread of vehicles for a better analysis and conclusions.

Brand	Model	Fuel	Exercise Category
Mini	One	Gasoline	Stationary
BMW	520D	Diesel	Stationary/Moving
BMW	i3	BEV	Stationary/Moving
Nissan	Micra K12	Gasoline/LPG	Moving
BMW	120D	Diesel	Moving
Peugeot	207	Diesel	Moving
Ford	Fiesta	Gasoline	Moving

Table 12: Specifications of the vehicles used on the Data Collection exercises

A wide number of specifications for each data collection event were collected for a better understanding of the data.

- Experiment ID: identifier of the experiment in each given day.
- Event ID: identifier of the use case (e.g. 0101).
- Event Description: name of the use case (e.g. Normal Wake State).
- Run: identifier of the exercise.
- Car: brand and model of the vehicle used for the data acquisition exercise.
- Fuel: vehicle type fuel.
- Type of road: type of road of the data acquisition exercise (e.g. asphalt).
- Windows: state of the windows (open or closed).
- Windows Description: description of the state of windows (e.g. front windows open).
- Radio: state of the radio (on or off).
- Radio description: description of the state of the radio (e.g. radio intensity: low).
- AC: state of the air conditioning (on or off).
- AC intensity: description of the state of the AC (e.g. AC intensity: low).
- Wipers: state of the wipers (on or off).
- Wipers Intensity: description of the state of the wipers (e.g. wipers intensity: low).
- Passengers: number of passengers on the vehicle.
- Passenger action description: description of the passengers actions and/or location (e.g. driver

and front passenger. Front passenger coughing.).

- Average Speed: average speed during the exercise.
- Location: general location where the exercise was performed.
- Obs: observations and/or new information that is not described on the other specifications.
- Done: if the exercise was completed (1) or it was just planned (0).

Figure 7: Setup placement inside the vehicle for the Data Acquisition of the EasyRide data

The setup was located on the windshield like shows Figure 7, with the microphone being placed on the middle of the windshield and the gas and particle sensor placed on the left side, leaving a gap between the both of them so that the PM2.5 fan cannot be heard.

Table 13 summarizes the distribution of each data acquisition (stationary and moving), where the total denotes the number of experiments gathered of each use case. Table 14 reports more specifically the stationary events collected and the moving events are described in Table 15.

Event ID	**Event Description**	**Stationary**				**Moving**				**Total**
		W0R0	W0R1	W1R0	W1R1	W0R0	W0R1	W1R0	W1R1	
0101	Normal Wake State	23	25	28	23	113	13	15	13	253
0102	Talking Inside the Vehicle	18	31	19	19	98	29	28	42	284
0103	Talking or Texting	25	3	10	-	119	-	16	-	173
0106	Singing	-	2	-	2	94	12	24	12	146
0204	Coughing	29	14	10	11	96	13	14	29	216
0206	Argumenting	-	-	-	-	94	18	17	9	138

Table 13: Distribution of each use case

Event ID	Event Description	Total events	Total events in hours
0101	Normal Wake State	99	2 h 12 min
0102	Talking Inside the Vehicle	87	1 h 56 min
0103	Talking or Texting	38	50 min 40 sec
0106	Singing	4	5 min 20 sec
0204	Coughing	64	1 h 25 min 20 sec
0206	Argument	0	0 h

Table 14: Distribution of the events when the car is stopped.

Event ID	Event Description	Total events	Total events in hours
0101	Normal Wake State	154	3 h 25 min 20 sec
0102	Talking Inside the Vehicle	197	4 h 22 min 40 sec
0103	Talking or Texting	135	3h
0106	Singing	142	3 h 9 min 20 sec
0204	Coughing	152	3 h 22 min 40 sec
0206	Argument	138	3 h 4 min

Table 15: Distribution of the events when the car is moving.

4.1.1.3 Data Labeling

The labeling of the data was performed considering the files resulted by the use of the setup for data acquisition: microphone, gas and particle sensors data files and the metadata file.

The data collected was organized as follows:

- Each day of data acquisition resulted in a folder with the day as a name
- Each day exercises, identified as run, were carried out resulted in several experiment files that are equivalent as one minute and twenty seconds of recordings

The metadata of each experiment is used for the identification of the variants of windows and radio and the use case, which was important for the labeling process. The structure of the labels were as follows: [identification of the exercise and experiment; use case ID and windows and radio variant; initial second; final second]. For instance an experiment of Normal Wake State with window open and radio off could be

labeled as [21_02_2021\run37\5; 0101W1R0; 0; 80].

Algorithm 1: Labeling strategy for the EasyRide data

```
function Labeling (path);
  Input: path of the data folder
for exercise in path do
    for experiment in exercise do
        Find microphone, PM2.5 and BME680 files of experiment;
        Find metadata of experiment;
        Save label structure into text file in folder;
    endfor
endfor
```

Algorithm 1 gives a simple representation of how the labeling script was created, allowing also for a understanding of how the files were organized inside the data folders, representative of each date of data collection.

4.1.2 Exploratory Data Analysis

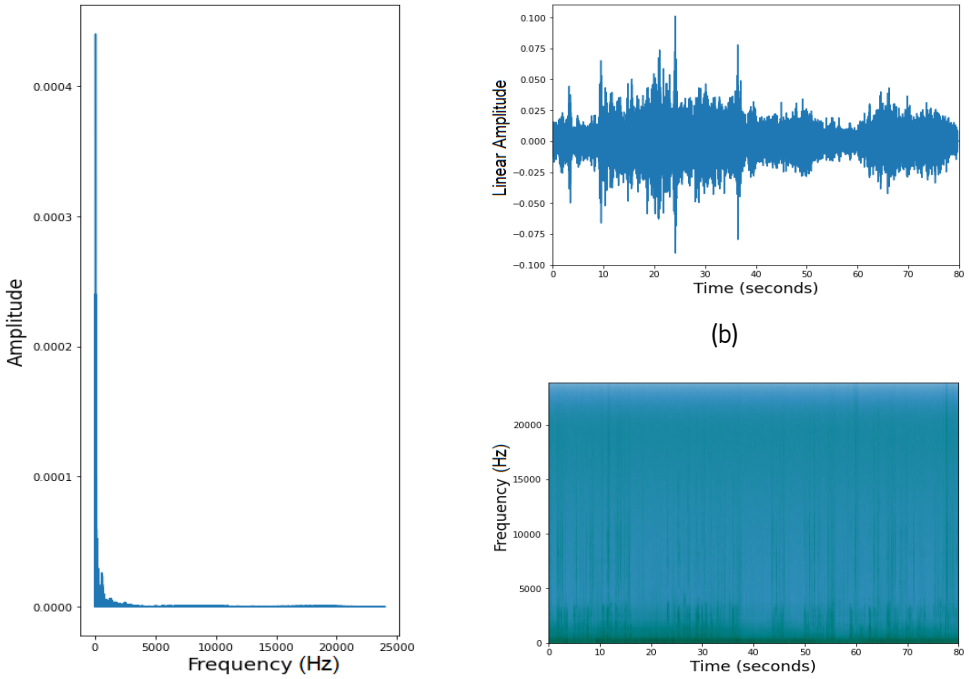
During the data acquisition exercises, an Exploratory Data Analysis (EDA) was performed in order to apprehend the state of the data and to perform changes in the setup for a better quality of the collected data.

The exercises where performed on different environments in order to understand the impact of different scenarios (heavy traffic, low traffic, for instance) and different vehicles in the data recorded.

The analysis is divided into two sections: Audio Anomaly Detection and Air Quality Analysis, the two use cases under study on this project.

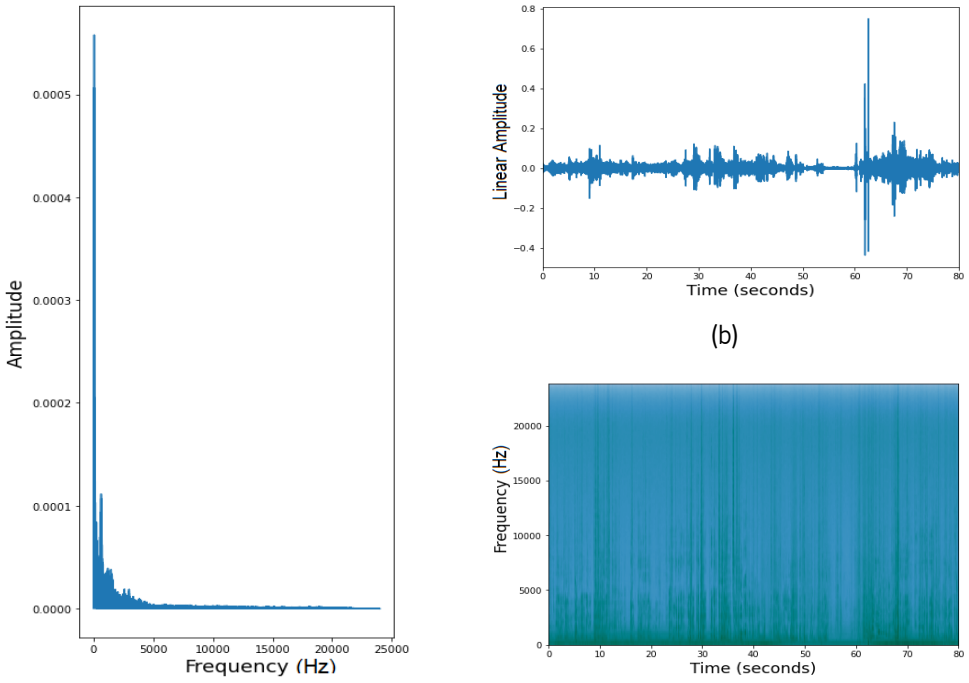
4.1.2.1 Audio Anomaly Detection

The EDA of the audio takes into consideration the definition of anomalous events in this setting: argumenting and coughing. By assessing the anomalous events, one can conclude that the most similar normal events are, respectively, talking and normal wake state. A comparative analysis of the amplitude and frequency is carried on considering these conclusions.



(a) (b) (c)

Figure 8: (a) Fast Fourier Transform, (b) signal representation and (c) spectrogram of a talking recording

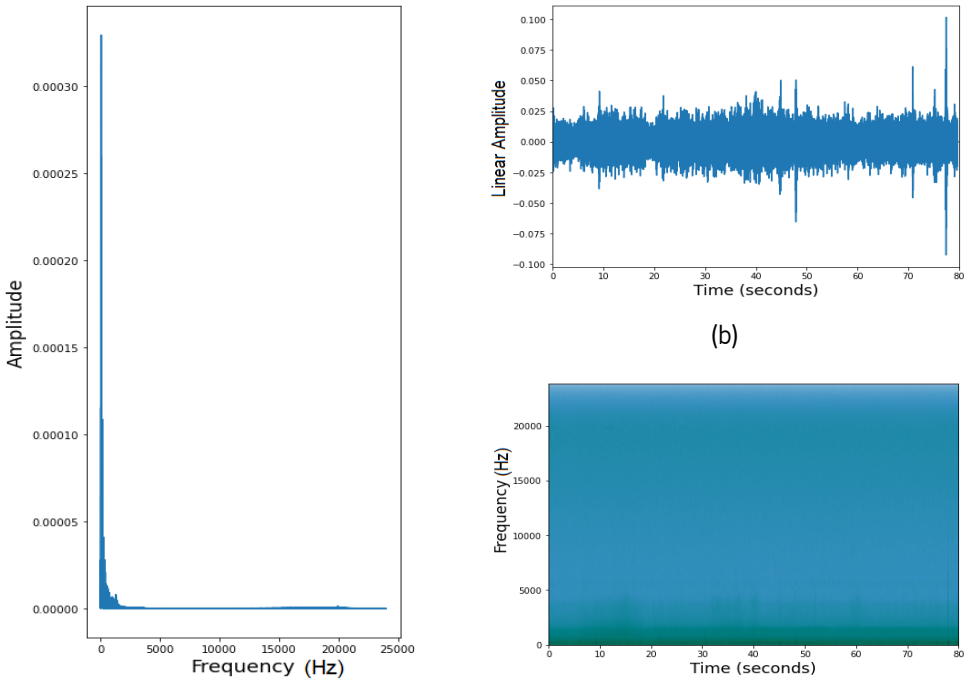


(a) (b) (c)

Figure 9: (a) Fast Fourier Transform, (b) signal representation and (c) spectrogram of an argument recording

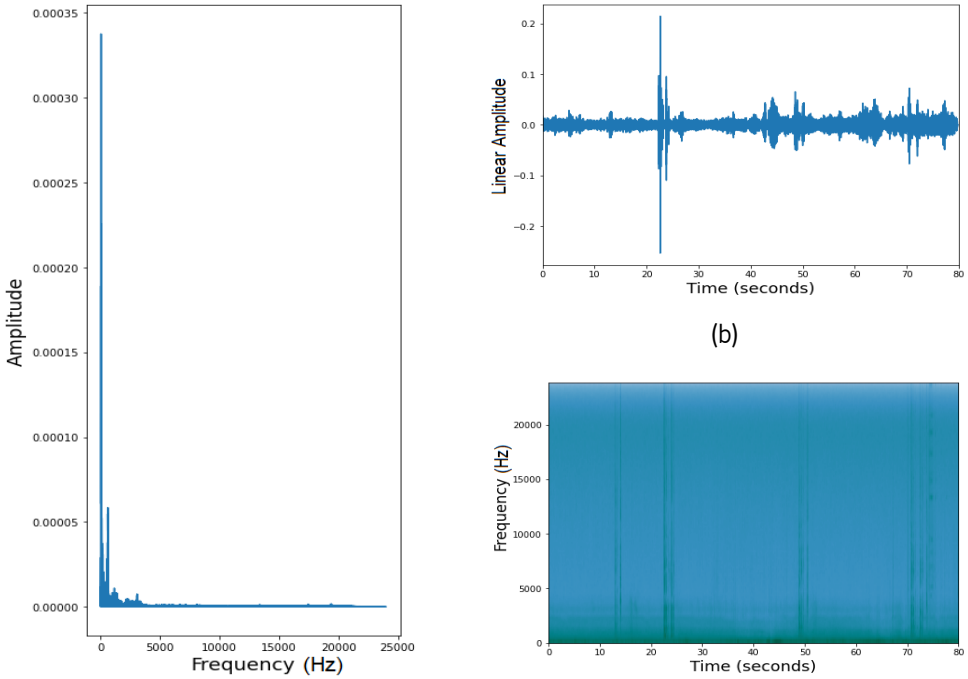
Figure 8b, representative of a talking recording, is possible to conclude that lower amplitudes are stimulated, when compared to the argument recording (Figure 9b). The peak reaches a value of 0.8 on the argument, while the maximum amplitude value on the talking recording is 0.1.

Besides the difference in amplitudes, the frequency aspect of the signal is also different between the two events. Comparing the Fast Fourier Transform (FFT) (Figures 8a and 9a) and the spectrogram (Figures 8c and 9c) of both events, it is possible to conclude that while the frequencies between 0Hz and 4000Hz are present on either events, the amplitude is higher on the argument recording and there is a presence on the whole spectrum, where it is possible to visualize the presence on higher frequencies, between 15000 and 20000Hz.



(a) (b) (c)

Figure 10: (a) Fast Fourier Transform, (b) signal representation and (c) spectrogram of a normal wake state recording



(a) (b) (c)

Figure 11: (a) Fast Fourier Transform, (b) signal representation and (c) spectrogram of a coughing recording

On the Normal Wake State recording (Figures 10a and 10c), only the frequencies until 2500Hz are stimulated. When comparing to the FFT graph of the coughing event (Figures 11a and 11c), the frequencies on this range have higher amplitudes and also frequencies until the 20000Hz are possible to encounter when an cough happens, for example between the 20 and 30 seconds of the recording.

Regarding the amplitude (Figures 10b and 11b), on the coughing recording the amplitude of the signal reaches higher values, having an absolute positive and negative peak surpassing 0.2.

4.1.2.2 Air Quality Analysis

The analysis of air quality is an important characteristic when considering the comfort of the occupants and its well being. A bad air quality inside a vehicle could lead to health risks for its occupants.

Due to organizational constraints, the recording of anomalous events for the air quality of the vehicle interior was not considered. The analysis performed is related to a normal flow on a route compared to a high traffic circumstance. Only the data retrieved from the particle sensor was studied on account of the sufficient information that it acquires.

