Universidade do Minho
Escola de Engenharia

José Gabriel Correia Neves Parpot

Sensor fusion for impact detection in vehicles

April, 2023

Universidade do Minho
Escola de Engenharia

José Gabriel Correia Neves Parpot

Sensor fusion for impact detection in vehicles

Master Thesis
Master in Masters in Informatics Engineering

Work developed under the supervision of:
André Leite Ferreira
João Miguel Lobo Fernandes

April, 2023

This document was created with the (pdf/Xe/Lua)LaTeX processor and the NOVAthesis template (v6.7.0) [1].

Acknowledgements

First i would like to leave my thanks to all the people who either directly or not influenced and aided me with the work of this thesis.

To my dear parents as without them i would not come as far as i am now regarding my academics, professional life but most importantly the person i am.

To my advisors André Ferreira and João Lobo for the opportunity of making this internship along side them and all the help and knowledge shared. Ao meu orientador André Ferreira, pela oportunidade de poder realizar o estagio e de toda a ajuda que foi me prestada.

To Bosch car multimedia, for the opportunity to have my internship in such a rich environment fill with awesome people to work with and allowing me to expand my boundaries with my skills and knowledge in the fields i touched.

To Bruno Silva, for their time to follow the work at a more scientific level throughout the thesis and all the help he provided during the course of it.

To Diana Silva, for all the support and joy she gave me for this last few years that allowed me to work this hole time with a positive mind and filled with joy.

To my brother for all the conversation that we spent together fueling my passion for the dreams we share

To all my friends, Diogo Dias, Mendes, Carlos, Zé Pedro, Rui and much more that i want to put here for enduring me and still giving me continuous support in the difficulties of life.

Obrigado por tudo!

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.

____________________, ______________________________

(Place) (Date)

____

(José Gabriel Correia Neves Parpot)

Abstract

Sensor fusion for impact detection in vehicles

With the advance of technology surrounding the automobile industry, we are starting to see a shift in the need a personal vehicle, opting more often for other options like rental cars and car-sharing services.

With this shift these services face more problems and more specific damage to the vehicles in the fleet. In order to help these services keep track of their fleet state and to help detect impacts if they happen, a multi-sensor fusion for impact detection in vehicles is proposed.

The main focus of this thesis is to implement a multi-sensor fusion approach to detect impacts in vehicles. A comparative study of the previously implemented solution is carried out to help develop and implement the suggested approach. One of the sub-objectives of this work is to find which of the two implemented fusion methods better improves the system performance.

The sensors that compose the detection strutcure, are a Inertial Measurement Unit (IMU) and a microphone, which are located inside the vehicle in different positions. Note that the structure used differs for each dataset.

The fusion works by combining the information of all the accelerometers placed in the vehicle. Two sensor fusion methods applied to the two datasets in this thesis are as follows: a complementary filter which is part of the data fusion level and the second consists in a feature fusion approach by fusion features from vairous sensor combinations.

Keywords: Machine learning, Sensor fusion, Impact detection, Data fusion, Feature fusion.

Resumo

Fusão sensorial para deteção de impact em veículos

Com o avanço da tecnologia no ramo da industria automóvel, estamos a começar a ver uma mudança na necessidade de compra e posse de um veiculo pessoal, optando por outras opções como serviços de aluguer de carros e *car-sharing*.

Com esta mudança, é esperado que estes serviços encontrem mais problemas, mais em especifico, danos nos veículos presentes na frota. De maneira a facilitar o acompanho do estado das suas frotas e a ajudar na deteção de impactos nos veículos quando acontece, é proposto um sistema de fusão sensurial com o objetivo de detetar impactos em veículos.

O maior foco desta tese é implementar um sistema de fusão sensorial para detetar impactos em veículos. Um estudo comparativo da solução anteriormente implementada vai ser realizado para ajudar a desenvolver e implementar a nova proposta. Um dos sub objetivos deste trabalho é encontrar qual dos dois metodos de fusão consegue melhorar o desempenho geral do sistema.

Os sensores que constituem o sistema de deteção são uma *IMU* e um microfone, que em sua vez são instalados no veiculo em diferentes posições. Neste caso a fusão ocorre ao combinar a informação proveniente de todos os acelerometros dentro do veiculo.

Dois métodos de fusão sensorial vão ser aplicados para fundir a informação medida de cada sensor. Isto em troca nos vai permitir uma melhor e mais robusta compreensão do ambiente em que se insere. Uma pesquisa detalhada sobre fusão sensorial foi realizada para completar este objetivo.

Palavras-chave: *Machine learning*, Fusão sensorial, deteção de impactos, fusão de data, fusão de características.

Contents

List of Figures

List of Tables

Acronyms

ANN Artificial neural network 15, 19, 21
AR Augmented Reality 19

EEG Electroencephalography 19, 21
EKF Extended Kalman Filter 20, 21

FN False Negative 39
FP False Positive 39, 43

HDF5 Hierachical Data Format 25

IMU Inertial Measurement Unit v, vi, 1, 2, 19, 20, 21
IoT Internet-of-Things 20

JDL Joint Directors of Laboratories 3, 5, 6, 7
JSON JavaScript Object Notation 25

KNN k-Nearest Neighbor 14, 19, 21

MCC Matthews Correlation Coefficient 39
ML Machine Learning 19, 30

SVM Support Vector Machine 14, 19, 21

TN True Negative 39

TP True Positive 39, 43

VR Virtual Reality 19

WSN Wireless Sensor Networks 20

1

Introduction

1.1 Context

With the accelerated growth of population density in cities we are progressing into a future where possessing a vehicle or multiple vehicles is not essential anymore. If for any reason a vehicle is needed, there is a plectra of choices for the user to choose from, such as for instance ride halling (Uber or Bolt), car-sharing (Share Now), and rental car services (Rent-A-Car). These solutions are increasingly popular due to the possibility to travel anywhere in the city without the downsides of owning a vehicle, like maintenance, taxes and insurance.

As more people adopt this kind of services it becomes harder to reliably check every vehicle for damages. These damages can be the result from collision with other objects, such as other vehicles or any kind of structure. Rental companies and car-sharing services can't reliably perform inspections to all the vehicles due to number in each fleet, and the inspections that are performed can miss detecting small damages on the vehicles. These can lead to major costs for these companies, for instance if a car is damaged it will need repairs and be taken out of service, aside from that if a vehicle is visibly damage people won't rent it as much, which leads to more losses.

To help these companies lower their costs and keep a track of the state of their fleet, one can implement various solutions, for example a possible solution can be collision detection to warn the user of a possible impact. The solution that is the focus on this thesis is a impact detection solution. This solution can lead to many benefits, like helping to detect damage that the inspection may miss.

It's now possible to gather information needed from a impact with the help of sensors in a vehicle. One example of these sensors is the combination of a gyroscope and accelerometer, IMU and for the audio produced by the impact a microphone. With the data capture from this sensors we can fuse then to have a better understanding of the event the car was in and in turn produce better predictions.

1.2 Objectives and expected results

The main goal of this thesis is to implement a multi sensor fusion approach for impact damage detection. The data needed to detect these events is collected through multiple devices, each one equipped with a IMU and a microphone for the audio.

The first objective set for this thesis is to investigate and evaluate the benefits of implementing a multi sensor system instead of a single sensor system. To accomplish this, sensor fusion techniques is applied to the data from two or more sensors and the accuracy compared with the accuracy of a single sensor.

Although we intend to develop and deploy a new multi sensor fusion we need to keep in mind the cost of using a large number of sensors. For this reason the second objective is laid out. Here we intend to find the minimum number of sensors while having an acceptable overall performance.

As for expected results, it's expected that the use of multiple sensors surpasses the performance of a single sensor in vehicle impact detection. It's also expected that the new multi sensor system also surpasses the already implemented system in terms of overall performance while trying to keep the costs to a minimum.