Column Name	Description
pm10 standard	Concentration of PM1.0, ug/m3
pm25 standard	Concentration of PM2.5, ug/m3
pm100 standard	Concentration of PM10.0, ug/m3
pm10 env	Internal test data
pm25 env	Internal test data
pm100 env	Internal test data
particles 03um	The number of particulate of diameter above 0.3um in 0.1 liters of air
particles 05um	The number of particulate of diameter above 0.5um in 0.1 liters of air
particles 10um	The number of particulate of diameter above 1.0um in 0.1 liters of air
particles 25um	The number of particulate of diameter above 2.5um in 0.1 liters of air
particles 50um	The number of particulate of diameter above 5.0um in 0.1 liters of air
particles 100um	The number of particulate of diameter above 10.0um in 0.1 liters of air

Table 16: PM2.5 data description

The variety of data collected through the communication protocol of the PM2.5 sensor to the system is described in Table 16, where a description of each data title of the data collected is presented.

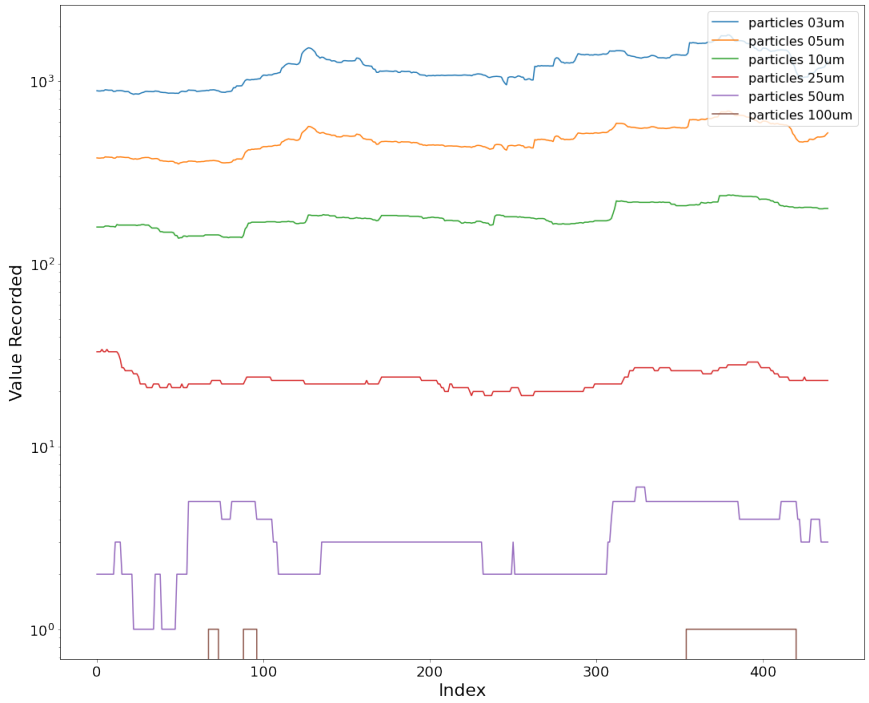


Figure 12: Normal recording captured by the particle sensor

Figure 12 is a representation of a normal recording with the particle sensor, where the values in all variables are stabilized and with no peak, considering that the values are almost linear. Different vehicles can have different linearities of the values considering, for example, the air isolation that it offers from outside influences.

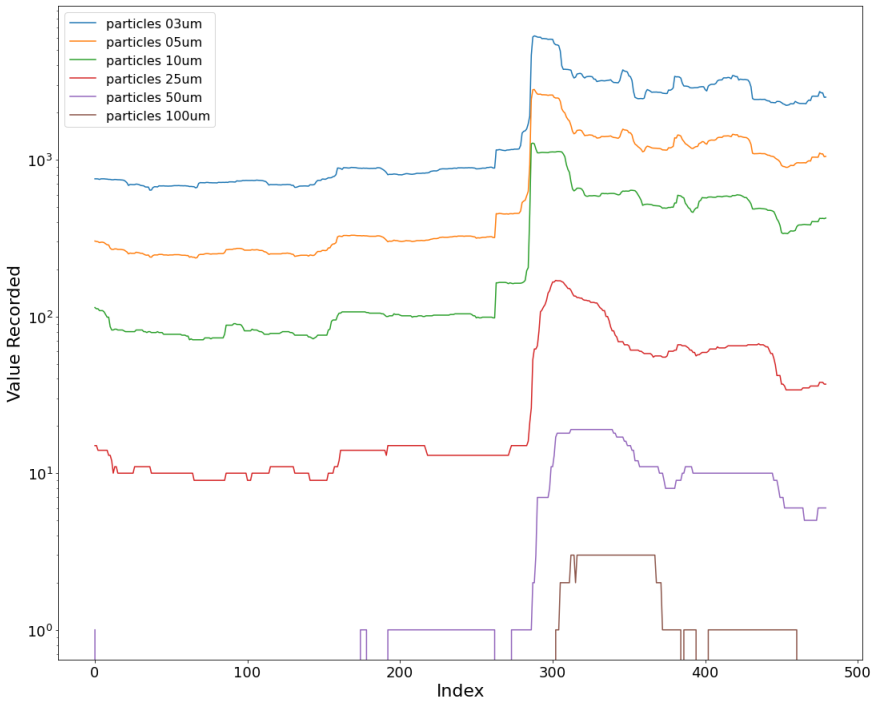


Figure 13: Heavy traffic recording captured by the PM2.5 sensor

In Figure 13, the values are similar to those on the normal example (Figure 12) until a sudden peak happens. Figure 13 represents a recording of a two-way road where both sides, that were stopped on the traffic light, started to move again, requiring a low gear and resulting in a higher quantity of gases expelled by the exhaust of the vehicle and those around it. The sensor was able to capture the abrupt augment of particles expelled resulting in a recording higher than those recorded before and on the normal example. It is also possible to conclude that the signal is gaining a negative slope after the peak and, if the recording would have continued, the values would possibly stabilize around normal values.

On both recordings, the variants of windows closed and radio off were identified, which allowed for this analysis to be made.

4.2 Conclusion and Future Work

The EasyRide project was crucial because it allowed for a practical implementation of the first stage of the Machine Learning (ML) process.

From the preliminar analysis of the data it was possible to change the initial configuration of the microphone, dismantle the setup because of the noise coming from the P2.5 fan being audible from the microphone, resulting in noise data, and also to study if the setup was obtaining the data with the preferred structure.

The main focus of the EDA process was the perception of the quality of the setup, i.e., whether the setup configuration allowed relevant data to be obtained given the project objective and requirements.

Considering these factors, the sensor is capable to capture events of heavy traffic that resulted in heavy expel of exhaust gases, an important feature for air quality analysis that is important to filter and

require more study.

The values of particles recorded increased meant that the recordings of the exterior environment where possible to evaluate based on the particles that traveled from the air duct that connects the exterior air environment to the interior.

Performing exercises in which one of the occupants of the vehicle is smoking would allow for a capture of an irreprehensible anomaly considering the high health risks for all the occupants, even more so if one of them is a child. If the occupant that is smoking the cigarette is the driver, there is a dangerous risk of accident because of visual, manual and cognitive distractions, a more instantaneous repercussion of this activity.

The next phases of the ML process were not developed because the main focus of this thesis is centered around the project described on the Chapters 5 and 6.

Nevertheless, the next phase would consist of applying data preparation tasks such as, for instance, establishing a group of features that best describe the anomalous events and perform more data collection exercises to augment the available data for the modeling phase.

Chapter

5 Use Case: SlimScaley - Business and Data Understanding

Exterior vehicle damage detection is the detection of damaging events that negatively affect the exterior appearance of the vehicle. These damage inflicting events can be caused by a person, structure, or object (human or non-human made) and can occur while the vehicle is moving or stationary. This system allows to alert the owner of the vehicle if any irregular event has been detected on the vehicle while the engine is running, moving or parked.

Considering that high energy damages involve a release of a lot of energy and, consequently, is easier to distinguish and identify when comparing with normal/background data, the real problem relies when a small damage happens. A small damage produces a response that can be similar to a normal driving event. Therefore, a Small Damage Detection (SDD) system is the scope of the project.

The process involved on the first phase of the implementation of an Anomaly Detection approach to the problematic is related to the Business Understanding and Data Understanding phase.

5.1 Business Understanding and Data Understanding

SlimScaley is a Bosch internal product where the primary goal of the SDD algorithm is to detect exterior impacts into a vehicle via a sensor set placed inside.

The set of sensors used to source information from the vehicle and its surroundings were an accelerometer, gyroscope and a microphone. The selection of sensors was based on the fact that most events can be perceived or detected by the forces involved as a result of the impact and by the resulting sound. The Inertial Measurement Unit (IMU) or combination of accelerometer and gyroscope, record information that allows to perceive these forces, and microphones allow to capture sounds that result from these events. Given that the accelerometer and gyroscope is triaxial, that is, the records are collected of the three axis (x, y and z), is possible to gather information from all the different angles on the vehicle.

An accelerometer measures acceleration forces that can be a dynamic force caused by movement, vibration or the static force like the constant force of gravity (Amin et al., 2016).

Given the context, the objective of applying an Anomaly Detection approach is to detect events that are considered anomalous, in this case the impacts/damaging events.

5.1.1 Data Collection

The setup is composed of multiple sensors and processors, in order to create a system capable of detecting a damage event, all mounted in a Printed Circuit Board (PCB).

Multiple setups were implemented with accelerometer and gyroscope, ranging from accelerometer and gyroscope mounted separately or using IMU to combine both in just one sensor. Regarding the microphone, two units are mounted in opposite corners of the PCB.

The accelerometer and gyroscope specifications are described below on the Table 17.

	Accelerometer			Gyroscope		
Sensor	**Output Data Rate**	**Bandwidth**	**Range**	**Output Data Rate**	**Bandwidth**	**Range**
BMI270	1600Hz	434Hz	± 8G	1600Hz	134Hz	±250deg/s
MPU9250	1000Hz	218.1Hz	± 8G	1000Hz	184 Hz	±250deg/s
SMA130	1000Hz	500Hz	± 8G	-	-	-
BMG250	-	-	-	1600Hz	523.9Hz	±250deg/s

Table 17: Sensor setup for exterior vehicle impact detection use case

The IMUs BMI270 and MPU9250 were used as accelerometer and gyroscope devices since they comprise both. The SMA130 and BMG250 were used together since they are an accelerometer and a gyroscope, respectively. Only one microphone model was used (see Table 18).

Sensor	Sample Rate	Bandwidth
SPG08P4HM4H-1	44100 Hz	10000 Hz

Table 18: Microphone setup for exterior vehicle impact detection use case

The sensors where placed on the windshield, right next to the rear-view mirror. After having a setup ready to collect data, a plan was created to outline events that needed to be collected to later analyze.

Two main groups were identified: damaging and non-damaging events. The plan was conducted so that the experiments could replicate as closely as possible real-life situations. Damaging events involved a careful setup for experiments since human safety was a priority.

The tables presented on the Annex II (Table 30, Table 31 and Table 32) represent the events to be collected and its identifiers.

The data collected was labeled during the execution of the exercises. For each event, a hot key was associated to each event to simplify and streamline the process. When an event starts, the corresponding key is pressed until the end of the said event.

The data resulting from the data collection exercises is aggregated between two files: a JSON and a H5 file.

The JSON files has all the metadata that allows for a better understanding of the characteristics of the collection, as well as the event label. A list of some of the key value attributes used is presented bellow:

- Car identification

- Event label
- Event start time
- Event end time
- Damage type
- Damage severity
- Road Type
- Weather type

The H5 file had all the recordings for the setup sensors. The name of the JSON and H5 file are the same, for an easier correspondence between the data recorded and the associated metadata.

5.1.1.1 Data Preprocessing

Data collection exercises were done using various setups, with different configurations, as explained above. In order to better analyze and take conclusions based on the data collected there is a need for standardize the data for all the setups and join the information of each pair JSON-H5 files.

Accelerometer and Gyroscope

The setup where all the sensors are aggregated is placed on the windshield of each of vehicle used on the data collection exercises. Since the sensors are placed on the windshield and cars have different windshield angles and heights, the rotation matrix of the accelerometer and gyroscope sensors need to be standardized in order to be consistent throughout all data available. A change in the height placement of the setup would mean, for instance, that the affect on the Z axis would be different, considering the calculus of the force of gravity. The accelerometer and gyroscope are deeply dependent on the position that the reading occurs. Given this problem, the data from different cars would not be interpretable in the same way, so everything is converted considering the matrix presented in the below image (Figure 14), taking into account the car coordinates of setup on the windshield.

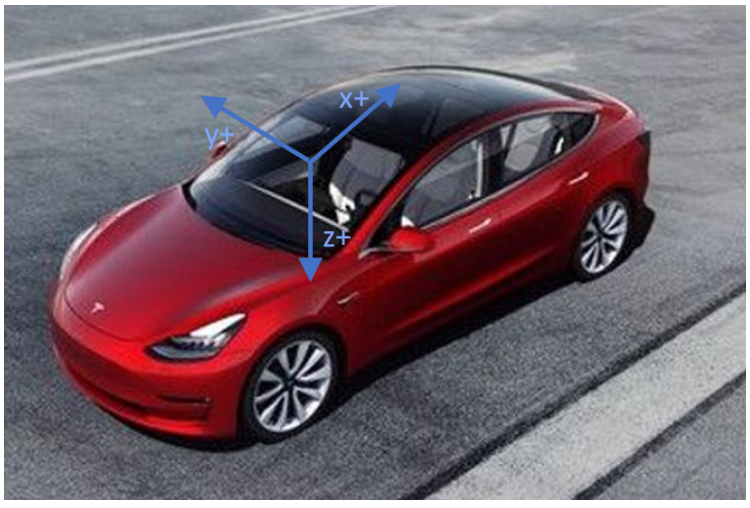


Figure 14: Accelerometer and gyroscope orientation in the rotation matrix.

Considering the orientation of the Z axis, a stationary vehicle, with no other forces but the gravitational force, will have a 1g reading on the accelerometer, caused by the mechanical force exerted in the upward direction by the ground.

After the application of a rotation matrix, depending on the sensor, different transformations were applied to uniformize the sampling rate and the bandwidth. Contemplating the data on Table 17, the focus was to resample the accelerometer and gyroscope sensor data to 1600 Hz and low-pass filter accelerometer data at 218 Hz and gyroscope data at 124 Hz.

Microphone

The same microphone, but with different configurations, was tested until a reliable architecture was establish.

In order to not loose all the data collected prior to that configuration, different transformations were applied, considering the configuration changes to the main one. Ultimately, the audio data was resampled to 24 kHz and low pass filtered to 11 kHz.

Subsequently, after all data was uniform on the different sensors, the information from the H5, where the sensor data was located, and all the metadata and labels derived from the corresponding JSON file was aggregated on a single H5 file.

5.1.2 Exploratory Data Analysis

The first data analysis is important in order to understand the data distribution and draw some conclusions about how to proceed in the next phases. Given that the problem lies on the problematic of anomaly detection, a statistical analysis on the data allows for a better understanding of the available data. Afterwards an in depth analysis of the events is performed in order to perceive details and knowledge of the damage and no damaging events.

5.1.2.1 Statistical analysis

From approximately 39h44 minutes of data recordings, which includes 13186 events manually reported, 7h52 minutes of those are recordings where damage occurred, translating to 2695 events of damage. Based on this analysis alone, is possible to conclude that the data is really unbalanced which is a problematic that needs to be considered on the next phases. An unbalanced dataset on an algorithm that does not complement this fact can lead to significant bias to the higher presence class. Also, as explained before, the use of metrics to evaluate the model that does not consider this aspect can lead to wrongful conclusions. Considering the aspect of anomaly detection, an unbalanced dataset is an expected condition of the data available in order to apply this methodology.

Figure 15 gives a distribution of all impact events, allowing for a better understanding of the quantity of data collected in each of the events categories.