1.3 Document structure

The remaining material in this document is divided into the following major sections:

- Chapter 2 describes the state of art reviewed in the area of sensor fusion. The chapter starts with a introduction on what is sensor fusion, followed by a sensor fusion taxonomy. After that, the main methods for each of the fusion levels, data level, feature level and decision level are presented. The chapters ends with some examples of implementation of various sensor fusion systems in a real context.

- Chapter 3 is composed of two major sections detailing all the progress done in this thesis. The first section details all the necessary steps to implement the proposed solution. The last section of this chapter is reserved for the results analysis regarding the two different fusion methods implemented to better understand the performance gain for each of them.

- As the final chapter of this thesis, Conclusion and Future work gives an overview of the project by summarizing the important steps and conclusions previously mention in the 3. One can also find possible solutions that can be used to replaced some of the methods used or help improve them upon.

2

State of the Art

2.1 Sensor fusion

The first appearance of data fusion in the literature was in the 1960s. It was implemented by the US for the field of robotics and defence. The first definition was in 1980 as the US department of defence established the Joint Directors of Laboratories (JDL) to address the main issues of data fusion and attempt to unify the terminology and procedures of this new field. Since then we can find sensor or data fusion in wide range of field as some examples in robotics, military applications, traffic control, medicine, etc. [2]

Many definitions of sensor fusion or data fusion have been proposed along the years and exist in the literature. One of the more accepted definitions by the community is the definition created by JDL that defines data fusion as process of dealing with association, correlation and combination of data or information, measured from one or more sources of data, in order to achieve a more refined position and complete and timely assessments of situations and events as well as their significance.[3]

Sensor fusion allows to integrate extracted information form several sources into a single signal or information. In many applications, the sources of information or data can be sensor or other devices that allow the understanding or measurement of the changing environment that it is presented. When data is collected or perceived from the sensors, "sensor fusion"or "data fusion"algorithms are used to process the data. [4]

Due to the advance of technology in the recent years, multisensor data fusion has received more attention both in the original are it was defined, military applications as nonmilitary applications. Data fusion uses techniques to combine data from multiple sensors, and related information from databases, in order to improve accuracies and inferences than otherwise could be achieved by a single sensor. Just as mention before, as the technology progresses we can see more and better sensor that make real-time fusion of data increasingly possible. [5]

2.1.1 Problems of multisensor data fusion

The majority of these issues arise from the data to be fused, imperfection and diversity of the sensor technologies, and the nature of the application environment. [6]

- **Data imperfection**: data provided by sensors is always affected by some level of impreciseness as well as uncertainty in the measurements.
- **Outliers**: the uncertainties in sensors arise not only from the impreciseness and noise in the measurements, but are also caused by the ambiguities and inconsistencies present in the environment.
- **Conflicting data**: it can be problematic to fuse this data especially if the fusion system is based on evidential belief reasoning and Dempster's rule of combination
- **Data modality**: sensor networks can collect the data similar (homogeneous) or different (heterogeneous) such as visual, audio, etc.
- **Data correlation**: a common and important issue in distributed fusion settings. Some sensor nodes are likely to be exposed to the same external noise biasing their measurements. This may influence the confidence of the results from the fusion algorithm.
- **Data alignment**: before the fusion occurs the data from the sensors must be transformed from each sensor's local frame into a commons frame.
- **Data association**: there are two different forms of data association: measurement-to-track association, which refers to the problem of identifying from which target, if any, each measurement is originated. The other one being track-to-track association, deals with distinguishing and combining tracks.
- **Operational timing**: The environment the sensors are covering can be composed of different aspects varying in different rates. Also, in case of homogeneous sensors, the frequency of the sensors may be different. A fusion method should incorporate multiple time scales in order to deal with such timing variations in data.
- **Static vs dynamic phenomena**: the phenomenon under observation may be time-invariant or varying with time.

2.1.2 JDL Basic Model

JDL model is the most popular conceptual model in the data fusion community, originated from the US Joint Directors of Laboratories. It was proposed in 1985 by the Department of Defense. The model consists of five processing levels, an associated database, and an information bus that connects the five components. [6]

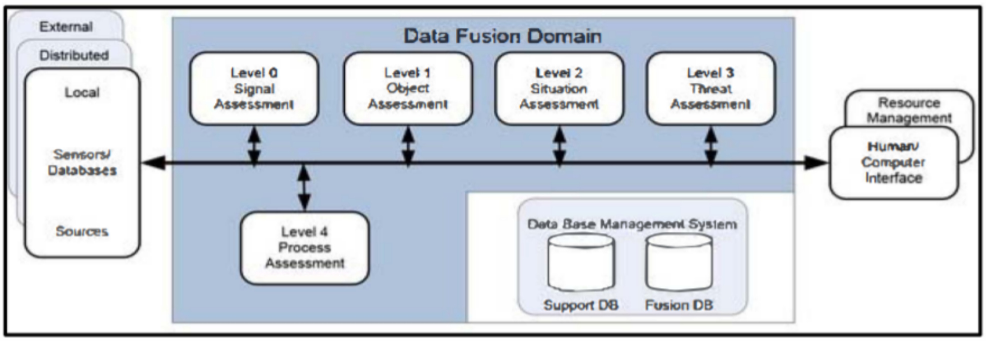


Figure 1: Basic Model of JDL adapted from [7].

The components that are a part of the JDL model are as follows [5]:

- **Sources**: the sources are in charge of providing the input data. Different types of sources can be implemented, like sensors, databases and human inputs.

- **Human/computer interaction**: consists of a interface that allows inputs from the operators and produces outputs to the operators. A way for the user to interact with the system.

- **Database management system**: stores the provided information and the fused results. This system is critical for the hole process due to the large amount of diverse information being stored.

The main core of the Data Fusion Domain is composed of:

- **Level 0 processing (Pre-Processing)**: performs pre-screening of data and then allocates it to the appropriate process. This is intended to reduce the processing load of the fusion processes.

- **Level 1 processing (Object Refinement)**: This level performs what is commonly known as data level fusion, where raw data is fused to get a better understanding of the environment. To achieve this data fusion methods are applied.

- **Level 2 processing (Situation Refinement)**: allows creating a dynamic expression of relationships between entities and events. This level can also be known as feature fusion where feature fusion methods are used.[8]

- **Level 3 processing (Threat Refinement)**: This level relates the current situations into the future and draws conclusions about them. Based on a priori knowledge this level tries to draw inferences about opportunities for operation. [8]

- **Level 4 processing (Process Refinement)**: This level is a process to assess and improve the performance of real-time systems. In order to get the best performance, this levels monitors the system and reallocates sensor and sources. [7]

Despite its popularity, the JDL model has many shortcomings, such as being too restrictive and especially tuned to military applications. The JDL formalization is focused on data (input/output) rather than processing.

2.1.3 Waterfall Model

Another model that's commonly found in data fusion literature is the waterfall model. This models differs from the JDL in the way it's structure, it emphasizes on the processing functions on the lower levels. Apart from that, the sensing and signal processing levels correspond to JDL level 0. The feature extraction and pattern processing levels correspond to the second level of the JDL model. After that, the decision making level corresponds to the third level of the JDL. [9]

Because of the similarities these models has with the JDL model, it suffers from the same drawbacks. It has been used in the defense data fusion community in Great Britain.[6]

The water fall model is composed of three levels as follows:

- **Signal and sensing processing level** - In this level, raw data is transformed to achieve the required information about the surrounding environment. In order to achieve this, data fusion methods are applied.

- **Feature extraction and pattern processing level** - In this level, feature extraction and fusion of those features happens. To achieve this feature fusion methods are used. This level aims to minimise the content of data whilst maximising the information it delivers.

- **Deciding making level** - The final level, relates objects to events. Here we fuse decisions to have a more complete or precise data. In order to achieve this, we apply decision fusion methods, which in turn will output the final decisions of our system.