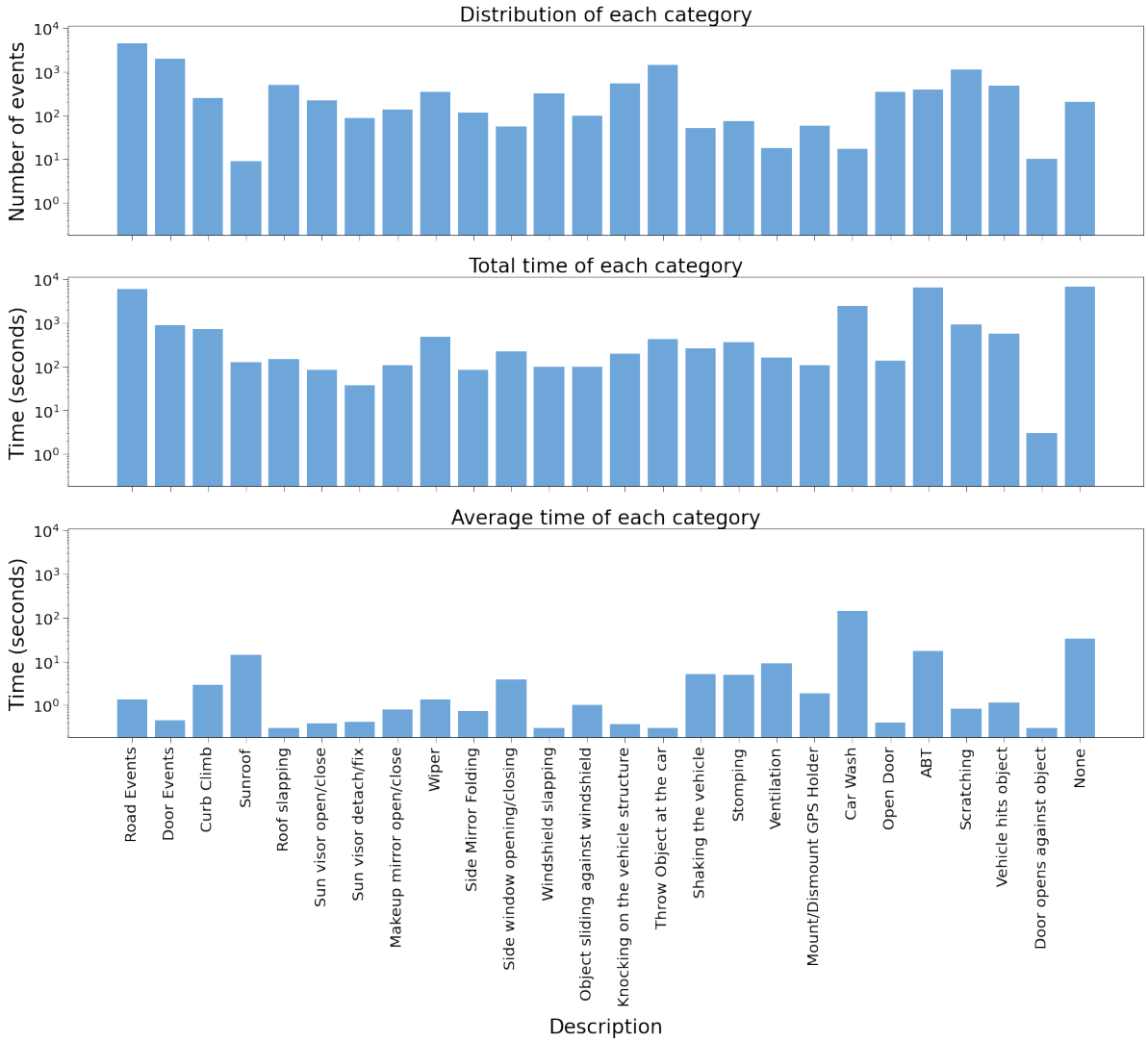


Figure 15: Statistical analysis of categories

Considering all the class categories presented on the dataset, the events characterized as “Road Events“ have a higher presence and events considering the “Sunroof“ of the vehicle have the lower presence.

The “Car Wash“ is the category with a higher average time, but considering that the process is time-consuming, the average of 300 seconds (4 minutes) is an understandable value.

Another important conclusion is that the average time for a lot of impact related events is 1 second, or close, which could mean that a 1 second window could obtain the signal information from most of impact-related events. The events with a higher average time are the ones where the occurrence of damage is not expected, such as “Car Watch“ and the actions involving the “Sunroof“.

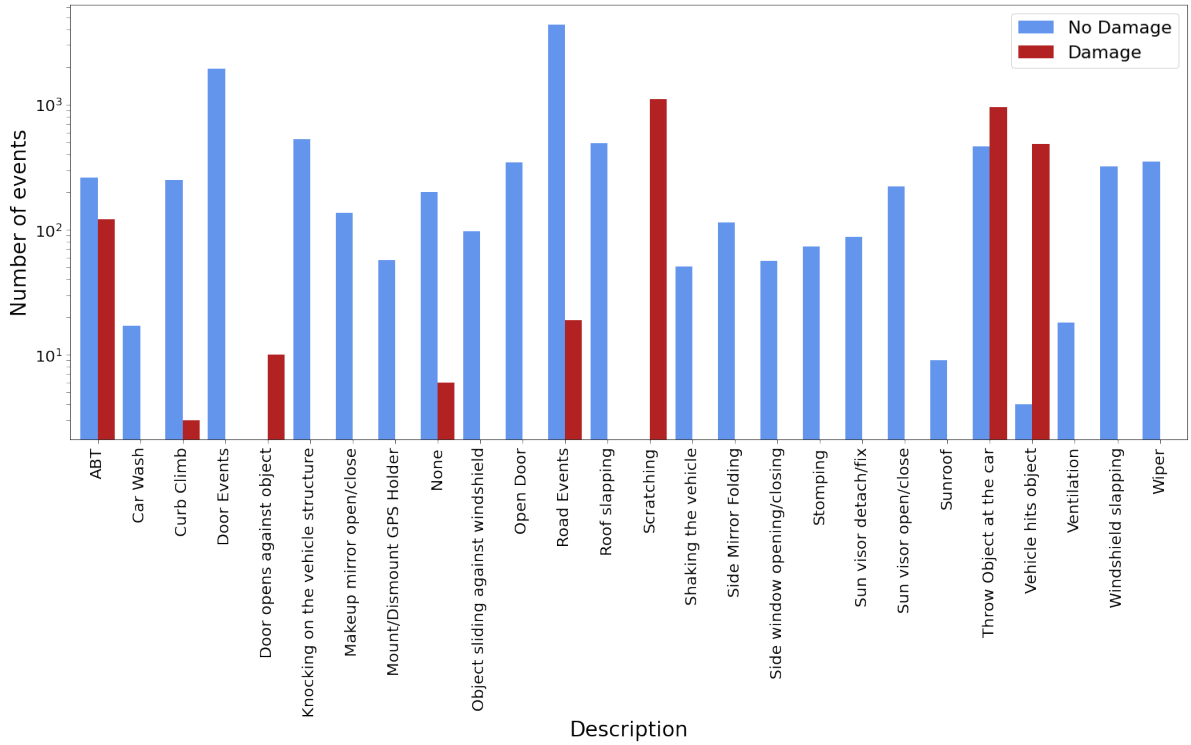


Figure 16: Damage/no damage distribution by events

Figure 16 gives a specific analysis on the distribution of the damage and no damage events in each of the events categories.

It would be expected that impact related classes of events to be characterized, by default, as damaging occurrences. The classes mentioned that involve impacts are the following:

- Throw Object at the car
- Scratching
- Vehicle hits object
- Door opens against object

On the data recorded, scratching and door opens against object always involve damage to the vehicle however, the other impact related classes have also non damaging events recorded. The impact with a low density object does not result in damage to a vehicle, this being a possibility of a non damaging data collected from each of the classes presented.

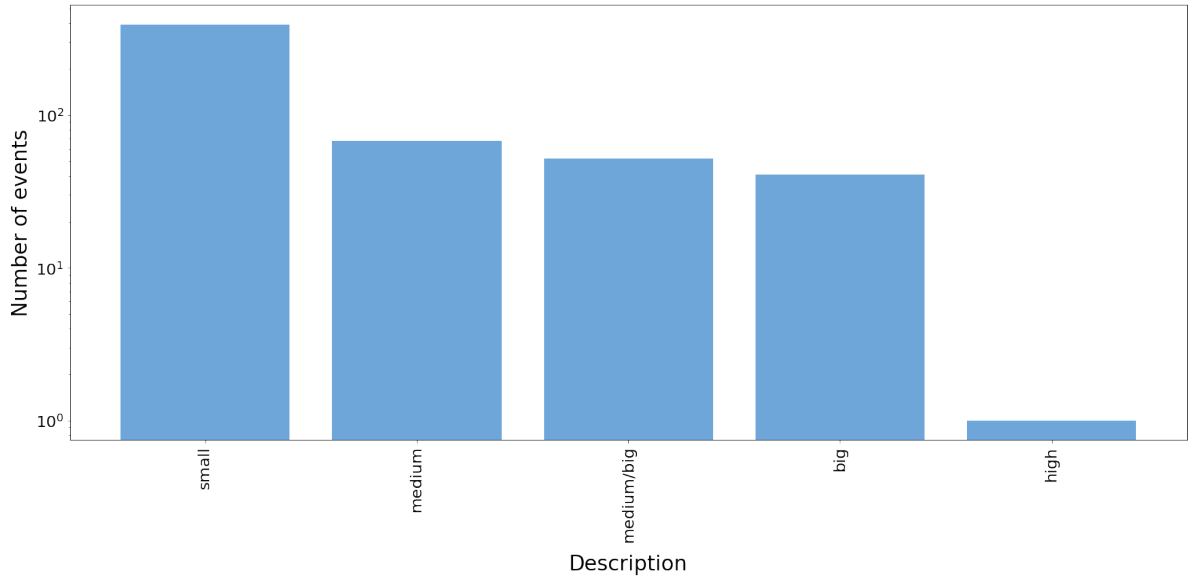


Figure 17: Distribution of the damage severity

Considering the data that resulted in damage, the damage severity on the vehicle was divided by levels: small, medium, medium/big, big and high.

Most of the damaging events captured on the data collection phase resulted in small damages to the vehicle, with only a small volume of high damage data collected.

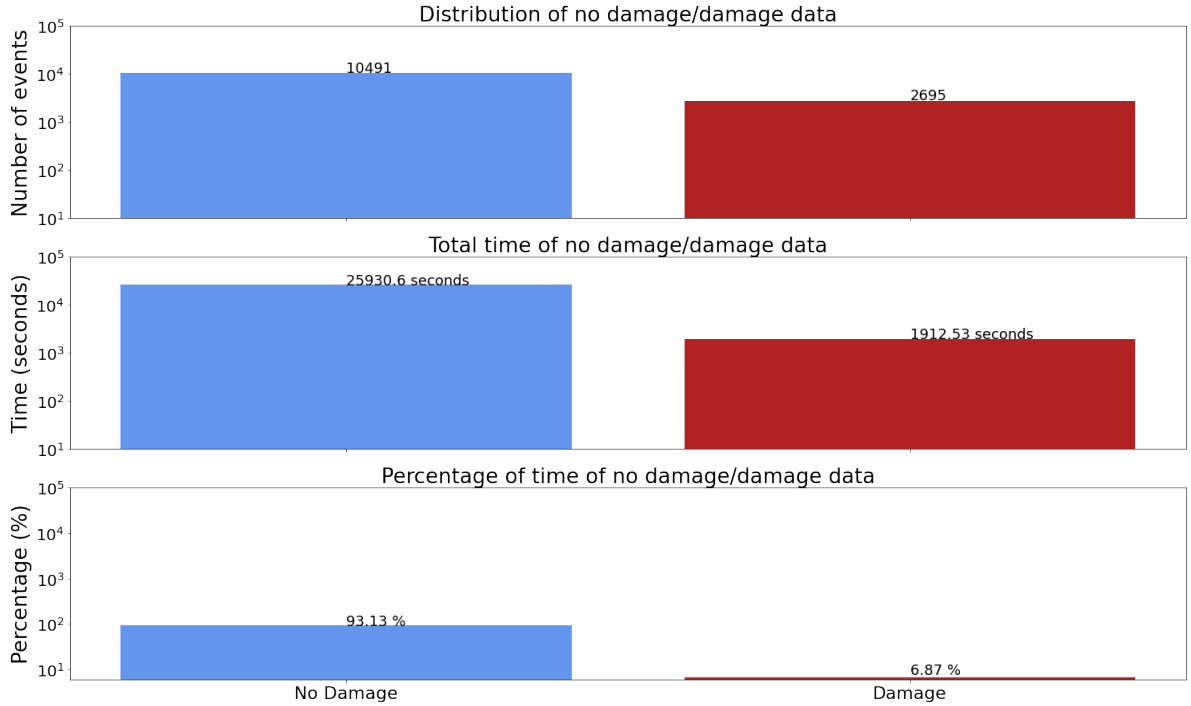
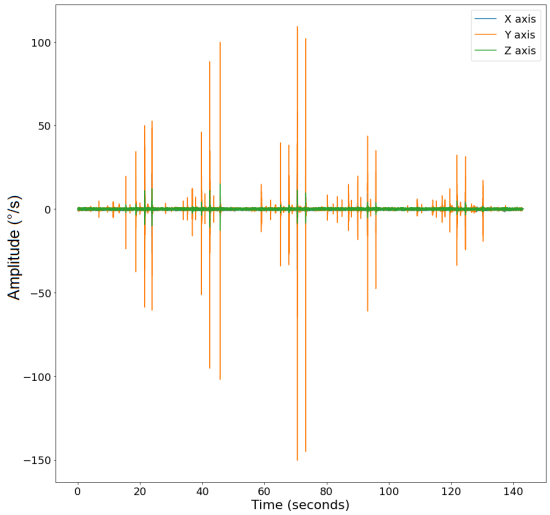


Figure 18: Statistical analysis of the damage/no damage data

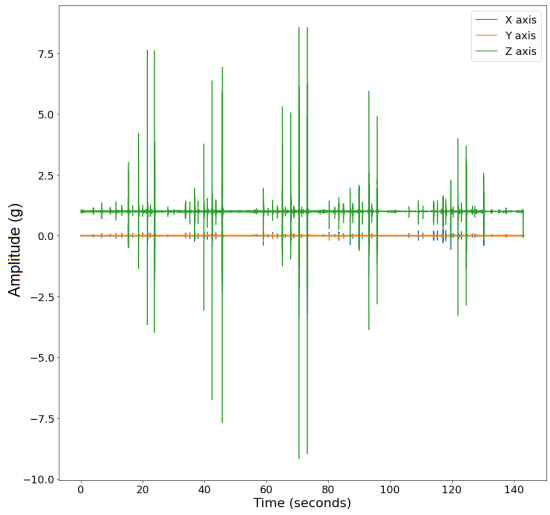
Looking at all the manually labeled data as an event, that resulted in damage or not, is possible to conclude that the number of damaging events captured is roughly 1/5 of the total of all events. In terms of time captured, the difference is even higher with damaging events corresponding to 6.87% of the total data.

5.1.2.2 Event depth analysis

Firstly, an analysis of the gyroscope, accelerometer and microphone data was performed, to better define which sensors to use for the resolution of the problematic.



(a)



(b)

Figure 19: (a) gyroscope and (b) accelerometer representation of a high energy door event.

As seen in Figure 19, the information obtained from the accelerometer and gyroscope is similar, where there is a response from both sensors when they are excited by the occurrence of an event. The use of the gyroscope sensor data, joined with the accelerometer data, would not bring additional information. As a result of this study, only the accelerometer data will be contemplated on the future phases.

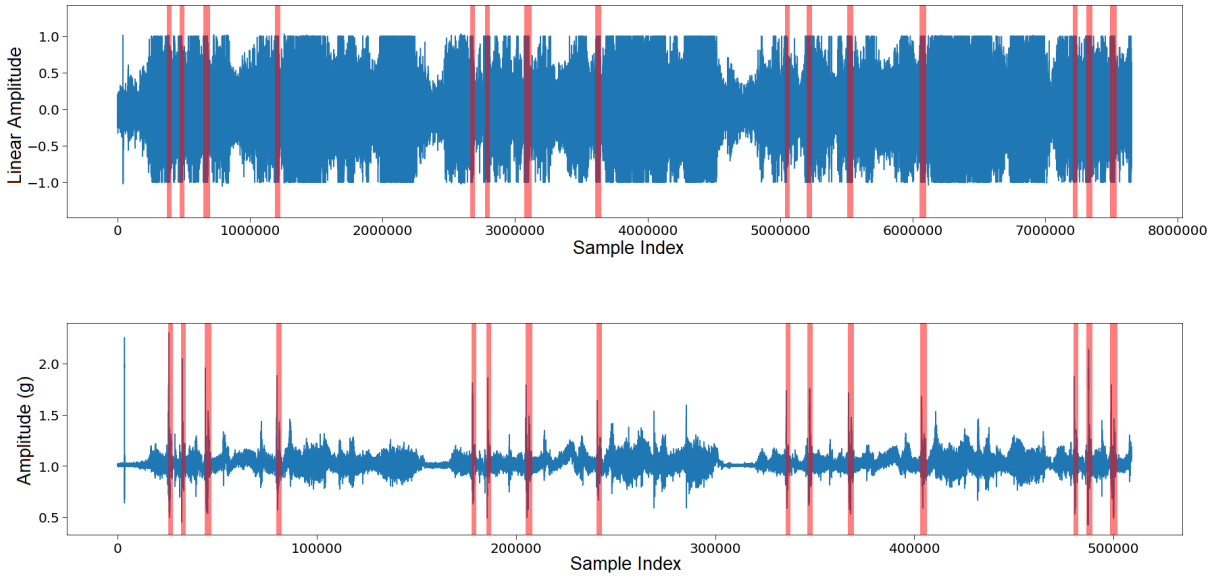


Figure 20: Saturation of microphone data (above) compared to the accelerometer data (bellow) with identification of labeled events.