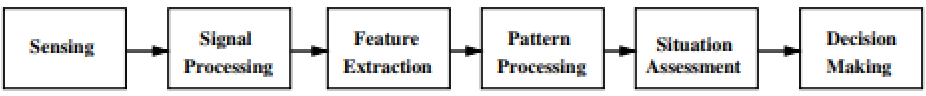


Figure 2: Waterfall model.

2.1.4 Boyd Model

Although this model is a classic decision-support mechanism in military information operations, it has also been used for sensor fusion.[6] The model is composed of 4 stages as follows:

- **Observe:** We can compare this level as the level 0 of the JDL model. Here raw data is fused to get a more robust measurement rather than using from a single sensor.
- **Orientate:** In this level we can correspond to levels 1,2 and 3 of the JDL model. As a result of that, feature and decision fusion techniques can be found to achieve a better understanding of the environment.
- **Decide:** This stage corresponds to the last level of the JDL model (Process refinement). Where we allocate the sensors and resources depending of the system needs.
- **Act:** This level cant be compared to the JDL model since the model doesn't close like the Boyd model. The only model that explicitly closes the loop by taking account of the effect of decisions in the real world.

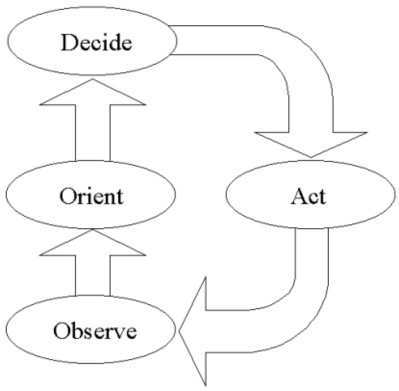


Figure 3: Boyd loop.

2.1.5 Data fusion architectures

1. **Centralized Architecture**: Data fusion is performed using sequential estimate techniques. The process of this method is performed in a single node as follows: data is transferred to a central processing device via communication networks or other mechanisms. [10]
2. **Decentralized Architecture**: Here there's no need for a central node, since the data is processed in each sensor node. Because of this type of architecture, it's commonly used in applications that have a large number of sensor. [10]

3. **Hybrid Architecture**: Represents the combination of the merger in data fusion and vector state. The state vector fusion is done to reduce the computational workload and communication demand. the advantage of the hybrid architecture lies in flexibility. [7]

2.1.6 Classification based on the relations between data sources

Sensor fusion can be categorized according to the type of relations between the configuration. Durrant-Whyte distinguishes three types of relations as indicated in figure 3;

1. **Complementary**: when the information provided by the input sources represents different parts of the scene and could thus be used to obtain more complete global information. This helps fix the problem of incompleteness of sensor data. In general its easy and quick to fuse complementary data from independent sensors. [6]

2. **Redundant**: when two or more input sources provide information about the same target and could thus be fused to increment the confidence. An example of this kind of configuration is to combine two different measurements from the same sensor at different time intervals.

3. **Cooperative**: when the provided information is combined into new information that is typically more complex than the original information. An example of cooperative classification, is by combining two two-dimensional images into a three-dimensional image. In contrast to complementary, cooperative sensor fusion is hard to design and implement. [6]

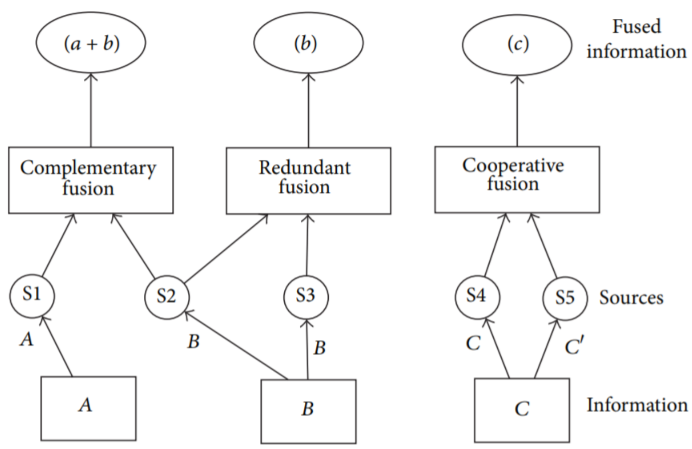


Figure 4: Whyte's classification based on the relations between the data sources.

Although there are three configuration to distinguish the types of relations between sensor, that doesn't mean we need to use only one configuration. Some applications use more than one of the three configurations in order to achieve a more robust and accurate data.

2.1.7 Classification based on input and output

1. **Data in-data out (DAI-DAO)**: the most basic or elementary data fusion method that is considered in classification. This type of data fusion process inputs and outputs raw data, the results are typically reliable or accurate. Data fusion at this level is conducted immediately after the data are gathered from the sensors. It is commonly referred as data fusion, or low level fusion. An example at this level would be using an average filter to combine two signals streams from different microphones into a more accurate and robust stream.[11]

2. **Data in-feature out (DAI-FEO)**: this is the next step in this five step hierarchy. At this level, the data fusion process employs raw data from the sources to extract features or characteristics that describe an entity in the environment. Fusion here processes inputs or output-fusion into features. It can be considered as feature fusion or data fusion. [12]

3. **Feature in-feature out (FEI-FEO)**: both the input and output of the data fusion process are features. Thus, the data fusion process addresses a set of features with to improve, refine or obtain new features. This process is also known as feature fusion. Techniques applied at this level are in general feature extraction and after that feature selection processes. [12]

4. **Feature in-decision out (FEI-DEO)**: this level obtains a set of features as input and provides a set of decisions as output. Most of the classification systems that perform a decision based on a sensor's inputs fall into this category of classification. This level is commonly reference as feature fusion.

5. **Decision in-decision (DEI-DEO)**: this type of classification is also known as decision fusion. It fuses input decisions to obtain better or new decisions. Although its not always necessary or the most advisable approach to use this classification, due to the need to have compatible sensors to permit the previous fusion, it's a feasible approach. This level is also known as decision fusion.

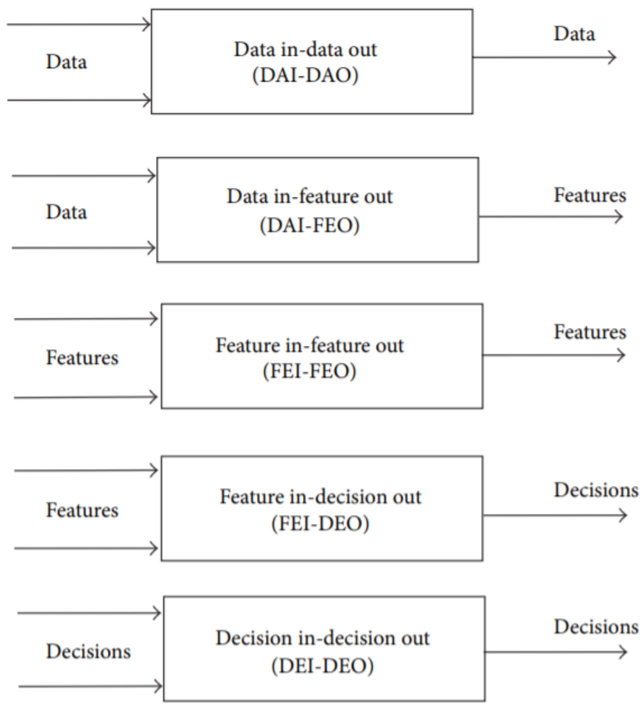


Figure 5: Dasarathy's classification, based on [12].

The levels of classification presented aren't mutually exclusive. It's encourage to use then in different stages of fusion and find the best solution for the problem we want to address.[11]

2.1.8 Classification based on level of abstraction

1. **Low level fusion**: the raw data are directly provided as an input to the data fusion process, which provide more accurate data than the the individual sources. [6]

2. **Medium level fusion**: characteristics or features are fused to obtain features that could give a better representation of the information. This level is also known as the feature or characteristic level.

3. **High level fusion**: also known as decision fusion, takes symbolic representations as sources and combines them to obtain a more accurate decision.