Figure 20 allows to perceive that the microphone data, even when there is not an occurrence of an event, there is a large portion of data with high saturated audio data. Being that the microphone system is independent from the accelerometer and gyroscope setup, the audio data has also synchronization problems regarding the accelerometer data.

The problem will then be analyzed from the perspective of using the accelerometer as the only sensor that provides valid or relevant data.

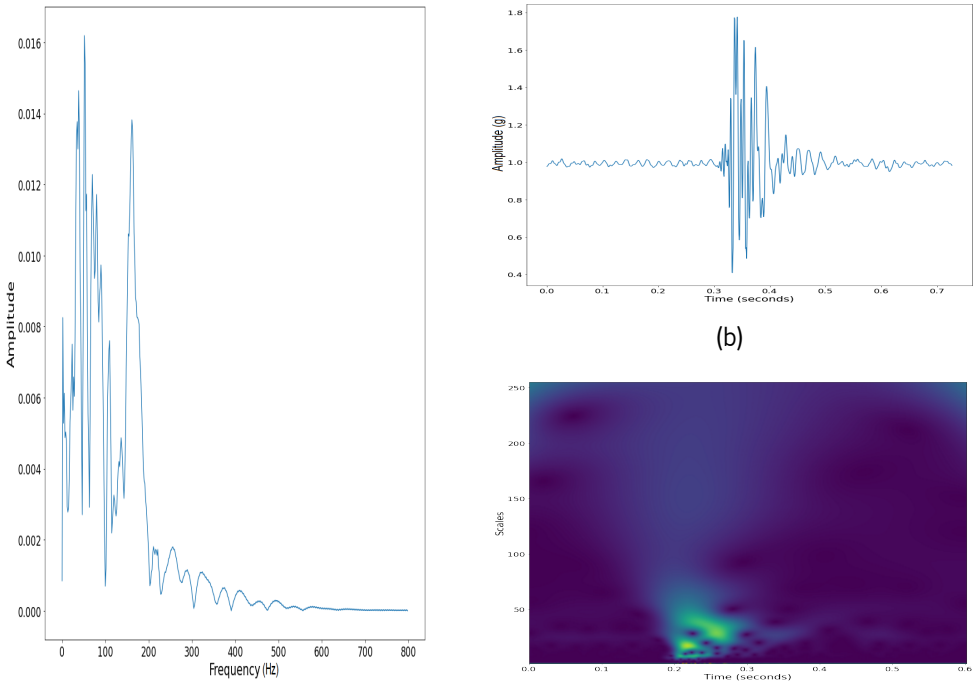
A study of damaging and non-damaging events will be carried out in order to better make conclusions and proceed on the Data Preparation step.

The analysis of the events were made considering the information from the accelerometer signal, the Fast Fourier Transform (FFT) and the wavelet (Discrete Wavelet Transform (DWT)).

Wavelet algorithms process data at different scales or resolutions. The FFT and the DWT are both linear operations that generate a data structure that contains $\log_2 n$ segments of various lengths, usually filling and transforming it into a different data vector of length 2^n, however wavelets are also located in space (Graps, 1995).

The signal analysis was carried by using the XYZ norm of the signal. Considering that the Z axis of the accelerometer is affected by the gravity, a measure of 1g was reduced on the data analyzed though FFT in order to remove the influence of gravity and reduce the representative peak at 0Hz.

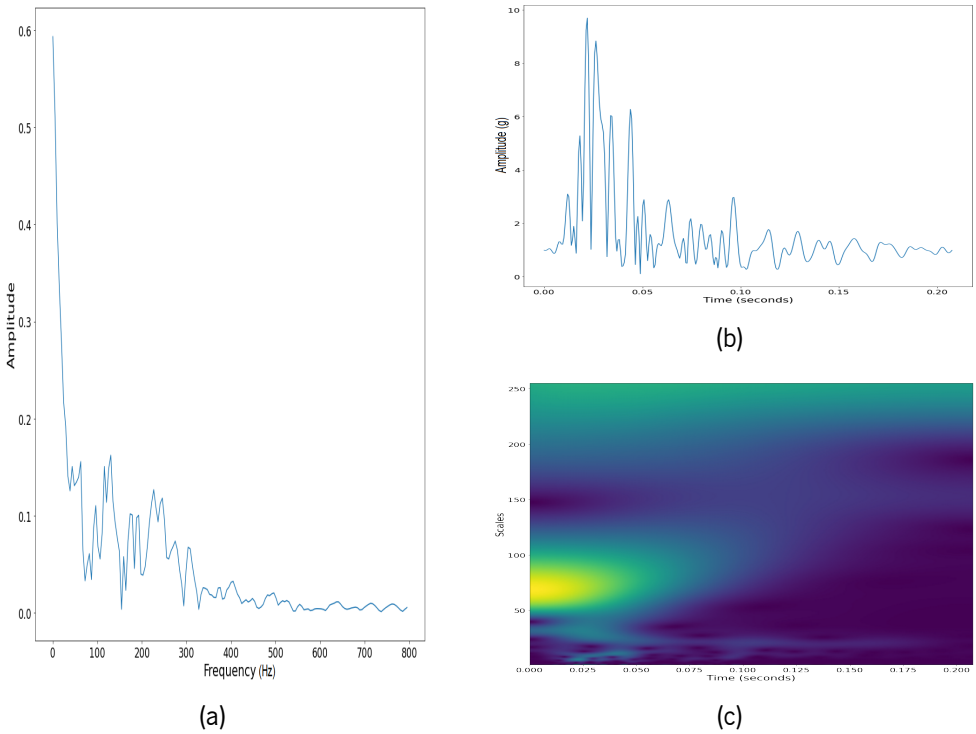
Non-damaging events



(a) (b) (c)

Figure 21: (a) Fast Fourier Transform, (b) signal representation and (c) wavelet representation of a regular door event.

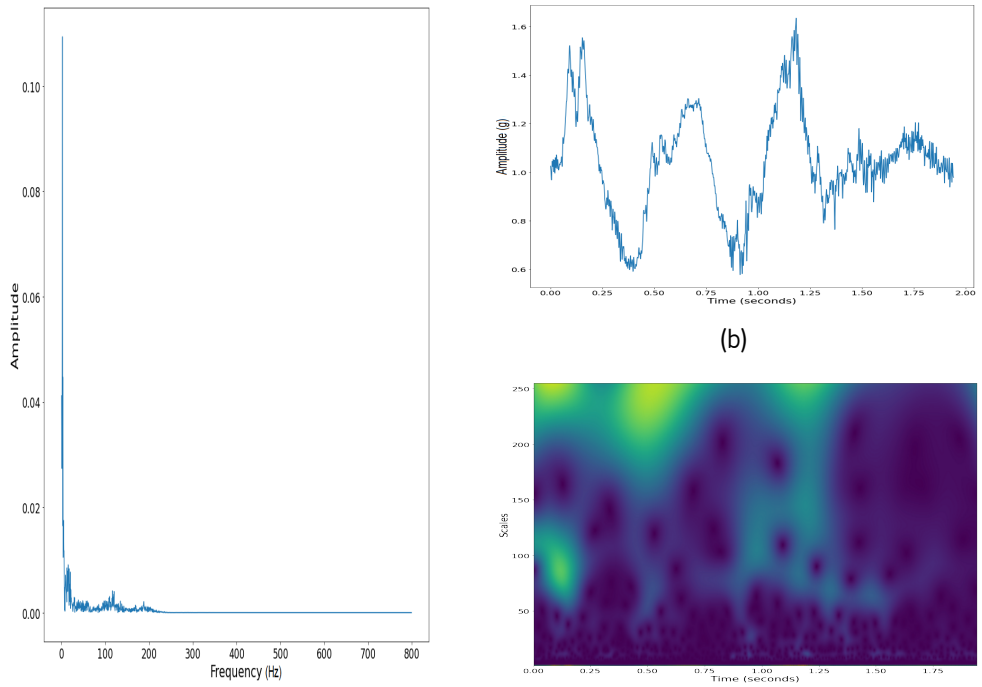
The signal on the Figure 21 is a representation of a normal recording of an event of doors. An event of doors has a low amplitude (Figure 21b) and almost no representation on the frequencies above 200Hz.



(a) (b) (c)

Figure 22: (a) Fast Fourier Transform, (b) signal representation and (c) wavelet representation of a high event of doors.

The Figure 22 represents the highest recording of a door event on the dataset. The amplitude reaches a value of 10g (Figure 22b) and there is presence of frequencies on the higher range, from 200 to 300Hz (Figure 22a). The occurrence of the highest frequencies happened where the signal excitation was higher, that is, reached the highest amplitude (Figure 22c).

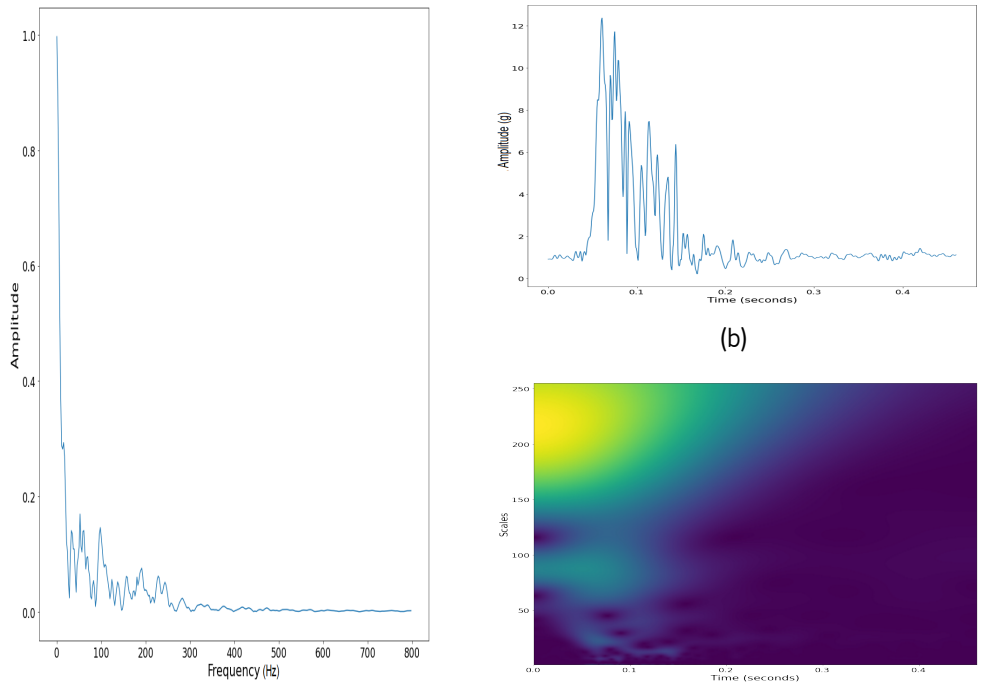


(a) (b) (c)

Figure 23: (a) Fast Fourier Transform, (b) signal representation and (c) wavelet representation of a speed bump.

The signal that represents a speed bump is really characteristic of this event, where its possible to visualize the passage of the two front wheels, the body between the wheels and the back portion (Figure 23b). In terms of frequencies, the speed bump event has a stronger presence on the lowest frequencies (Figures 23a and 23c).

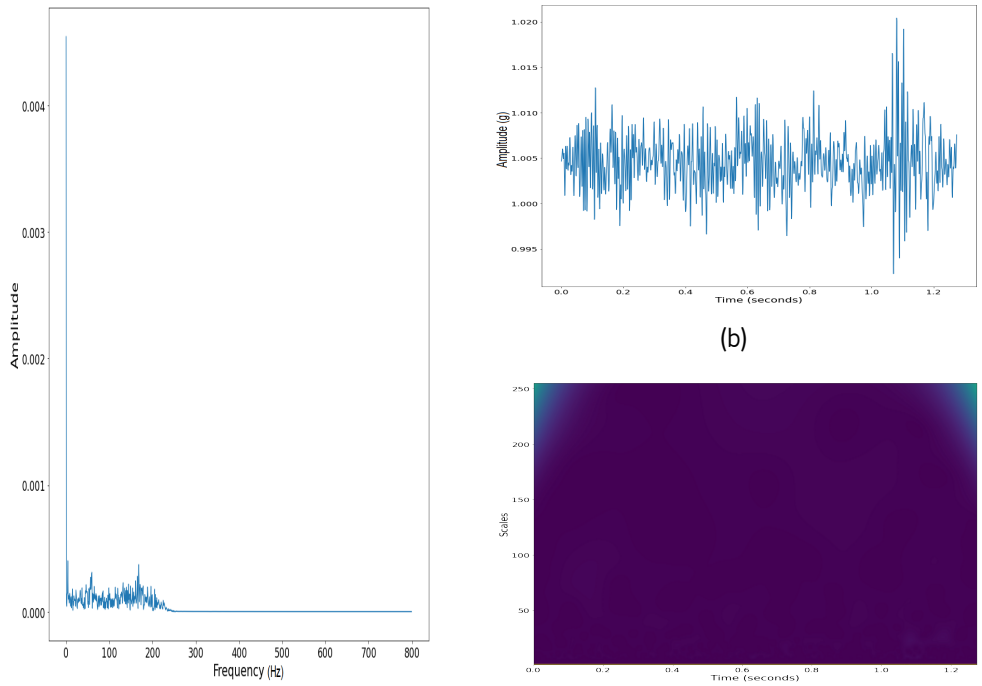
Damaging events



(a) (b) (c)

Figure 24: (a) Fast Fourier Transform, (b) signal representation and (c) wavelet representation of a high impact event of vehicle hits object.

The impact on Figure 24 releases a lot of energy, reaching an amplitude of 12g (Figure 24b). The event had a representation of a wide range of frequencies (Figure 24c), with frequencies above the 200Hz having a high presence on the FFT graph (Figure 24). Consequently, is possible to bring to a conclusion that high impacts have higher amplitudes and a wider range of excited frequencies.

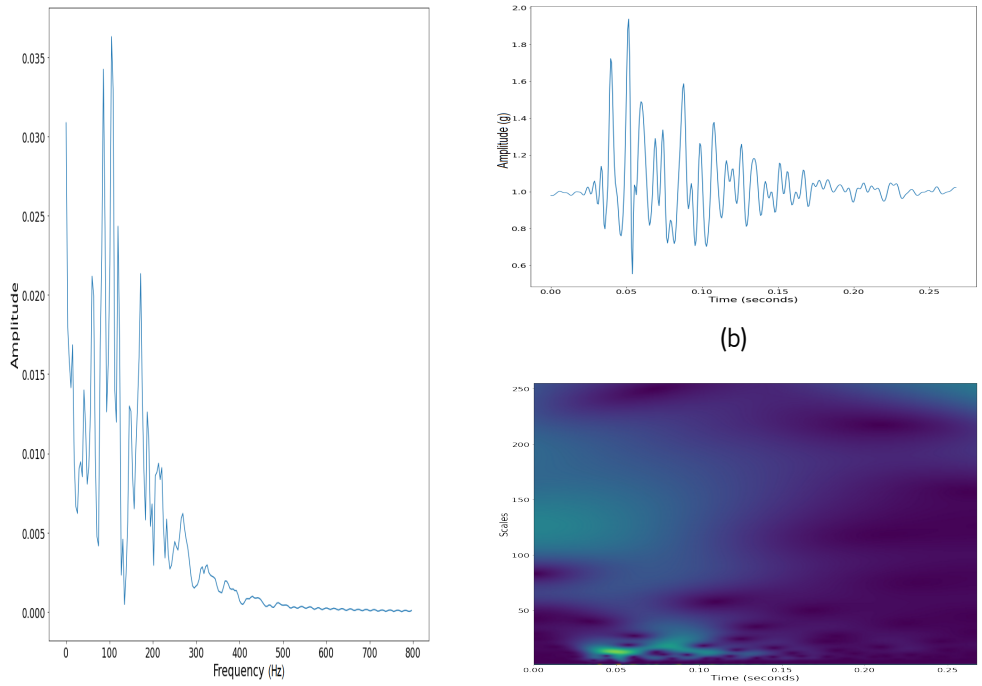


(a) (b) (c)

Figure 25: (a) Fast Fourier Transform, (b) signal representation and (c) wavelet representation of a scratching event.

The event of scratching is not possible to be characterized by the accelerometer data. Taking into account that the signal represented on the Figure 25b considers the information from the gravity, the signal only heights more 0.02g than the axis represented by the information of the gravity (Z axis), where the representative value is 1g.

Furthermore, the information from the FFT (Figure 25a) and the DWT (Figure 25c) also corroborates this conclusion.



(a) (b) (c)

Figure 26: (a) Fast Fourier Transform, (b) signal representation and (c) wavelet representation of a door opening against object.

The event of a door opening against an object (Figure 26). The impact is highly represented on the range of frequencies between 100 and 200Hz (Figure 26a). However, even through the event is characterized as being an impact, the amplitude reached is lower than the one on the Figure 22, representative of a door closing. Therefore, is possible to conclude that a door event can have a similar or higher representation that those of a impact-related signal.

Chapter

6 Use Case: SlimScaley - Data Preparation and Modeling phases

After the processes to understand the raw data available to solve the problem by an Anomaly Detection approach, the following phases where conducted and are described on this chapter:

- Data Preparation
- Modeling
- Model Evaluation and Results Analysis

6.1 Data Preparation

As explained on the Chapter 3, the Data Preparation step is paramount to the modeling task. It is an essential step as a prerequisite to put data in context in order to turn it into insights and eliminate bias resulting from poor data quality.

A threshold-based filtering strategy, data cleaning and windowing strategy were performed on the data before the application of Feature Engineering task on the accelerometer data.

6.1.1 Distribution Analysis and Threshold Definition

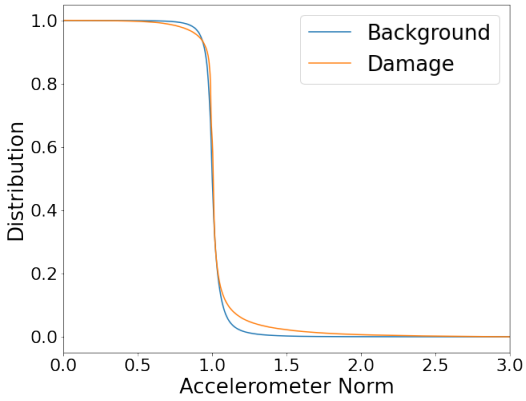
Some events of damage and non-damage have low energies associated with it, resulting in little or no defining or differentiating characteristics to this signals.

Also, considering the application of the system on a real environment, if the event classification system did not have a triggering mechanism and was inferring all the time, it could result in a system overload and delay on the evaluation process.

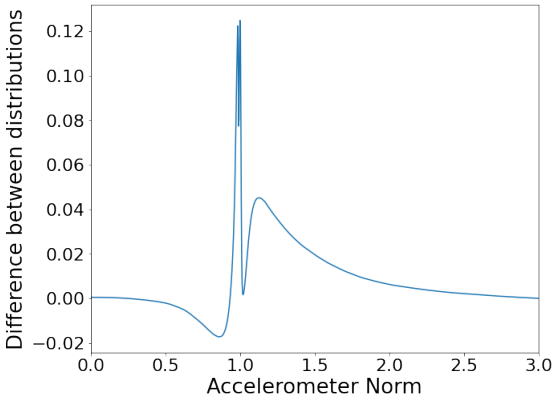
Considering this aspects, a filtering strategy was implemented in order to separate relevant from irrelevant events.

A reduction of the background/non-damaging events analyzed by the model results in a better defined strategy considering that the project objective is to detect the damaging events.

The definition of a threshold for separation of relevant and irrelevant events was made taking into account the norm of the accelerometer axis of all the samples.



(a) Sample Inverted Cumulative Distribution Function.



(b) Difference between distributions.

Figure 27: Distribution analysis of damage/no damage data.

Firstly, an analysis of the Cumulative Distribution Function (CDF) of all events of the data set, defined why its label, was performed (Figure 27).

The CDF implies that distribution function of X, evaluated at x, is the probability that X will take a value less than or equal to x (Deisenroth et al., 2020).

Looking at the Figure 27a, the inverted of the CDF is represented, the definition is: distribution function of X, evaluated at x, is the probability that X will take a value higher than or equal to x.

Considering this definition, the range of values between 1.0 and 1.5 on the accelerometer norm represent a value where the distribution of the damage events is higher than those of background. A value in this range would favor the number of damage events selected to non-damage events. However, as seen on the peaks of the Figure 27b, the values closer to 1.0 should be ignored because of abrupt changes, which can indicate that those values are not stable.

The lowest possible value between the defined range should be considered as the threshold, seeing that the values closer to 1.5 should also be ignored because of its ability to filter out more damaging events, when the objective is to maximize the number of those events.

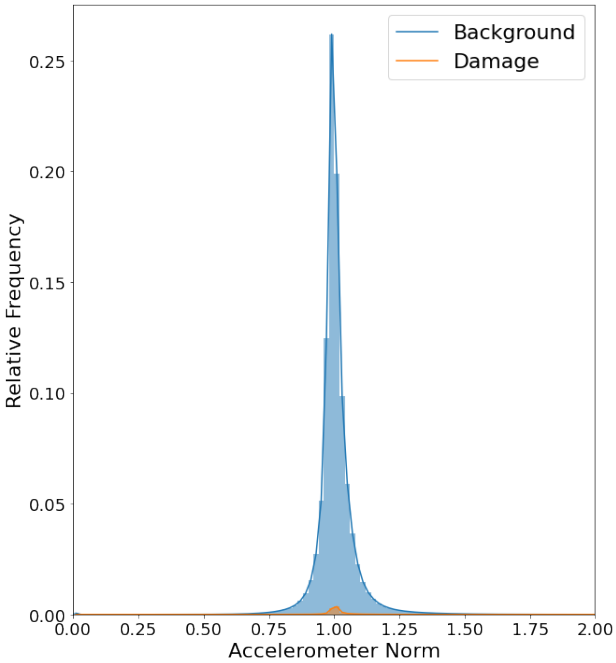


Figure 28: Histogram of damage/no damage data.

The final value was chosen from an analysis of the histogram of the damage and no damage data (Figure 28), where the 95th percentile on the normal data corresponds to the accelerometer norm of 1.2g.

Taking into account the analysis carried out by the application of a CDF and histogram on the data, 1.2 was considered to be the best value for the threshold application.

6.1.2 Data Cleaning

After the definition of the threshold value that is best applicable to the problem, and before the data set design, some data cleaning was necessary to implement to the data.

Following an analysis of the labeled events, some labels were removed because they were considered to be incorrect where the initial sample is higher or equal than the final sample.

Furthermore, the events of scratching were also removed. As studied on the Exploratory Data Analysis (EDA) phase, the scratching events are not apprehended by the accelerometer data. Even though scratches result in damage to the vehicle, the usage of those events on the models would result on a wrongful classification. For the identification of scratches, an analysis with more sensors, for instance a microphone, could result on those events to be well classified.

6.1.3 Dataset Design

The data involved on the creation of the windows for the train and test datasets were specifically from different folders. Because of the importance of the test set in terms of analyzing the performance of the models, the folder of the testset data had 0 background events manually classified as damage, that is, had 0 False Positives (FPs). This characteristic is crucial for the correct classification of the damaging events

as anomalies.

The distribution of the manually labeled events throughout the folders identified as train (Figure 29) and test (Figure 30) sets is possible to analyze on the next figures.

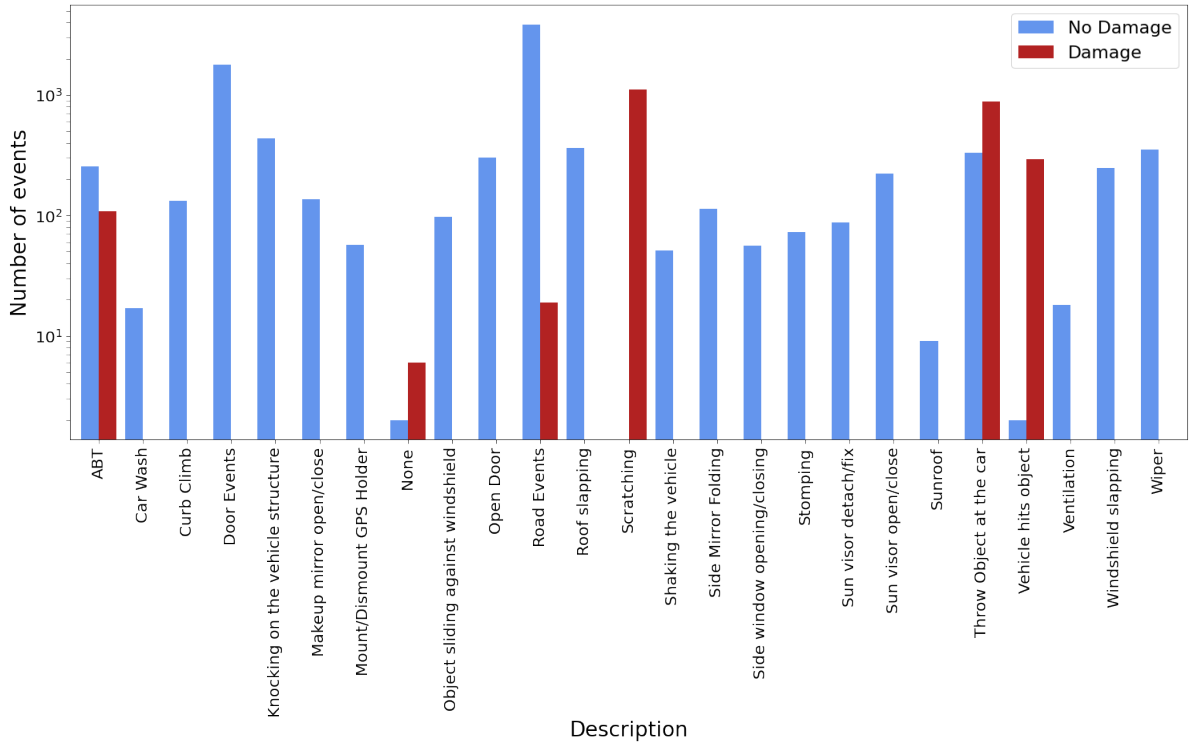


Figure 29: Distribution of the train dataset with manually labeled events.

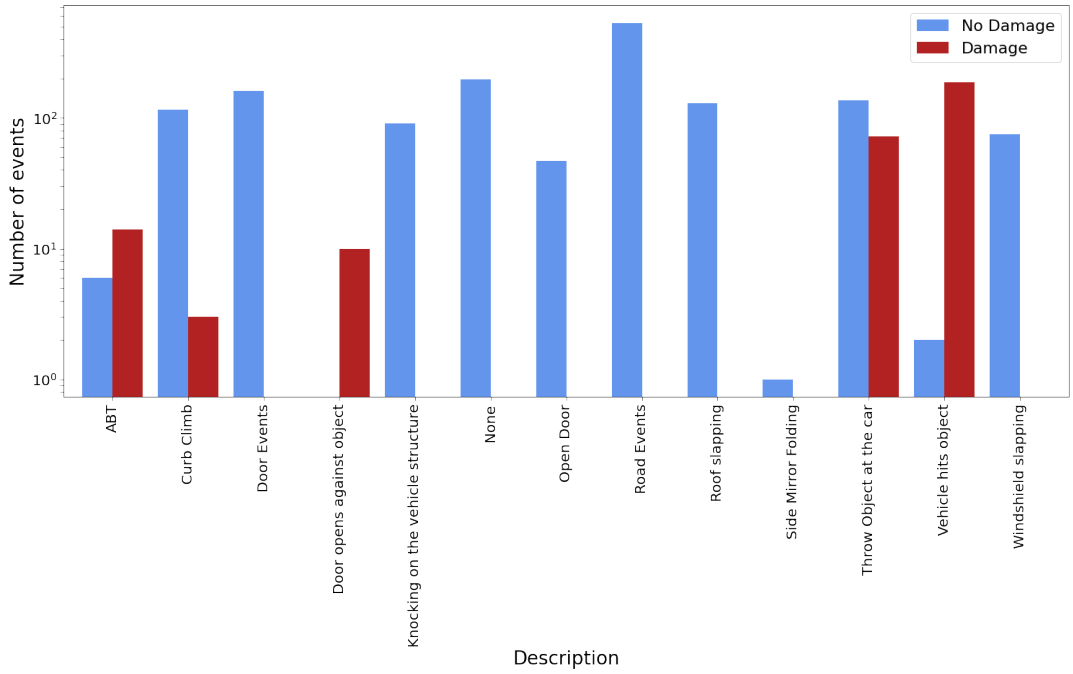


Figure 30: Distribution of the test dataset with manually labeled events.

Taking into consideration the different size windows of the manually labeled events, as seen on the Subsection 5.1.2, a processing and standardization of the window size needed to be processed in order

to use the accelerometer for Feature Engineering tasks. A non-standardized data would lead to incorrect feature creation as, for instance, features like the mean are extremely influenced by the size of the windows.

The analysis and construction of the datasets followed some specifications considering the two classes of data: damage (anomaly) and non-damaging/background (normal) data. This being said, the dataset was built differently for damaging events and non damaging events.

Considering that the damaging events are the subject of study, ensuring that the window capture the relevant information of this events is crucial for a well defined data set.

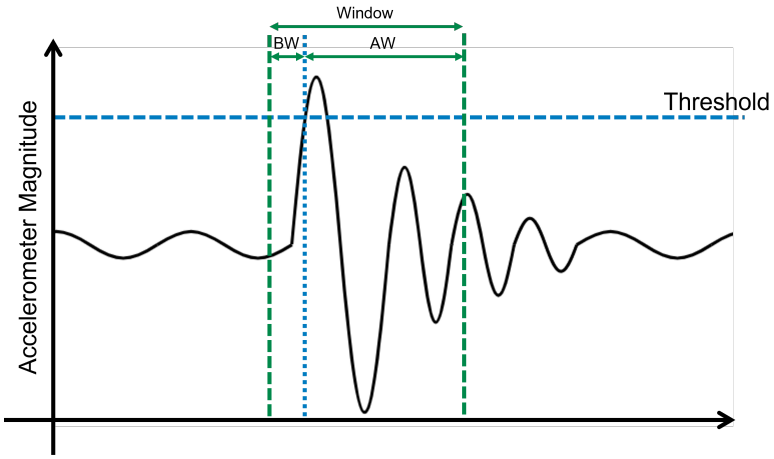


Figure 31: Methodology of construction of the dataset window. After Window (AW) and Before Window (BW) correspond to 0.9 and 0.1 seconds, respectively.

To define an event, anomalous or normal, two values had to be specified: the size of the window that frames an event and the position in the window of the first value above the threshold. To have a continuous analysis of events on an automotive system, a window of 1 second was defined, as a bigger window size could delay the response in case of an impact, which corresponds to 1600 samples of accelerometer data, as it has a sample rate of 1600Hz. Considering the ripple effect on the signal when an event occurs, the window was constructed considering 10% of the window size before the first value above the threshold (BW), that is, 160 samples, and the 1439 samples after the defined value (AW). The windows constructed where non-overlapping windows, as an overlap would allow the same event to be analyzed more than one time (Figure 31).

Damage events where designed based on their initial labeled window, were the first XYZ L2 norm (Jeffrey et al., 2000) value, as the XYZ norm provides the overall magnitude of the forces, above the threshold was framed on a window with 1600 samples. The information of the X, Y and Z data on this window was saved, as well as the associated label. “None“ labeled damaging events were not regarded, as a well defined event classification was crucial to the analysis.

Looking at the creation of the windows of the damaging events, if the background windows did not follow the same parameters because:

- Only the information on the initial labeled window would be regarded on the creation of the background window.
- There would be only one background window for each of the initial labeled window.

Considering that a high amount of data from background is important for the application of an Anomaly Detection approach, and the ability for the system to then be able to recognize background events that exceed the threshold as non-damaging, the selection of non damage events followed a new set of rules discarding the information of the manual labels. Files where there was the occurrence of damage were not considered on the creation of background events, regarding to the possibility of residual damaging information, that was not included on the window frame, being considered as background.

In any recording, that did not have damage events, the positions of all reference indexes from the XYZ norm that exceeded the threshold were calculated and, after that, a non overlapping sliding window created background events considering the positions of these values. Just like on the damaging windows, the X, Y and Z axis data was saved. The background events were labeled based on the occurrence of an labeled event in the window and, if no manually labeled event is encountered, were labeled as “None“.

The creation of the windows on the train and test datasets have taken for consideration the process explained. The distribution of the damage and background windows on each dataset is detailed on the Figure 32.