4. **Multiple level fusion**: this level addresses data provided from different levels of abstraction.

2.1.9 Advantages of using multi sensor vs single sensor

1. **Enhanced signal to noise ratio** - the merging of various streams of sensor data decreases the influences of noise.

2. **Diminished ambiguity and uncertainty** - the use of data from different sources reduces the ambiguity of output.

3. **More reliable** - the data provided from a single sensor are usually more unreliable.

4. **Robustness** - the use of several similar sensors provides redundancy, which in the end raises the fault tolerance in the case of a sensor failure.

5. **Improved precision/resolution** - when measuring the same attribute from different sensors are merged, the granularity of the resulting value from the merge is finer than in the case of a single sensor.

2.1.10 Data fusion

Data fusion or signal level fusion is the lowest level of fusion. Its targeted to combine raw data from sensors of the same type to produce a new raw data, which in turn gives a more robust and informative measurement than from a single sensor. Because the need to have measurements from commensurate sensors in order to perform the fusion process, it becomes difficult to use in every applications and it's preferably to use feature or decision fusion or a combination of the three .

2.1.10.1 Data Fusion Methods

1. **Kalman Filter** Kalman filter as been used as the base for many sensor fusion algorithms. By gathering noisy measurements made to the system we can estimate it's state. The first step is to define the initial value of the system state. We can accomplish this by using other methods or by measuring directly the system. We also need to get the prediction error associated to the prediction. To measure the prediction error the Kalman filter uses a correlation between the prediction and the the actually real value.[13, 14] Let X be the system state and Y be the measurements made to the system.

$$\hat{X}_0 = X(0), \tag{2.1}$$

$$P_0 = P(0), \tag{2.2}$$

Here, $\hat{X}_k$ represents the predicted state the system is in and P_k represents the error covariance matrix of the time step k. For each prediction step we make, the state and error covariance of the state are determined based on the dynamics of the system. Let A_k be the dynamics of the system, Q_k be the convariance matrix of motion error, $\hat{X}_{k,k-1}$ and $P_{k,k-1}$ be the state prediction with the associated error covariance matrix, respectively.[14]

$$\hat{X}_{k.k-1} = A_k \hat{X}_{k-1}, \tag{2.3}$$

$$P_{k,k-1} = A_k P_{k-1} A_k^T + Q_k \tag{2.4}$$

After a new set of measurements are acquired from the system, the state of the system along side the associated error covariance are adjusted.

Although it's a common fusion method used in many applications, it was originally designed for linear systems. If the system changes to nonlinear, it will be require changes to fit the new system. Another shortcoming is the high computational need that comes with it, leaving it undesirable in some cases, especially if we are considering implementing a real-time application. [14]

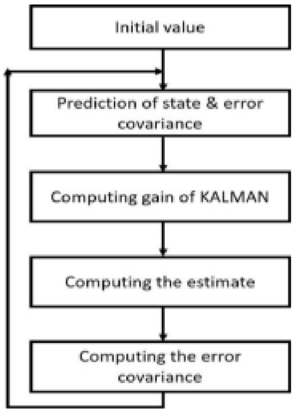


Figure 6: Kalman filter flow chart adapted from [15]

2. **Weighted Average** One of the more simplest methods of data fusion. It consists in taking an average of all the sensor measurements and combining the information in order to create a more robust measurement of the environment. As some sensors can have worst reading than others it wouldn't be wise to consider every sensor as the same. For this reason a weight is assigned to each stream of data, which depending on the value will have more or less relevancy to the final estimate. Theses weights can be manually assigned or determined from other factors.[16, 17]

   Let x be the estimate we want to calculate, n the number of fused stream data and w the weight associated to the streams.

$$x_{fused} = \sum_{i=0}^{n} w_i x_i \tag{2.5}$$

3. **Complementary filter** The complementary filter is usually used to remove the noise of a measurement by fusing the data from two or more sensors. Some applications prefer to use this filter instead of the Kalman filter, because of the low computational, processing power and the accuracy improvements it provides. [18]

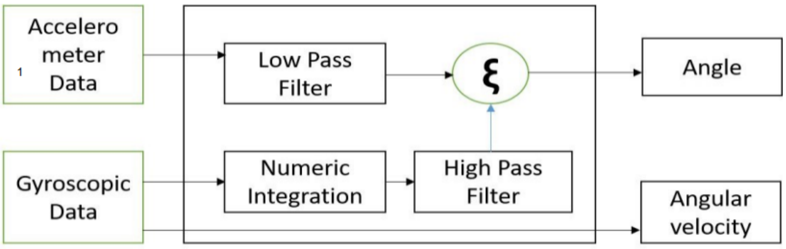


Figure 7: Complementary filter Block Diagram

The complementary filter is composed of a low and high pass filter, which after the two pass filters are applied to the data, the output is fused to get a better measurement than it would be possible using a single sensor.[15]

The purpose of a low pass filter is to attenuate signals with a frequencies higher than a selected cutoff. This type of filters is used to fix jitters in measurements for example coming from accelerometer. In contrast the high pass filter is used to passe signals that are above a selected cutoff frequency and attenuates the signals that are lower than that cutoff. It's usually found to fix drifts in measurements for example in gyroscopes that tend to drift in time.

2.1.11 Feature fusion

Feature sets extracted from multiple data sources can be fused to create a new high-dimensional feature vector, we can achieve this by concatenating two or more different feature vectors into one vector.[19] At this level machine learning and patter recognition algorithms are applied to vectors with characteristics that later on can be combined to form joint characteristics joint vectors from which the classification is carried out.

2.1.11.1 Feature Fusion Methods

1. **Instance-Based Learning** After we define and build the feature set, we can use pattern classification techniques to fuse the data from the acquired sensor into more relevant events. The method Instance-based learning is a learning technique, which after we encountered a new sample we can make the decision on how to generalise beyond the training data. We can make the decision by using a nearest neighbour classifier to separate the unlabelled observation and measure its distance from the different labelled samples in the feature set.

Let u be an unlabelled observation, v be the nearest neighbour and $c(v_i)$ is the assigned label of v_i. We can classify u as $c(v_i)$ if

$$d(u, v_i) = \min_{1 \leq j \leq N} d(u, v_i) \tag{2.6}$$

We can measure the level of thrust in the assigned label by using the distance between the test sample and the selected reference points and by using the different occurrences for each class in the selected set. When we select the set of reference points based on a premeditated value k, we use the learning technique known as k-Nearest Neighbor (KNN). [20, 21]

2. **Support Vector Machine** One of the most widely used algorithms for classification problems is Support Vector Machine (SVM). The SVM tries to evaluate a linear hyperplane between two classes, although it can theoretically have an infinite number of hyperplanes. The main purpose of the SVM is to not only to classify with a high level of certainty the various sample points, but also to maximize the minimum distance between the optimally divided hyperplane and all training sample points.[22]

   For the classification to happen we need to give a training dataset with the training samples and a label associated with each sample. Let x_i be the features samples, and $y_i \in$ {-1,+1} the target label, where $y = +1$ corresponds to the class C_1 and $y = -1$ corresponds to the class C_2, of the training dataset represented by $\{x_i, y_i\}, i = 1, 2, ..., N$, where N represents the number of samples in the training dataset. If the training dataset is to be separate in a linear fasion, the hyperplane equation is $w \cdot x + b = 0$. Therefore the sample (x_i, y_i) needs to satisfy:

$$y_i[(w \cdot x_i) + b)] \geq 1, i = 1, 2, ..., N \tag{2.7}$$

   Where w is the plane normal vector and b represents the constant term. Support vectors are sample points that are close to the hyperplane and have an influence on the position and the orientation of the hyperplane. We can maximize the margin of the classifier using this support vectors.

3. **Artificial neural network** The fused output is a combination of input signal and corresponding weights. Several fusion methodologies are used and depending on the input and outputs required the stages in the model can perform either signal, feature as decision level fusion. Despite that, it's more commonly found applied at the feature level fusion. [23]

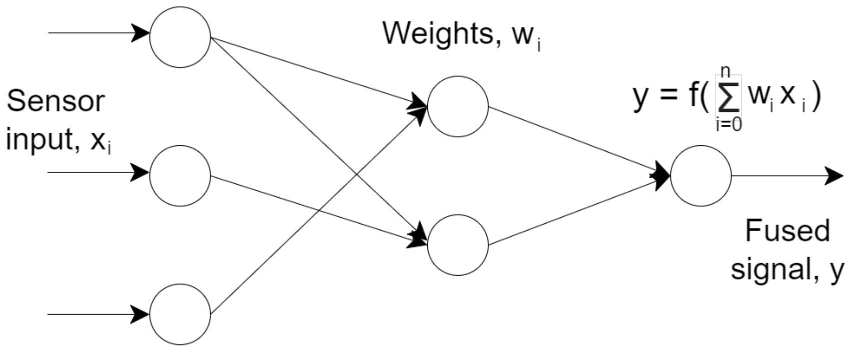


Figure 8: Neural network structure for sensor fusion.

Artificial neural network (ANN) were developed following the way the human brain functions. Therefore, ANN are composed of simple processors nodes which are called "neurons"and are linked by connections with a weight associated to them.

The ANN architecture is composed of a input layer, which receives the input values from external sources. Following the input layer we have one or more hidden layers, that consists in a set of neurons connected to the input layer and output layer or if there's multiple hidden layers, connected to another hidden layer. Finally there's the output layer, which gives us the output of the network. This output can be a single neuron that ranges between 0 and 1 or multiple output neurons.[24]

The way we make the neural network to learn can vary according with the objective we want to achieve. The most common learning method is supervised learning were we compare the output of the network with the desired output and depending on the result we adjust the connection weights of each connection to provides us with a more accurate output.[25]

2.1.12 Decision fusion

Process of selecting a class hypothesis or decision from the set of local hypotheses generated by individual sensors. A decision is taken by the knowledge, provided by several sources of the perceived situation. Decision-level fusion output is a unique decision obtained from local decision of multiple(homogeneous or heterogeneous) sensors, therefore it utilizes the information that has been already abstracted to a certain level through preliminary senor data-or feature-level processing such that high-level decision can be made.[26]

2.1.12.1 Decision fusion Methods

1. **Bayesian Methods** - Its build upon the Bayes theorem. Information fusion based on the Bayesian inference provides a formalism for combining evidence according to the probability theory rules. Using the conditional probability terms that describe beliefs and attributes values in the interval [0,1], where zero indicates a complete lack of belief and one indicates an absolute belief.[27]

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)} \tag{2.8}$$

   $P(Y|X)$ represents the belief in Y given the information of X. This probability is obtained by multiplying the probability of the hypothesis $P(Y)$ by having the probability of X given that Y is true.

   The main disadvantages of the Bayesian methods are the following:

   - Difficulty in obtaining the hypothesis value of the previous probabilities.
   - The complexity of the method exponentially increases when there are multiple potential hypotheses, along with a substantial number of events that depends on the conditions.
   - In order to use the Bayes theorem the hypothesis need to be mutually exclusive.

2. **Demspter-Shafer Theory** - The Dempster-Shavfer theory is a mathematical theory that generalizes the Bayesian theory. The difference between Dempster-Shafer theory and the previous one is the frame of discernment. This is defined by Θ which represents all the possible states that define the system.

   Due to the system being only in one state, Θ is mutually exclusive. It's called frame of discernment, because the elements in Θ are applied to discern the current state of the system.[27–29]

   We also have 2^{Θ} which represents the hypotheses. A probability is attributed to each hypothesis $H \in 2^{\Theta}$, based on the evidence E, according to the mass function $m : 2^{\Theta} \to [0, 1]$, which satisfies:

$$m(\emptyset) = 0. \tag{2.9}$$

The sum of the all mass function of 2^Θ is one. The Dempster-Shafer theory defines the belief function bel : $2^\Theta \longrightarrow [0,1]$ over Θ to express incomplete beliefs in a H.[28]

$$bel(H) = \sum_{A \subseteq H} m(A). \tag{2.10}$$

To determine the plausibility of each hypothesis, the function $pl : 2^\Theta \longrightarrow [0,1]$ over Θ is defined.

$$pl(H) = 1 - dou(H) = \sum_{A \cap H = \emptyset} m(A). \tag{2.11}$$

The Dempster-Shafer inference, contrary to the Bayesian inference, does not require a priori probabilities, because at the instant that the information is provided the probability is assigned. It can also be used to represent incomplete Knowledge and updating beliefs.

3. **Fuzzy logic** - In contrary to other logical system, fuzzy logic is used in environment of uncertainty and imprecision. The method provides a way of representing incomplete or imprecise data. In a nutshell, in fuzzy logic everything, including truth, is a matter of degree [30, 31].

   - **Fuzzy Sets** - Its used to handle the concept of uncertainty and partial truth, which enables the modeling of natural language. Fuzzy sets, together with fuzzy reasoning systems, are given the tools to write software, which enables computing systems to understand vague terms, and to reason with these terms.[32]

     The main purpose for fuzzy sets lies in the capability to structure ambiguous and imprecise data.

   - **Definitions** - Let X be the domain of observable universe, and $x \in X$ be a specific element of the domain X, and the fuzzy set A is specify by a membership mapping function:

$$\mu_A(x) : X \longrightarrow [0,1] \tag{2.12}$$

     Therefore, $\mu_A(x)$ represents the level of truth or certainty that the element x belongs to fuzzy set *A*.

   - **Membership functions** - is used to attribute each elements of the domain X of the corresponding fuzzy set A. The membership functions can take any shape over the domain which the fuzzy sets are established, and need to satisfy two major constraints:

     - Any membership function has a range of [0 1].
     - For each $x \in X, \mu_A(x)$ must be unique.

   - **Fuzzy Operators** - In contrary to bivalent logic, fuzzy logic doesn't have a finite set of possibilities for each input, which requires the operator to be conveyed as functions for all the probable fuzzy values. Let A and B be fuzzy sets in the domain X.

– **Operator AND**

$$\mu_{A \cap B}(x) = min(\mu_A(x), \mu_B(x)) \quad \forall x \in X \tag{2.13}$$

– **Operator OR**

$$\mu_{A \cup B}(x) = max(\mu_A(x), \mu_B(x)) \quad \forall x \in X \tag{2.14}$$

– **Operator NOT**

$$\mu_{\overline{A}}(x) = 1 - \mu_A(x) \tag{2.15}$$

- **Fuzzy Rules** - The fuzzy system behavior is dictated by a set of if-then statements linguistic rules involving fuzzy sets, fuzzy logic and fuzzy inference. Fuzzy rules usually follow the form:

$$if \longrightarrow \textit{A is x then B is y} \tag{2.16}$$

- **Fuzzy Inference System** - The fuzzy inference system is composed of fuzzy sets that operate on fuzzy rules, which makes the knowledge base, and three components, each performing a specific reasoning process.

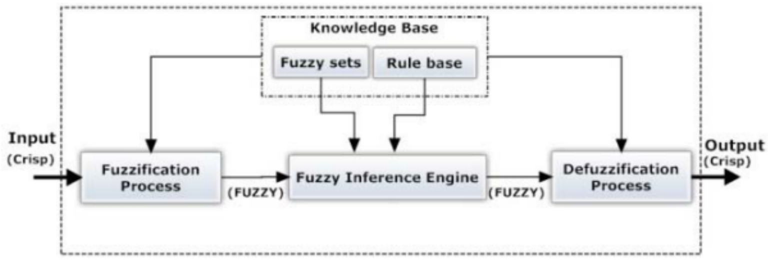


Figure 9: Fuzzy Inference Architecture.