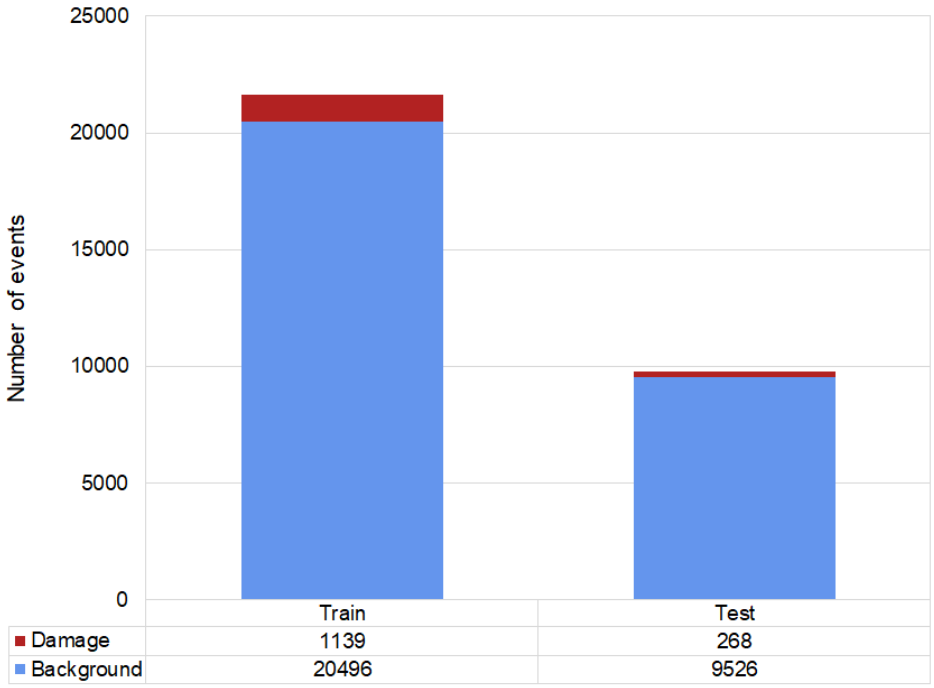


Figure 32: Distribution of the train and test datasets with events created on a 1 second window with a 0.1 reference index position regarding the window size.

6.1.4 Feature Engineering

The next phase after the dataset was created with the necessary structure, as well as the events filtered, the application of Feature Engineering tasks was necessary for a better explainability of both classes. The Feature Engineering tasks were supported by the information obtained on the EDA phase (Subsection 5.1.2).

The accelerometer event, as described above, was constructed by a windows of 1600 data points for each of the accelerometer axis: X, Y and Z.

Regarding the frequencies up to 10Hz, because they where present on all events and were considered as noisy data, a highpass filter of 10Hz was applied to the accelerometer data on all axis.

The L2 norms XYZ, XY, XZ and YZ, calculated at each sample in the event window, was also incorporated onto the data to analyze, as the XYZ norm provides the overall magnitude of the forces, the same applies to the other norms however their values only concern two axis. The information derived from the calculation of the norms results in the distance of the vector coordinate from the origin of the vector space, resulting in a positive distance value. The application of the norm removes the directionality of the applied force, as a negative force applied on the X and Y axis would result on a positive XY L2 norm value. The L2 norm is often used when fitting machine learning algorithms as a regularization method, e.g. a method to keep the coefficients of the model small and, in turn, the model less complex (Briggs et al., 2000).

Considering the specific frequency ranges that are present in damaging and non-damaging events, a wavelet transform was applied to the norms (XYZ, XY XZ and YZ) and axis (X, Y and Z). The Table 19 gives information about the wavelet scales chosen and the corresponding frequency.

Scale	8	16	32	64	128
Frequency	162.5	81.25	40.625	20.3125	10.15625

Table 19: Association between each wavelet scale and the respective frequency.

Besides the frequency analysis made using wavelets, a bandpass filter on the range [200, 300Hz] was also extracted of the norms and axis because of the presence of this range of frequencies on damaging events.

After the analysis, each event is then composed of 49 windowed streams of data.

In terms of application of Feature Extraction, a set of features from different domains were extracted from each of the windowed streams. The analysis carried out on the Section 2.2 Literature Review supported the selection of features presented on the Table 20.

Domain	**Features calculated**
Time	Autocorrelation, Zero Crossings, Peak to Peak Distance, Count Mean Crossings, Negative Turning, Positive Turning, Absolute Energy, Mean Differences, Median Differences, Distance, Sum of Absolute Differences, Slope, Area Under Curve, Absolute Sum of Changes, Count Above Mean, Count Bellow Mean, First Max Location, First Min Location, Mean Absolute Differences
Statistical	Kurtosis, Skewness, Root Mean Square, Median Absolute Deviation, Interquartile Range, Variance, Standard Deviation, Mean Absolute Deviation, Mean, Median, Max, Min, 5th Percentile, 25th Percentile, 75th Percentile, 95th Percentile, Range
Spectral	Total Energy, Spectral Distance

Table 20: Features extracted separated by domain.

A total of 38 features are presented on the Table 20, out of those 19 are time related features, 17 are statistical and 2 spectral features.

Besides the features described on the Table 20, the Signal Magnitude Area (SMA) on the raw accelerometer axis data was also calculated:

$$SMA = \sum_{i=1}^{n} (|X(i)|) + (|Y(i)|) + (|Z(i)|)$$

The number of features obtained from the accelerometer data after the Feature Extraction phase, that was used to describe each event, was 1863.

The curse of dimensionality (Bellman, 1957) refers to various phenomena that arise when analyzing and organizing data in high-dimensional spaces that do not occur in low-dimensional settings. Considering the amount of features extracted from each event, Feature Selection was necessary to apply. Taking into account the fact that the algorithms used in the modeling phase are semi-supervised and therefore have no associated Feature Selection techniques, the application of Feature Selection techniques was developed based on the *scikit-learn* (Pedregosa et al., 2011) approaches: SelectKBest, VarianceThreshold and SelectFpr. SelectKBest and SelectFpr are supervised techniques, while VarianceThreshold is unsupervised.

Considering the problematic of high correlated features, where redundant information from two or more features are used on the model, the features with a 95% or higher correlation where discarded.

VarianceThreshold removed all features that remained constant throughout the dataset and also quasi-constant features. The application of VarianceThreshold, and the removal of high correlated features, resulted in a removal of 196 features, leaving 1667 features out of the initial 1863. This two strategies where always applied for its ability to remove redundant features out of the dataset.

SelectKBest, as the name imply, selects the features based on the k highest scores. The score function chosen was chi2, that computes chi-squared stats between each non-negative feature and class. To apply this technique the features need to be non-negative, so a normalization of the data to the range [0,1] was performed. Then, the 10% highest scored features were selected, which corresponds to the selection of 166 features.

The SelectFpr filters the features by its p-values below alpha based on a False Positive Rate (FPR) test. The score function, similar to the SelectKBest technique, was chi2. The alpha chosen was 0.01, which means that features with p-values less than 0.01 were selected. The application of the SelectFpr resulted in 634 features selected out of 1667.

The completion of the application of Feature Selection techniques resulted in 3 datasets with 1667, 166 and 634 features that where trained and tested for the problematic.

The only Feature Reduction approach used was Uniform Manifold Approximation and Projection (UMAP). UMAP (McInnes et al., 2020; Sainburg et al., 2021), firstly published in 2018, was used to project the raw accelerometer data into a two-dimensional space. The UMAP was instantiate as the dimensionality reducer with target dimensionality n_components = 2, resulting in a 2D representation of the data. UMAP is faster than the widely used Feature Reduction unsupervised linear transformation technique Principal Component Analysis (PCA) and retains the global data structure much better (Albrecht et al., 2020). The UMAP was applied to the raw axis (X, Y and Z) accelerometer windowed data, due to the ability of UMAP

to represent the information of the accelerometer on a different dimension (Figure 33).

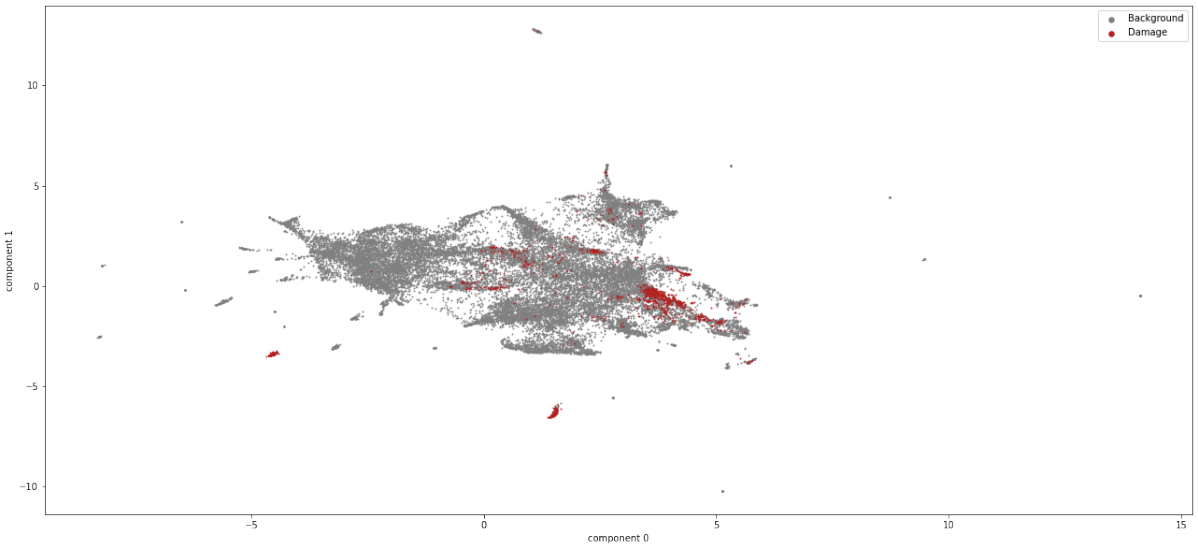


Figure 33: UMAP application on the accelerometer data.

The objective of the usage of UMAP was to try to separate the damage and background data. In Figure 33 is possible to see the red clusters, representative of damaging data. However, is not possible to complitely separate the damage from the background data using only this approach, as some of damaging representative points are scattered throughout the figure.

6.2 Modeling

Considering the aspect that the labels of the windows were available, the emphasis was on a semi-supervised Anomaly Detection approach because of its potential to focus on having appropriate explainability of one of the classes to fit the model.

Semi-supervised Anomaly Detection takes into consideration one of the classes to proceed with the training of the algorithms. Taking into account the requisite of Anomaly Detection of having a substantial difference between the normal class and the anomalous and a larger representation of normal data available, the normal class was regarded as being the one with the non-damaging events. The validation and test of the model is when the anomalous/damaging data is regarded.

For a search of the best hyperparameters for each model that best fit the data, a GridSearch was performed. Measuring the models based on a non-nested cross validation bias the model to the dataset, resulting in an overly-optimistic score of the model with a specific set of hyperparameters (Cawley et al., 2010). A nested cross validation with grid search was then implemented to choose the best hyperparameters for each algorithm and assert the model quality.

The classical nested cross-validation procedure, just as the one presented on *scikit-learn* (Pedregosa et al., 2011), version 1.1.0, is used in a supervised ambient as it takes the labels for the evaluation of the models. A new semi-supervised nested cross-validation procedure was created for this proposal, taking into consideration the structure and logic behind the traditional nested cross-validation. In this process,

only the training data was used.

On an Anomaly Detection application, the normal data is represented by a 1 and the anomalies by a -1. The training dataset is only composed of background (normal) data, which means that the associated label would be 1.

Taking into consideration that Anomaly Detection algorithms give a score value to each sample, normally positive for inlier (normal) and negative for outlier, the objective was to have the highest mean among the training data. The models with a higher mean of scores meant a better a fit for the model because it would have considered the data to have fewer outliers. Taking into consideration that the fit of the models would have only background (normal) data, a higher mean score of a model compared to another would mean that the first had more correctly identified normal data that the latter.

Algorithm 2: k-Fold Nested Cross-Validation with hyperparameter tuning

```
function NestedCrossValidation;
Input: M: model to Nest Cross Validate
Input: H_sets: set of hyperparameters for the model
Input: D: training data
Input: K_1: number of outer folds
Input: K_2: number of inner folds
for i = 1 to K_1 splits do
    Split D into D_i^train, D_i^test for the i^th split
    for j = 1 to K_2 splits do
        Split D_i^train into D_j^train, D_j^test for the j^th split
        foreach h in H_sets do
            Train M on D_j^train with hyperparameter set h
            Compute scores S_j^test for M with D_j^test
        endfch
    endfor
    Select optimal hyperparameter set (h*) from H_sets where mean(S_j^test) is best
    Train M with D_i^train using h*
    Compute scores S_i^test for M using with D_i^test
endfor
```

The Algorithm 2 represents a generalized structure of a k-Fold Nested Cross-Validation with hyperparameter tuning. The definition of the best score model is dependent of the Machine Learning (ML) algorithm chosen, where on Isolation Forest, for example, the score is equal to the depth of the leaf containing the sample. This results on a lower score on a normal sample when comparing to an anomaly sample, as a result of the latter needing more splits on the tree to be defined.

The models used for training were the One-Class Support Vector Machine (OCSVM), Isolation Forest, Gaussian Mixture Model, Local Outlier Factor (LOF), k-Means and an integration of UMAP with OCSVM. Five outer and inner folds were chosen on the Nested Cross-Validation.

6.2.1 k-Means

The k-Means algorithm aims to partition n observations into k ($\leq$ n) clusters in which each observation belongs to the cluster with the nearest mean (centroid), minimizing the variance within the cluster.

Considering $D = x_1, \ldots, x_n$ as the data set to be clustered, k-means can be expressed by an objective function that depends on the proximities of the data points to the cluster centroids as follows:

$$\min_{\{m_k\}, 1 \leq k \leq K} \sum_{k=1}^{K} \sum_{x \epsilon C_k} \pi_x dist(\mathbf{x}, \mathbf{m}_k)$$

where π_x is the weight of $\mathbf{x}$, n_k is the number of data object assigned to cluster C_k, $\mathbf{m}_k = \sum_{x \epsilon C_k} \frac{\pi_x x}{n_k}$ is the centroid of cluster C_k, K is the number of clusters set by the user and the function "dist" computes the distance between object x and centroid $m_k, 1 \leq k \leq K$ (J. Wu, 2012).

Hyperparameter	Description
n_init	Number of times the k-means algorithm will be run with different centroid seeds. The final results will be the best output of n_init consecutive runs in terms of inertia.
max_iter	Maximum number of iterations of the k-means algorithm for a single run.
n_clusters	The number of clusters to form as well as the number of centroids to generate.

Table 21: Hyperparameters used on GridSearch of k-Means.

The n_clusters hyperparameter was defined as 2 and 3 for the GridSearch analysis. The definition of 3 clusters was to try to better separate events similar to damage as a different cluster.

6.2.2 One-Class Support Vector Machine

The objective of the Support Vector Machine (SVM) algorithm is to find a hyperplane in an N-dimensional space, with N number of features, that distinctly classifies the data points (Cortes et al., 1995).

Any hyperplane can be written as the set of points x that satisfy

$$w^T x - b = 0$$

where w is the normal vector of the hyperplane.

OCSVM, proposed by Müller et al. (2001) is similar to the classical SVM, but it instead uses a hyperplane which is far from the origin.

The process of the OCSVM can be described as follows (Schölkopf et al., 1999):

- Projection of the point to a higher dimensional space.
- Separation of all the data points from the origin in the feature space using a hyperplane.

- Unlike traditional SVM, where there is a use of soft margin for smoothness, there is a use of a parameter that fixes fraction of outliers in the data.
- Maximize the distance between the hyperplane and the origin.
- The points lying below the hyperplane and closer to origin are outliers.

Hyperparameter	Description
kernel	Specifies the kernel type to be used in the algorithm.
gamma	Kernel coefficient.
nu	An upper bound on the fraction of training errors and a lower bound of the fraction of support vectors.

Table 22: Hyperparameters used on GridSearch of OCSVM

The nu hyperparameter is important when applying an Anomaly Detection strategy with OCSVM as its fine-tuning can make the model better handle outliers and prevent overfitting.

6.2.3 Local Outlier Factor

LOF was proposed by Breunig et al. (2000) is a well-known local anomaly detection algorithm and also introduced the idea of local anomalies. To calculate the LOF score, three steps have to be computed (Goldstein et al., 2016):

1. The k-Nearest Neighbors (k-NN) have to be found for each record x. In case of distance tie of the k^{th} neighbor, more than k neighbors are used.
2. Using these k-NN N_k, the local density for a record is estimated by computing the Local Reachability Density (LRD):
$$LRD_k(x) = \left(\frac{\sum_{o \in N_k(x)} d_k(x,o)}{|N_k(x)|} \right)^{-1}$$
whereas $d_k(\cdot)$ is the reachability distance of the k-NN.
3. The LOF score is computed by comparing the LRD of a record with the LRDs of its k neighbors:
$$LOF(x) = \frac{\sum_{o \in N_k(x)} \frac{LRD_k(o)}{LRD_k(x)}}{|N_k(x)|}$$

The LOF score is thus basically a ratio of local densities.

k-NN algorithm is a method proposed by Cover et al. (1967) and is used for classification and regression problems.

A k-NN algorithm is used based on the idea that similar data points are close to each other, that is, an instance should be similar to a majority of its k immediate neighbors, rather than to a centroid or an aggregate over a large set of data points.