– **Fuzzification:** its a mathematical procedure for converting an element in the universe of discourse into a membership value of the fuzzy set. [33]

– **Fuzzy Inference:** by using the membership functions generated from the a priori method, logical operations and fuzzy rules, allows the mapping of fuzzified inputs into fuzzified outputs.

– **Defuzzification:** the originated fuzzy sets generated by the fuzzy inference process are mathematically combined, most commonly using the Mamdani inference method and Center of Gravity method, to make with a single number as output.

2.2 Research studies in the topic

This section presents a in-depth review of articles about Sensor fusion and the different methods used to fuse the different sensors found in each article. This gives an overview of some studies where sensor fusion is applied in a real world context. When looking for articles relating to sensor fusion in a damage detection context, little was found. Despite that, we can have an idea based on the articles found of possible solutions for damage detection problems.

When looking for articles regarding sensor fusion, the article in question needed to have at least one fusion technique performed in one or more sensor presented. Although little was found regarding damage detection, the majority of articles fall in the activity recognition and fall detection areas. As the years pass we can see an improvement not only in the sensors quality by also the amount of sensor available, for example each smartphone comes equipped with various sensor than can facilitate these problems.

Ujwal Koneru et al., [34] describes a solution to the problem of accurate tracking in Augmented Reality (AR) and Virtual Reality (VR). Tracking in this context is a vital functionality for many applications like gaming, simulation, etc. In this study,the use of accelerometers and gyroscopes to measure rates and accelerations, and a camera sensor to process the position and orientation are presented. The fusion here happens in the highest level of fusion (decision fusion) by using Fuzzy rule sets and adaptive filtering of data. The authors were able to achieve similar accuracy outcome as some of the other commercial tracking systems at a fraction of the cost.

Liang Liu et al.,[35] presents a high level fusion (decision fusion) approach for fall detection application. The study invols the use of two Doppler range control radar sensors and in order to classify the data three classifiers were used, KNN, Bayes and SVM. As for sensor fusion, the authors chose to implement a Choquet fuzzy integral fusion. The fusion of information, resulted in a better accuracy compared to the use of a single sensor, where each classifier produced a reasonable result between 0.88 and 0.97 and the fuzzy integral fusion produced results between 0.95 and 0.98, proving a better overall classification.

Simon O'Regan et al.,[36] in the context of automatic detection of Electroencephalography (EEG) artefacts from head movements, uses a feature fusion approach to compare and increase the accuracy of the detection. In this article the authors extract signals from gyroscopes mounted within an EEG headset and the signals provided from the headset. Three architectures are presented, each one belonging to detection of artifacts from gyroscopes, EEG and the fusion of the two using a SVM classifier. Better accuracy results were obtained using the fused information than only using the signal information from the gyroscopes or EEG headset. It was also stated that the fused information presented a better classification performance and robustness.

On the topic of fall detection, Anita et al., [37] gives a comparison on performance of different Machine Learning (ML) classifiers and the impact of using sensor fusion opposed of using the sensors individually. As for the classifiers the article presents ANN, KNN, Random forest and XGBoost implementations. The data gather for the study, where from a vital signs sensor and IMU sensor, in order to extract the heart rate and position of the user. When comparing the performance across all the different classifiers it was demonstrated that when combining the information from the heart rate with a IMU, the accuracy improved. Despite that, Random forest was the classifier with the best accuracy across all scenarios when applied the windowing technique. The sensor fusion in this case was performed in the feature fusion and data fusion domain.

Paola Pierleoni et al., [38] presents a different approach for a wearable fall detector directed for elderly people. Here the fusion of data is done in the lowest level (data level fusion) where raw data collected from the sensors is fused to give a better output. The raw data is gather from 4 different sensors, an accelerometer, gyroscope, magnetometer and finally a barometer. Fusing this sensor proves a better detection accuracy and better estimation of the position of the user. As for the architecture, the authors fuse the raw data coming from the accelerometer, gyroscope and magnetometer using a Quaternion-based Madgwick filter and later fusing the measures coming from the barometer with a complementary filter. The results compared to other studies in this field provide a better accuracy, in some cases an increase of more than 20%.

Nikhil kumar et al., [39] describes a more related problem to the one presented in this thesis dissertation. Here the authors are trying to build a Internet-of-Things (IoT) platform for reporting vehicle accidents detected using sensor fusion as a way to improve the accuracy of the system. In order to find the model to use in the detection and classification system, it was presented three machine learning models, those being, Naive Bayes, Gaussian mixture model and decision tree. As for sensors, the work is comprised of a 9-axis inertial and GPS built in sensors in the *SAMSUNG Galaxy S8* Android smartphone and a *Sensordrone* which measures environmental variables, such as temperature, humidity, CO, etc. here the fusion happens in the feature fusion level and data fusion level. As for data fusion methods, the article presents the implementation of complementary filter and for feature fusion implements machine learning classifiers and moving-maximum function. As for the results the Naive Bays model proved to outperform the other model with and average score of 0.95.

Filipe Felisberto et al., [40] tackles the problem of monitoring the elderly, focusing in a low-cost deployment. Here the authors present a Wireless Sensor Networks (WSN) , where sensor fusion is integrated to give a more robust understanding of the information. In order to gather the information needed, it was used three sensors, accelerometer, gyroscope and magnetometer. Sensor fusion was used to obtain the correct orientation of the node, with the use of thee data fusion algorithms, Extended Kalman Filter (EKF), Direct Cosine Matrix and control algorithm which fused each individual sample ignoring past knowledge. The study shows that the use of EKF proved to be the most beneficial to cope with noise spikes, smoothing them out immediately and handle the incremental error. This articles presents promising results, even surpassing other previously cited WSN projects in monitoring user's movements.

Table 1, presents for all the articles described above, the sensors used, which level or levels of fusion was the fusion applied, and finally the methods used to accomplish the fusion process.

Article	**Sensors**	**Fusion level**	**Methods**
Ujwal Koneru et al., [34]	Accelerometer, gyroscopes, Camera	High	Fuzzy sets; Adaptive filtering
Liang Liu et al., [35]	Doppler range control radar	High	Choquet fuzzy integral
Simon O'Regan et al., [36]	gyroscopes, EEG headset	Medium	Feature processing, SVM
Anita et al., [37]	IMU, vital signs	Low, Medium	Feature processing, ANN, KNN, Random forest, XGBoost
Nikhil kumar et al., [39]	9-axis inertial GPS, *Sensor-drone*	Low, Medium	Complementary filter, Feature processing, Naive Bayes, Gaussian mixture, Decision tree
Paola Pierleoni et al., [38]	accelerometer, gyroscope magnetometer, barometer	Low	Quaternion-based Madgwick filter Complementary filter
Filipe Felisberto et al., [40]	accelerometer, gyroscope magnetometer	Low	EKF, Direct Cosine Matrix control algorithm

Table 1: Articles information in sensor fusion

3

Approach for fusion benchmark

3.1 Data Analysis

This section describes the beginning of the practical work. After all the literature and knowledge gathering that encapsulates this thesis has finished, the next step, data gathering can begin. Data gathering presents a necessary step in order to follow a well taught out plan.

3.1.1 Data gathering structure

Before approaching the data and performing all the necessary steps for fusion, a better understanding of the setup used to capture the data is needed.

As presented before the main focus of this thesis is the use of sensor fusion to improve the detection of vehicle impacts . There is a need to have at least more than one sensor in the vehicle to perform sensor fusion. Keeping this structure in mind a system containing three devices, each with an accelerometer, gyroscope and a microphone, were installed in a vehicle.

As shown in figure 10, the three devices placed in the vehicle are as follows: windshield, right rear panel and central console. The system was setup this way to test and understand the detection behaviour of each device location and the different combinations the system can have. Note that this setup is specific to the previously mentioned structure. Later on, a new setup is presented with a different layout for the devices.

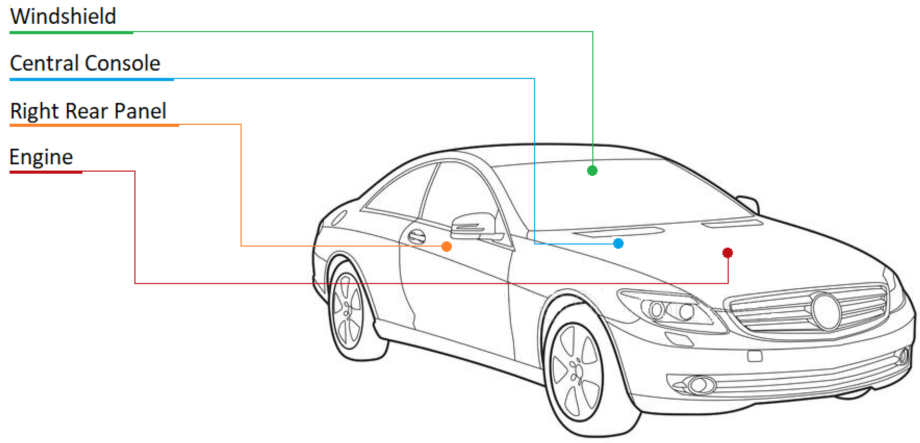


Figure 10: AudiA5 multi-sensor setup.

The following table details the specification of each sensor inside the device used. The same sensor's configurations and models are present in every chosen device location.

Sensor	**Sample Rate**	**Bandwidth**	**Range**
Accelerometer	1600Hz	434Hz	± 8G
Gyroscope	1600Hz	134HZ	± 250 deg/s
Microphone	44100Hz	1000Hz	-

Table 2: Sensor specifications

The three devices were selected due to the capability of recording the vehicle's state at any given time. For this project, only the accelerometer is considered as the main objective set is to fuse the measurements between each accelerometer device location.

3.2 Data Gathering

The next phase consists of collecting data with the previous setup and all the steps to ensure data can be used for machine learning applications. The purpose of this phase is to record and store multiple events in a database so that later, it can be used to train and test the models to detect impact in the vehicle. Before this project, data collection campaigns took place, which helped speed the testing and development of sensor fusion solutions.

With this in mind, a series of planned recordings were carried out, recording the necessary events to build a damage detection dataset. Multiple events with different types of impacts were recorded, allowing the model to learn and detect more than one impact.

For the model to predict if an impact caused damage to the vehicle or not, two types of events were defined: either the impact resulted in damage to the vehicle's exterior or the exterior was undamaged. With this in mind, the recorded events had to replicate as closely as possible real scenarios for which the model could later predict. Note that some events recorded did not have any impact on the vehicle, for example, closing and opening doors, but could be mistaken as one, to help the model correctly predict and understand what classifies as an impact that resulted in damage and background. Due to the dangerous nature of experiments, a meticulous and thought-out safety setup was required not to harm anyone in the process.

The following tables describe the events gathered in the different experiments needed to complete the dataset. Table 3 consists of the events that can result in damage to the exterior of the vehicle, and Table 4 represents the events that do not cause damage to the exterior.

ID	Description
1	Pothole
2	Sewage Cap
3	Speed Bump
4	Expansion Joints
5	Rumble Strips
6	General Bump
7	Door Close
8	Curb Climb
9	Door Open
10	Roof Slap
11	Open/Close Sun Visor

ID	Description
12	Detaching/Fixing Sun Visor
13	Wiper
14	Side Window Opening/Closing
15	Opening/Closing Makeup Mirror
16	Mirror Folding Open/Close
17	Windshield Slap
18	Object Sliding Against Windshield
19	Mount/Dismount GPS Holder
20	Car Wash
21	Open/Close Sunroof
22	Open/Close petrol cap

Table 3: No-Damage events

ID	Description
1	Knock
2	Object Impact
3	Scratching
4	Vehicle hits object
5	Door opens against object

Table 4: Damage events.

All the events listed in the tables above are stored in a file with an associated ID. With this structure, it becomes more effortless in the implementation phase, as only the ids for each event are needed instead of writing the whole name. Note that some events have variants, one example being, the event "Door Open", which has variants for "Trunk", "Front Right", "Back Right", and so on, but only the primary type was described due to a large number of variants.

The list of events can grow for each data-gathering campaign as different events can be recorded if needed. Once a specific event benefits the model's performance, other campaigns will take place. Furthermore, only some of the events presented above were recorded for each vehicle, as recording all of the events would prove a waste of time and could confuse the model when predicting.

The data originated from the collection are stored in two separate files keeping the same name for easier correspondence when loading and processing the data. The measurements containing the sensor values and timestamps are stored in a Hierachical Data Format (HDF5) format, more commonly known as HDF5. The JavaScript Object Notation (JSON) file serves the purpose of storing general attributes of the recording. The following Table 5 lists all the attributes stored in each JSON file.

Attributes
CarID
Initial Timestamp (UNIX timestamp)
Event Label
Event Start Time
Event End Time
Damage Status
Damage Type
Damage Severity
Road Type
Weather Condition

Table 5: JSON attributes list.

3.2.1 Labelling

This section describes the process of labelling the recorded data from the previous section. In order to ensure that all the data is labelled correctly, a two-step setup was implemented.

In the first step, the labelling happens during the data gathering using an internal tool developed to work with the setup mentioned earlier. This tool allows, in real time to specify the time interval an event happens, with the use of a specific hotkey which is pressed at the beginning of the said event until the event ends.

Although this tool allows precise labelling and first-hand confirmation, human error is something to always have into consideration. For this reason, a second internal tool was implemented after all the data had been collected and stored to help fine-tune the label of all the events. The tool used can be visualized in the following image 11 and allow users to change the label meta-data and adjust the time interval of a particular event.

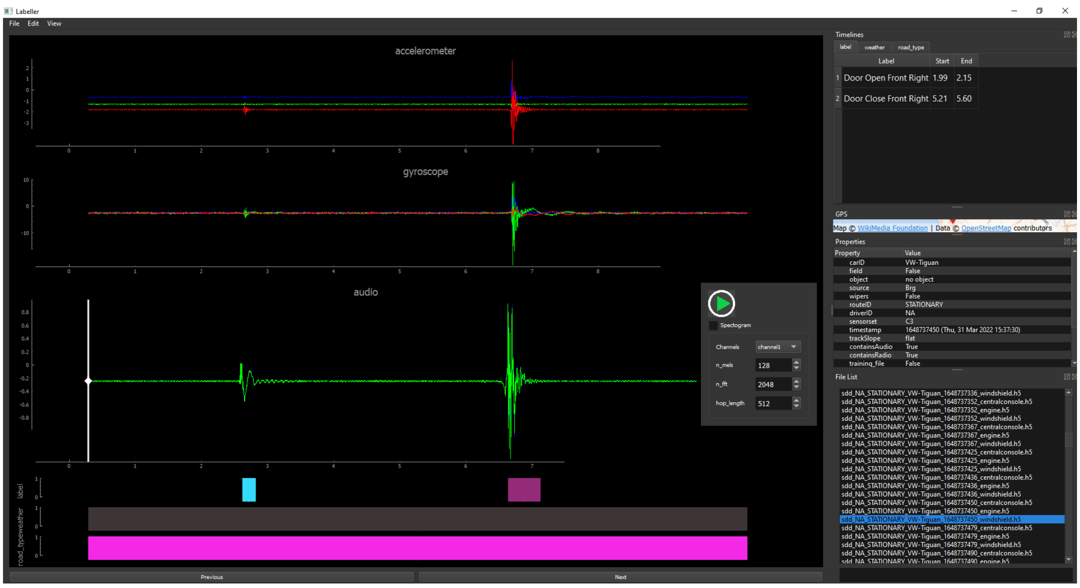


Figure 11: Labeller tool.

This tool was also a valued asset to the first part of data analysis since the data of multiple events can be displayed and shown to the user. This permits us to have a better understanding of how the different device locations record the same event.

3.2.2 Datasets Description

Due to the nature of this work, the dataset needed at least two device spots in different vehicle locations to perform sensor fusion.