Hyperparameter	Description
n_neighbors	Number of neighbors.
algorithm	Algorithm used to compute the nearest neighbors.
leaf_size	Leaf size, which can affect the speed of the construction and query, as well as the memory required to store the tree.
contamination	The amount of contamination of the data set, i.e. the proportion of outliers in the data set.
p	Parameter for the Minkowski metric.
novelty	Outlier Detection (novelty=False) or Novelty Detection (novelty=True)

Table 23: Hyperparameters used on GridSearch of LOF.

The hyperparameters studied on the application of the GridSearch technique were the ones described on Table 23.

Considering the p hyperparameter, the Minkowski distance, d_{mi}, of order ℓ between two points $p = (p_1, \ldots, p_d)$ and $q = (q_1, \ldots, q_d) \in \mathscr{D}$ is defined as:

$$d_{mi}(p, q) = \left(\sum_{i*1}^{d} |p_i - q_i|^l \right)^{\frac{1}{l}}$$

If $\ell = 2$ the Minkowski distance is equal to the Euclidean distance and for $\ell = 1$ this distance is equal to $\left(\sum_{i*1}^{d} |p_i - q_i| \right)$ and is known as the Manhattan distance or L_1 distance.

6.2.4 Gaussian Mixture Model

A Gaussian Mixture Model is a weighed sum of M component Gaussian densities as given by the equation

$$p(x|\lambda) = \sum_{i=1}^{M} w_i \, g(x|\mu_i, \Sigma_i)$$

where x is the D-dimensional continuous-valued data vector (features), w_i, $i = 1, \ldots,$ M, are the mixture weights, and $g(x|\mu_i, \Sigma_i)$, $i = 1, \ldots,$ M, are the component Gaussian densities (Reynolds, 2009).
Each Gaussian component in the mixture is comprised of the following parameters:

- A mean μ_i that defines its center.
- A covariance Σ_i that defines its width.
- A mixing probability pi that defines how big or small the Gaussian function will be.

Hyperparameter	Description
n_components	The number of mixture components.
reg_covar	Non-negative regularization added to the diagonal of covariance.
n_init	The number of initializations to perform.
covariance_type	String describing the type of covariance parameters to use.
max_iter	The number of Expectation-Maximization iterations to perform.

Table 24: Hyperparameters used on GridSearch of Gaussian Mixture Model.

A covariance matrix is symmetric positive definite, so the data, before being applied to the Gaussian Mixture model, had to be normalized to the range [0,1] to comply to this characteristic.

6.2.5 Isolation Forest

The Isolation Forest approach assumes that anomalies are easier to isolate from the rest of the data than normal instances. (F. T. Liu et al., 2008, 2012).

F. T. Liu et al. (2012) defines isolation as "separating an instance from the rest of the instances". In general, an isolation-based method measures individual instances susceptibility to be isolated and anomalies are those that have the highest susceptibility.

Isolation Forest (iForest), builds an ensemble of iTrees for a given dataset and anomalies are the points that have shorter average path lengths on the iTrees.

Let $\mathcal{X} = \{x_1, \ldots, x_n\}$ be a set of d-dimensional points and $\mathcal{X}' \subset \mathcal{X}$. An iTree is defined as a data structure that:

- for each node $\mathcal{T}$ in the tree, $\mathcal{T}$ is either an external-node with no child, or an internal-node with one "test"and exactly two daughter nodes ($\mathcal{T}_l$ and $\mathcal{T}_r$)
- a test at node $\mathcal{T}$ consists of an attribute q and a split value p such that the test $q < p$ determines the traversal of a data point to either $\mathcal{T}_l$ or $\mathcal{T}_r$.

In order to build an iTree, the algorithm recursively divides $\mathcal{X}'$ by randomly selecting an attribute q and a split value p, until either

- the node has only one instance, or
- all data at the node have the same values.

Hyperparameter	Description
n_estimators	The number of base estimators in the ensemble.
contamination	The amount of contamination of the data set, i.e. the proportion of outliers in the data set.

Table 25: Hyperparameters used on GridSearch of Isolation Forest.

6.3 Model Evaluation and Results Analysis

A separation of the training dataset into training and validation datasets was necessary as a semi-supervised Anomaly Detection approach only uses the normal data for the training of the models. The validation dataset was composed of all the damage data from the train dataset, and 20% of the background information.

The top ten set of hyperparameters resulting from the application of the k-fold nested cross validation with 5 folds to GridSearch were then used to train the model to the training data.

Two metrics were used to choose the best models from each of the algorithms:

- Matthews Correlation Coefficient (MCC) on the validation dataset.
- Number of FPs on the test dataset.

The validation and test datasets MCC was calculated inferring on the whole data, to understand the implication of the use of those hyperparameters on more data.

As said above, the selection of the best models was based on the validation MCC, being the test MCC just used as a confirmation that the model is performing in accordance with the validation results.

The MCC was calculated by comparing the inferred labels from the models and the true label of each of the windows.

Another statistic of evaluation and comparison was the number of FPs obtained in the test dataset. FPs, in this problem, are events that did not result in damage but were classified as damage-causing. Having the lowest number as possible of the these was one of the requisites of the project since they would trigger the system to wrongfully alarm the driver of a vehicle damage.

Algorithm	Validation MCC	Test MCC
OCSVM	**0.63**	0.14
Isolation Forest	0.4	0.16
Gaussian Mixture Model	0.36	0.0
k-Means	0.13	**0.33**
LOF	0.46	0.21
UMAP and OCSVM	0.36	0.0

Table 26: Validation and test MCC

The validation and test MCC of the best models is described on the Table 26. The Gaussian Mixture Model, as well as the joined model with UMAP and OCSVM, did not generalize well to the data, as the test MCC is 0.0, which means a random classification. In terms of the validation MCC, the OCSVM model had the best results, but the test MCC has dropped a lot compared to the validation. The different distribution of abnormal and normal data on both datasets could be the reason for the test MCC to be a lot lower than the validation MCC. In terms of a lower variance between the validation and test MCC, the LOF model has the best results. Considering that the LOF algorithm was created to be applied to Anomaly Detection problems, the results are not unexpected.

Algorithm	True Negative	False Positive	False Negative	True Positive
OCSVM	3250	850	**62**	1077
Isolation Forest	2662	1438	189	950
Gaussian Mixture	2447	1653	232	907
k-Means	4011	**89**	1046	93
LOF	3654	446	500	639
UMAP and OCSVM	3486	614	1098	41

Table 27: Confusion Matrix results on the validation set

When looking at the Confusion Matrix of the validation dataset (Figure 27), the lowest FPs come from the k-Means model. Nonetheless, the trade-off of having a low FPR and a MCC of 0.13 is to have a higher number of False Negatives (FNs).

Algorithm	True Negative	False Positive	False Negative	True Positive
OCSVM	8774	752	179	89
Isolation Forest	5762	3764	**28**	240
Gaussian Mixture	4901	4625	122	146
k-Means	9453	**73**	202	66
LOF	8397	1129	115	153
UMAP and OCSVM	8243	1283	235	33

Table 28: Confusion Matrix results on the test set

The model with the lowest FPs on the test dataset is also the k-Means model (Table 28). From an analysis of the FPs of all the models it was possible to conclude that the events that were worst classified were events related to doors and bottoming out of the vehicle.

Taking into account that the number of damaging windows is 21.7% of the whole validation dataset, and only 3% on the test dataset, the imbalance in distributions could be resulting in a difficulty for the model to have similar MCC results on the validation and testing dataset. The removal of events on the validation dataset would not be an option, as the objective of the project was for there to not be a defined distribution. As the objective of the project is to implement a system for impact detection on vehicles, defining a distribution of impacts that can occur would invalidate its use on a real setting system.

Chapter

7 Conclusion and Future Work

This dissertation was developed as an internship for Bosch Car Multimedia, in Braga, where the main objective was to use Anomaly Detection to classify as a damage or background a series of accelerometer data. Anomaly Detection is a binary technique that allows to detect data that do not conform to expected behavior. The data was presented as pairs H5-jsons, where each H5 file had all the raw sensor data and the corresponding json had information about the label and the conditions of how the data was collected, e.g., weather, type of surface where most of the collection was performed, driver, among others.

The other objective of this thesis was to understand and apply a cycle of a Machine Learning (ML) project, where the methodology used was Cross Industry Standard Process model for the development of Machine Learning applications with Quality assurance methodology (CRISP-ML(Q)).

As described before, the early work developed in Chapter 4 was important for both the objectives of the dissertation. The process of the Data Collection, which was not possible to perform on the SlimScaley project, was crucial to understand how the data is collected and how to assure quality on the collection and the following processing of the data. Another important aspect of the EasyRide project was the fact that it allowed to introduce the problematic of Anomaly Detection.

On the SlimScaley, an internal project of Bosch, most of the work was developed, going from the initial understanding of the business side, until the modeling phase.

The algorithms that allowed for the best results where the Local Outlier Factor (LOF) and One-Class Support Vector Machine (OCSVM). Despite the promising results, considering that Anomaly Detection is a technique that allows for the detection of anomalies, the fact that the models of the Chapter 6 cannot accurately identify all damaging events as anomalies and even identify normal/background data as anomaly can be corroborated by the Exploratory Data Analysis on Chapter 5. Some events involving doors can have a magnitude similar, or even higher, than some events that originate damage. Furthermore, other background events, for instance bottom out, also have a high energy representation on the accelerometer axis. The fact that those events cannot be discriminated as normal could be the result of some factors:

- The features are not explanatory, that is, cannot correctly describe the normal class, resulting in a difficulty of distinguishing the normal from abnormal data.
- The data available was not representative of each event class and/or not in quantities for the application of Anomaly Detection.
- The Anomaly Detection approach cannot be implemented as a solo concept to this problematic.

An example of feature that was considered to initially be explanatory of a high intensity event (damage)

was zero-crossings. However, a weak signal with a high variance can also have a high presence of zero-crossings. A research of the explainability of the features needs to be carried in order to limit ambiguous features.

Anomaly Detection is a technique that allows to identify occurrences of anomalous events. The definition of anomaly takes into consideration that those class which are considered anomalous, are composed of rare events. Anomaly Detection is specially used on unbalanced datasets, where the best results come from extremely low presence of abnormal data. Collecting more data could mitigate the problematic of the distribution of anomalous/normal events and result in a higher performance from the models.

One aspect of Anomaly Detection that is important to emphasis is a specific set of anomalies: novelties. Novelties are data that has never been seen by the model. An approach with Neural Networks, in its original form, can only look at classes of data which have already seen and, if confronted by new data, can only attribute one of the classes. On the other hand, Anomaly Detection has the ability to determinate if new data is normal, that is, was trained as a normal instance, or abnormal, not being limited by the classes of data.

An Anomaly Detection algorithm as a filtering step on the modeling pipeline could potentiate the performance of the whole system. Anomaly Detection integrated with a Sensor Fusion algorithm could be a possible study for future work. The ability of Sensor Fusion to gather information from different sensors could better specify the aspects of damaging and non-damaging data, integrated with the ability of filtering some data with Anomaly Detection, could enable the models for an overall better classification and detection of damage.

If the system were to be integrated with the dashboard of the vehicle, information of the instant velocity and the state of the doors, for instance, would be retrieved. This would allow the system to know when a vehicle was parked or moving, and this information could be used to filter out some events that would not happen when the vehicle was parked and, furthermore, the information of the doors would filter out the necessity of analyzing door events from the Small Damage Detection (SDD) system.

Bibliography

Aggarwal, C. C. (2016). *Outlier analysis* (2nd). Springer. https://doi.org/10.1007/978-3-319-47578-3_1. (Cited on page 4)

Albrecht, J., Ramachandran, S., & Winkler, C. (2020). *Blueprints for text analytics using python*. O'Reilly Media, Inc. (Cited on page 59)

Alonso, J., López, J., Pavón, I., Recuero, M., Asensio, C., Arcas, G., & Bravo, A. (2014). On-board wet road surface identification using tyre/road noise and support vector machines. *Applied Acoustics*, *76*, 407–415. https://doi.org/10.1016/j.apacoust.2013.09.011 (Cited on page 15)

Amin, M., Reaz, M. B. I., Nasir, S., & Bhuiyan, M. (2016). Low cost gps/imu integrated accident detection and location system. *Indian Journal of Science and Technology*, *9*. https://doi.org/10.17485/ijst/2016/v9i10/80221 (Cited on page 38)

Astarita, V., Caruso, M. V., Danieli, G., Festa, D. C., Giofrè, V. P., Iuele, T., & Vaiana, R. (2012). A mobile application for road surface quality control: Uniqualroad. *PROCEDIA: SOCIAL & BEHAVIORAL SCIENCES*, *54*, 1135–1144. https://doi.org/10.1016/j.sbspro.2012.09.828 (Cited on page 12)

Beijing Hike IoT. (2017). *Hk-a5 laser pm2.5/10 sensor*. Retrieved February 16, 2021, from https://dfimg.dfrobot.com/nobody/wiki/1211380111ac0284979e33578da23a37.pdf. (Cited on page 24)

Bellman, R. (1957). *Dynamic programming*. American Association for the Advancement of Science. (Cited on page 59)

Bello Salau, H., Aibinu, A., Onumanyi, A., Onwuka, L., Dukiya, J., & Ohize, H. (2018). New road anomaly detection and characterization algorithm for autonomous vehicles. *Applied Computing and Informatics*, *In Press*, 1–10. https://doi.org/10.1016/j.aci.2018.05.002 (Cited on page 13)

Bello Salau, H., Aibinu, A., Onwuka, L., Dukiya, J., Bima, M., Onumanyi, A., & Folorunso, T. (2015). A new measure for analysing accelerometer data towards developing efficient road defect profiling systems. *Journal of Scientific Research and Reports*, *7*, 108–116. https://doi.org/10.9734/JSRR/2015/16840 (Cited on page 13)

Bosch. (2017, July). *Bme680 - datasheet*. Retrieved February 16, 2021, from https://cdn.sparkfun.com/assets/8/a/1/c/f/BME680-Datasheet.pdf. (Cited on page 25)

Breunig, M. M., Kriegel, H.-P., Ng, R. T., & Sander, J. (2000). Lof: Identifying density-based local outliers. *SIGMOD Rec.*, *29*(2), 93–104. https://doi.org/10.1145/335191.335388 (Cited on page 63)

Briggs, W., Henson, V., & McCormick, S. (2000). *A multigrid tutorial, 2nd edition*. Society for Industrial; Applied Mathematics. (Cited on page 58)

Cawley, G. C., & Talbot, N. L. (2010). On over-fitting in model selection and subsequent selection bias in performance evaluation. *J. Mach. Learn. Res.*, *11*, 2079–2107 (Cited on page 60)

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys*, *41*(3). https://doi.org/10.1145/1541880.1541882 (Cited on pages 5 and 6)

Chen, K., Tan, G., Lu, M., & Wu, J. (2015). Crsm: A practical crowdsourcing-based road surface monitoring system. *Wireless Networks*, *22*. https://doi.org/10.1007/s11276-015-0996-y (Cited on page 15)

Chicco, D., & Jurman, G. (2020). The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC Genomics*, *21*. https://doi.org/doi.org/10.1186/s12864-019-6413-7 (Cited on page 11)

Chowdhury, S. S., Islam, K. M., & Noor, R. (2020). Anomaly detection in unsupervised surveillance setting using ensemble of multimodal data with adversarial defense. (Cited on pages 15 and 16)

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Mach. Learn.*, *20*(3), 273–297. https://doi.org/10.1023/A:1022627411411 (Cited on page 62)

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, *13*(1), 21–27. https://doi.org/10.1109/TIT.1967.1053964 (Cited on page 63)

Deisenroth, M. P., Faisal, A. A., & Ong, C. S. (2020). *Mathematics for machine learning*. Cambridge University Press. (Cited on page 53)

Deming, S., Michotte, Y., Massart, D., Kaufman, L., & Vandeginste, B. (1988). *Chemometrics: A textbook*. (Cited on page 20)

Dogru, N., & Subasi, A. (2012). *Traffic accident detection by using machine learning methods*. http://sedac.ciesin.columbia.edu/es/esi/es%C4%B12005. (Cited on page 14)

Douangphachanh, V., & Oneyama, H. (2014). Formulation of a simple model to estimate road surface roughness condition from android smartphone sensors. *2014 IEEE 9th International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, 1–6. https://doi.org/10.1109/ISSNIP.2014.6827694 (Cited on page 12)

Edgeworth, F. Y. (1887). Xli. on discordant observations. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, *23*(143), 364–375. https://doi.org/10.1080/14786448708628471 (Cited on page 5)

Engelhardt, S., Oksuz, I., Zhu, D., Yuan, Y., Mukhopadhyay, A., Heller, N., Huang, S., Nguyen, H., Sznitman, R., & Xue, Y. (2021). *Deep generative models, and data augmentation, labelling, and imperfections: First workshop, dgm4miccai 2021, and first workshop, dali 2021, held in conjunction with miccai 2021, strasbourg, france, october 1, 2021, proceedings*. Springer International Publishing. https://doi.org/10.1007/978-3-030-88210-5. (Cited on page 20)

Eriksson, J., Girod, L., Hull, B., Newton, R., Madden, S., & Balakrishnan, H. (2008). The pothole patrol: Using a mobile sensor network for road surface monitoring. *MobiSys'08 - Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services*, 29–39. https://doi.org/10.1145/1378600.1378605 (Cited on pages 13 and 14)

Fawcett, T. (2006). Introduction to roc analysis. *Pattern Recognition Letters*, *27*, 861–874. https://doi.org/10.1016/j.patrec.2005.10.010 (Cited on pages 8 and 10)

Fernandes, J. M., & Machado, R. J. (2016). *Requirements in engineering projects*. Springer International Publishing. https://doi.org/10.1007/978-3-319-18597-2_3. (Cited on page 19)

Gao, J., & Tan, P. (2006). Converting output scores from outlier detection algorithms into probability estimates. *Sixth International Conference on Data Mining (ICDM'06)*, 212–221. https://doi.org/10.1109/ICDM.2006.43 (Cited on page 11)

Gherabi, N., & Kacprzyk, J. (2021). *Intelligent systems in big data, semantic web and machine learning* (1st ed.). Springer. https://doi.org/10.1007/978-3-030-72588-4. (Cited on page 19)

Ghezzi, C., Jazayeri, M., & Mandrioli, D. (1991). *Fundamentals of software engineering*. Prentice-Hall, Inc. (Cited on page 19)

Goldstein, M., & Uchida, S. (2016). A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PLOS ONE*, *11*(4), 1–31. https://doi.org/10.1371/journal.pone.0152173 (Cited on page 63)

Graps, A. (1995). An introduction to wavelets. *IEEE Comput. Sci. Eng.*, *2*(2), 50–61. https://doi.org/10.1109/99.388960 (Cited on page 46)

Hamon, R., Junklewitz, H., & Sanchez, I. (2020). Robustness and explainability of artificial intelligence. *Publications Office of the European Union* (Cited on page 18)

Han, J., Kamber, M., & Pei, J. (2012). 1 - introduction. In J. Han, M. Kamber, & J. Pei (Eds.), *Data mining (third edition)* (Third Edition, pp. 1–38). Morgan Kaufmann. https://doi.org/10.1016/B978-0-12-381479-1.00001-0. (Cited on page 18)

Hawkins, D. (1980). *Identification of outliers*. Chapman; Hall. (Cited on page 4)

Hoffmann, H. (2007). Kernel pca for novelty detection. *Pattern Recogn.*, *40*(3), 863–874. https://doi.org/10.1016/j.patcog.2006.07.009 (Cited on page 19)

Jeffrey, A., & Zwillinger, D. (2000). *Table of integrals, series, and products*. Elsevier Science. https://doi.org/10.1016/B978-0-12-294757-5.X5000-4. (Cited on page 56)

Kriegel, H., Kröger, P., Schubert, E., & Zimek, A. (2011). Interpreting and unifying outlier scores. *Proceedings of the 11th SIAM International Conference on Data Mining, SDM 2011*, 13–24. https://doi.org/10.1137/1.9781611972818.2 (Cited on page 11)

Lazzeri, F. (2020). *Overview of time series forecasting*. John Wiley & Sons, Ltd. https://doi.org/10.1002/9781119682394.ch1. (Cited on page 5)

Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation forest. *2008 8th IEEE International Conference on Data Mining*, 413–422. https://doi.org/10.1109/ICDM.2008.17 (Cited on page 65)

Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2012). Isolation-based anomaly detection. *ACM Trans. Knowl. Discov. Data*, *6*(1). https://doi.org/10.1145/2133360.2133363 (Cited on page 65)

Liu, H., & Motoda, H. (2012). *Feature selection for knowledge discovery and data mining* (1st ed.). Springer US. https://doi.org/10.1007/978-1-4615-5689-3. (Cited on page 19)

Lukyanenko, R., Castellanos, A., Parsons, J., Chiarini Tremblay, M., & Storey, V. C. (2019). Using conceptual modeling to support machine learning (C. Cappiello & M. Ruiz, Eds.), 170–181. https://doi.org/10.1007/978-3-030-21297-1_15 (Cited on page 17)

Matthews, B. (1975). Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA) - Protein Structure*, *405*(2), 442–451. https://doi.org/10.1016/0005-2795(75)90109-9 (Cited on page 10)

McInnes, L., Healy, J., & Melville, J. (2020). Umap: Uniform manifold approximation and projection for dimension reduction. (Cited on page 59)

Mehrotra, K. G., Mohan, C. K., & Huang, H. (2017). *Anomaly detection algorithms and principles*. (Cited on pages 8 and 11)

Miljković, D. (2010). Review of novelty detection methods. *The 33rd International Convention MIPRO*, 593–598 (Cited on page 5)

miniDSP. (2018, September 14). *Uma-8-sp user manual*. Retrieved February 16, 2021, from https://www.minidsp.com/images/documents/UMA-8-SP%5C%20User%5C%20manual.pdf. (Cited on page 24)

Mitchell, T. M. (1997). *Machine learning* (1st ed.). McGraw-Hill, Inc. (Cited on page 18)

Morgenthaler, S. (2009). Exploratory data analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, *1*(1), 33–44 (Cited on page 19)

Müller, K., Mika, S., Rätsch, G., Tsuda, K., & Schölkopf, B. (2001). An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, *12*(2), 181–201. https://doi.org/10.1109/72.914517 (Cited on page 62)

Nahid, A., Awal, M., Nandy, A., Alahe, M., Uddin, S., & Alam, S. (2019). Feature extraction and classification of eeg signals for seizure detection. https://doi.org/10.1109/ICREST.2019.8644337 (Cited on page 20)

Pearson, K. (1904). On the theory of contingency and its relation to association and normal correlation (Cited on page 8)

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830 (Cited on pages 59 and 60)

Prado-Romero, M. A., & Gago-Alonso, A. (2016). Detecting contextual collective anomalies at a glance. *23rd International Conference on Pattern Recognition (ICPR)*, 2532–2537. https://doi.org/10.1109/ICPR.2016.7900017 (Cited on page 6)

Qin, T. (2020). *Dual learning* (1st ed.). Springer. https://doi.org/10.1007/978-981-15-8884-6. (Cited on page 19)

Ren, J. S., Wang, W., Wang, J., & Liao, S. (2012). An unsupervised feature learning approach to improve automatic incident detection. *15th International IEEE Conference on Intelligent Transportation Systems*. https://doi.org/10.1109/itsc.2012.6338621 (Cited on page 15)

Reynolds, D. (2009). Gaussian mixture models. In S. Z. Li & A. Jain (Eds.), *Encyclopedia of biometrics* (pp. 659–663). Springer US. 10.1007/978-0-387-73003-5_196. (Cited on page 64)

Sabri, M. (2021). *Data scientist pocket guide: Over 600 concepts, terminologies, and processes of machine learning and deep learning assembled together (english edition)*. BPB Publications. (Cited on page 20)

Sainburg, T., McInnes, L., & Gentner, T. Q. (2021). Parametric umap embeddings for representation and semisupervised learning. *Neural Computation*, *33*(11), 2881–2907 (Cited on page 59)

Salgado, C. M., Azevedo, C., Proença, H., & Vieira, S. M. (2016). Noise versus outliers. *Secondary analysis of electronic health records* (pp. 163–183). Springer International Publishing. 10.1007/978-3-319-43742-2_14. (Cited on pages 4 and 5)

Schlegel, B. (2019). *Off-board car diagnostics based on heterogeneous, highly imbalanced and high-dimensional data using machine learning techniques*. kassel university press. https://doi.org/doi:10.17170/kobra-202008141582. (Cited on page 10)

Schölkopf, B., Williamson, R., Smola, A., Shawe-Taylor, J., & Platt, J. (1999). Support vector method for novelty detection. *NIPS*, *12*, 582–588 (Cited on page 62)

Shafique, U., & Qaiser, H. (2014). A comparative study of data mining process models (kdd, crisp-dm and semma). *International Journal of Innovation and Scientific Research*, *12*(1), 217–222 (Cited on page 17)

Shearer, C. (2000). The crisp-dm model: The new blueprint for data mining. *Journal of data warehousing*, *5*(4), 13–22 (Cited on page 17)

Silva, N., Soares, J., Shah, V., Santos, M. Y., & Rodrigues, H. (2017). Anomaly detection in roads with a data mining approach. *CENTERIS 2017 - International Conference on ENTERprise Information Systems*, *121*, 415–422. https://doi.org/10.1016/j.procs.2017.11.056 (Cited on page 15)

Soares, J., Silva, N., Shah, V., & Rodrigues, H. (2018). A road condition service based on a collaborative mobile sensing approach. *2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. https://doi.org/10.1109/PERCOMW.2018.8480346 (Cited on page 15)

Sola, J., & Sevilla, J. (1997). Importance of input data normalization for the application of neural networks to complex industrial problems. *IEEE Transactions on Nuclear Science*, *44*(3), 1464–1468. https://doi.org/10.1109/23.589532 (Cited on page 20)

Song, X., Wu, M., Jermaine, C., & Ranka, S. (2007). Conditional anomaly detection. *IEEE Transactions on Knowledge and Data Engineering*, *19*(5), 631–645. https://doi.org/10.1109/TKDE.2007.1009 (Cited on page 6)

Studer, S., Bui, T. B., Drescher, C., Hanuschkin, A., Winkler, L., Peters, S., & Müller, K.-R. (2021). Towards crisp-ml(q): A machine learning process model with quality assurance methodology. *Machine Learning and Knowledge Extraction*, *3*(2), 392–413. https://doi.org/10.3390/make3020020 (Cited on pages 18, 19, and 20)

Tang, J., Chen, Z., Fu, A., & Cheung, D. (2007). Capabilities of outlier detection schemes in large datasets, framework and methodologies. *Knowledge and Information Systems*, *11*, 45–84. https://doi.org/10.1007/s10115-005-0233-6 (Cited on page 9)

Tharwat, A. (2020). Classification assessment methods. *Applied Computing and Informatics*. https://doi.org/10.1016/j.aci.2018.08.003 (Cited on page 9)

Theodoridis, S., & Koutroumbas, K. (2009). Chapter 5 - feature selection. In S. Theodoridis & K. Koutroumbas (Eds.), *Pattern recognition (fourth edition)* (Fourth Edition, pp. 261–322). Academic Press. https://doi.org/10.1016/B978-1-59749-272-0.50007-4. (Cited on page 20)

Trebuna, P., Halcinová, J., Fil'o, M., & Markovic, J. (2014). The importance of normalization and standardization in the process of clustering, 381–385. https://doi.org/10.1109/SAMI.2014.6822444 (Cited on page 20)

Tukey, J. W. (1977). *Exploratory data analysis* (Vol. 2). Reading, Mass. (Cited on page 19)

Visengeriyeva, D. L., Kammer, A., Bär, I., Kniesz, A., & Plöd, M. (2021, December 15). *Crisp-ml(q). the ml lifecycle process.* https://ml-ops.org/content/crisp-ml. (Cited on page 21)

Vittorio, A., Rosolino, V., Teresa, I., Vittoria, C. M., Vincenzo, P. G., & Francesco, D. M. (2014). Automated sensing system for monitoring of road surface quality by mobile devices. *PROCEDIA: SOCIAL & BEHAVIORAL SCIENCES*, *111*, 242–251. https://doi.org/10.1016/j.sbspro.2014.01.057 (Cited on page 12)

Wirth, R. (2000). Crisp-dm: Towards a standard process model for data mining. *Proceedings of the Fourth International Conference on the Practical Application of Knowledge Discovery and Data Mining*, 29–39 (Cited on pages 17 and 76)

Wu, C., Wang, Z., Hu, S., Lepine, J., Na, X., Ainalis, D., & Stettler, M. (2020). An automated machine-learning approach for road pothole detection using smartphone sensor data. *Sensors*, *20*. https://doi.org/10.3390/s20195564 (Cited on page 16)

Wu, J. (2012). *Advances in k-means clustering: A data mining thinking*. Springer Berlin Heidelberg. (Cited on page 62)

Yoon, J., Noble, B., & Liu, M. (2007). Surface street traffic estimation, 220–232. https://doi.org/10.1145/1247660.1247686 (Cited on page 13)

Zhao, M., & Chen, J. (2020). A review of methods for detecting point anomalies on numerical dataset. *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, *1*, 559–565. https://doi.org/10.1109/ITNEC48623.2020.9085206 (Cited on page 6)

Zhou, Z., Sun, L., Zhang, Y., Liu, X., & Gong, Q. (2020). MI lifecycle canvas: Designing machine learning-empowered ux with material lifecycle thinking. *Human–Computer Interaction*, *35*(5-6), 362–386 (Cited on page 18)

Zhu, T., Wang, J., & Lv, W. (2009). Outlier mining based automatic incident detection on urban arterial road. *Proceedings of the 6th International Conference on Mobile Technology, Application & Systems*. https://doi.org/10.1145/1710035.1710064 (Cited on page 14)

Annex

I Summary of the phases of the Cross Industry Standard Process for Data Mining (CRISP-DM)

Business Understanding	Data Understanding	Data Preparation	Modeling	Evaluation	Deployment
Determine Business Objectives • Background • Business Objectives • Business Success Criteria ***Assess Situation*** • Inventory of Resources • Requirements, Assumptions and Constraints • Risks and Contingencies • Terminology • Costs and Benefits ***Determine Data Mining Goals*** • Data Mining Goals • Data Mining Success Criteria ***Produce Project Plan*** • Project Plan • Initial Assessment of Tools and Techniques	***Collect Initial Data*** • Initial Data Collection Report ***Describe Data*** • Data Collection Report ***Explore Data*** • Data Exploration Report ***Verify Data Quality*** • Data Quality Report	***Data Set*** • Data Set Description ***Select Data*** • Rationale for Inclusion/ Exclusion ***Clean Data*** • Data Cleaning Report ***Construct Data*** • Derived Attributes • Generated Records ***Integrate Data*** • Merged Data ***Format Data*** • Reformatted Data	***Select Modeling Technique*** • Modeling Technique • Modeling Assumptions ***Generate Test Design*** • Test Design ***Build Model*** • Parameter Settings • Models • Model Description ***Assess Model*** • Model Assessment • Revised Parameter Settings	***Evaluate Results*** • Assessment of Data Mining Results w.r.t. Business Success Criteria • Approved Models ***Review Process*** • Review of Process ***Determine Next Steps*** • List of Possible Actions • Decision	***Plan Deployment*** • Deployment Plan ***Plan Monitoring and Maintenance*** • Monitoring and Maintenance Plan ***Produce Final Report*** • Final Report • Final Presentation ***Review Project*** • Experience Documentation

Table 29: Wirth (2000): Tasks and Outputs of the CRISP-DM Reference Model

Annex

II SlimScaley: Data Collection Planning - Event Description

Event ID	Description
EV01	Door slamming
EV02	Low curbs bump with wheels
EV03	Speed bumps
EV04	Stomping occupant/party people
EV05	Trunk lid open/close
EV06	Hood open/close
EV07	Hood slamming
EV08	Sunroof open/close
EV09	Switch pushed in mirror panel
EV10	Roofline slapping
EV11	Sunvisor open/close
EV12	Sunvisor detach/fix
EV13	Make up mirror o/c
EV14	Wiper flapping
EV15	Rear mirror (int.) adjusting
EV16	Side mirror folding
EV17	Side mirror collision
EV18	Object placed on roof
EV19	Wiper activation/deactivation
EV20	Side window opening/closing
EV21	Door slamming engine off
EV22	Speed bumps + ABS braking
EV23	Windshield slapping (mosquito...)
EV24	Smartphone/Navi holder fixation
EV25	Object sliding against windshield
EV26	Switch dipping mirror
EV27	Ventilation maximum
EV28	ABS-braking event
EV29	ESP-intervention in curves

EV30	Rough road
EV31	Belgisch block
EV32	Aquaplaning track
EV33	Engine load change acc
EV34	Engine load change dec
EV35	Carwash
EV36	High-pressure washer
EV37	Wiper on frozen windshield
EV38	Engine on (auto start/stop)
EV39	Person knocking on the vehicle structure

Table 30: Non damaging events

Event ID	Description
EV42	Scratching with object across vehicle
EV43	Bump collision (get bumped by another vehicle)
EV44	Side collision with another vehicle (left / right)
EV45	Hitting the car with object (baseball bat / hammer)
EV46	Throw object at the car

Table 31: Stationary damaging events

Event ID	Description
EV47	Speeding over speed bump / pothole
EV48	Vehicle hits object
EV49	Vehicle side drags on object
EV50	Vehicle drives over obstacle
EV51	Scrape with object when cornering
EV52	Vehicle bumps with rim on low curb

Table 32: Damaging events in movement