Only two of the many datasets collected previously in this work meet the requirements to perform sensor fusion. The nomenclature for each dataset follows the vehicle model name. The first dataset used in this project was the AudiA5 dataset, which, as mentioned before, resulted in the collection of events using an AudiA5 model. The second dataset is the Tiguan dataset. Since the dataset was still in the collection/labelling step, it became available only in the later phases of development.

Both datasets follow the same data structure mentioned in the Data Gathering section, storing all sensor measurements and information with each device location name associated. The difference lies in the labels and the amount/name of the device locations used.

3.2.2.1 AudiA5 Dataset

The events contained in this dataset were recorded with a four-device location setup in an AudiA5 vehicle. The device locations are as follows: windshield, right rear panel, central console and engine. Although the dataset presents four devices, only three are considered for this thesis. Due to the small number of events recorded using the device placed in the engine was left out.

When creating datasets one objective was making it a balanced dataset, meaning there needed to be almost the same number of events with the label 'damage' as events that did not. As shown in the table 12 the dataset consists of 494 events with no damage and 710 with damage.

carID	number of occurences	no damage events	damage events	no damage - damage events precentages
AudiA5	1204	494	710	41.03% 58.97%

Figure 12: AudiA5 damage vs non-damage events.

The following image 13 illustrates the different events stored in the AudiA5 dataset. Note that the events shown in the image are from the latest version of the AudiA5 dataset, which changed from the first version used in this project. Each event has the number of occurrences which resulted in no damage to the vehicle represented in blue and in orange the occurrences that did not.

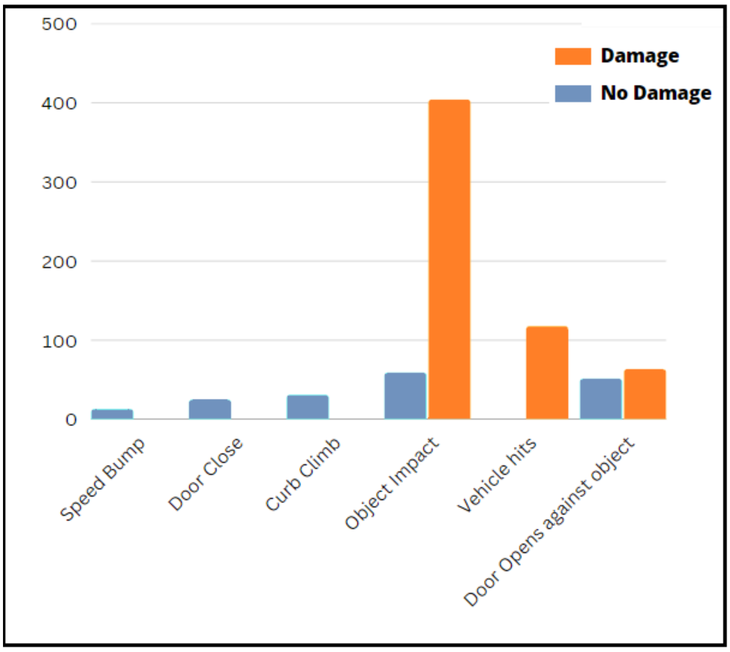


Figure 13: AudiA5 last update events recorded.

3.2.2.2 Tiguan Dataset

Following the same approach used with the AudiA5 dataset, the recording sessions for the Tiguan dataset followed a multi-device locations setup. As opposed to the previous dataset, the number of devices decreased to only three locations, removing the device placed in the right rear panel. In this instance, all device locations are present in the fusion and testing phases.

While working with this dataset, more recording sessions were planned and recorded, which led to a change in the dataset size and performance. The image 14 demonstrates that the dataset consists of 1836 events with the label "damage"set to false and 664 events with the label "damage"set to true, resulting in an unbalanced dataset.

carID	number of occurences	no damage events	damage events	no damage - damage events precentages
VW-Tiguan	2500	1836	664	73.44% 26.56%

Figure 14: Tiguan damage vs non-damage events.

Although the Tiguan dataset follows the same approach as the AudiA5, the events recorded differ. For example, the Knock event represents the majority of events that failed to inflict exterior damage to the vehicle. Applying the same model used in the AudiA5 dataset for this new dataset, as long as it performs relatively the same, validates the results obtained in the AudiA5 and proves once again that sensor fusion can improve the model performance. Another advantage of using different events is that merging both datasets generates more data for training and testing purposes.

The following graph displays the events contained in the Tiguan dataset. The reason the events "object impact"and "vehicle hits object"have both true and false values for the label "damage"happens when the recorded event displayed no visible damage in the car. This is necessary so the model can learn what constitutes impact with damage associated with it and what does not.

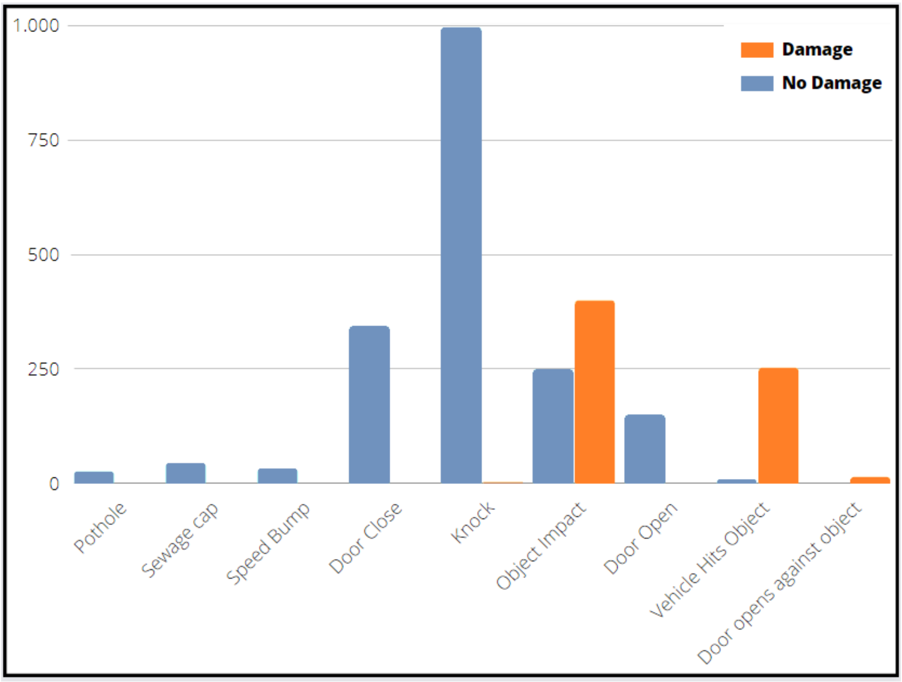


Figure 15: Tiguan events recorded.

3.3 Initial Iteration

This section presents a set of early experiments and different approaches used for the project that ultimately ended up being discarded. Although these approaches failed as possible solutions, they proved to help understand the overall problem which allows to define a more suitable solution.

3.3.1 Vehicle pitch and roll

Research of possible solutions was carried out to find an approach to follow. The method found consists in the fusion of an accelerometer with a gyroscope to determine the pitch and roll of a vehicle. The pitch and roll deemed reasonable as the impact can be detected by the shift in the axis.

The purpose of this approach is to fuse data from an accelerometer and a gyroscope to determine the pitch and roll of the vehicle, allowing to detect of changes in the axis when an impact to the vehicle occurs. Before fusion can happen, some obstacles need to be resolved. The issues surround measurement deviations in the readings over time.

Both sensors suffer from measurement issues in their way. For example, an accelerometer can have jitters in the reading, resulting in incorrect readings. Different from the accelerometer a gyroscope will have a drift over time in the readings making it unreliable. Luckily one can find the solution for both issues with the help of filters.

Although a wide variety of filters could be used, based on the previous research the filters commonly utilised are the low pass filter and high pass filter. The process of implementing each one is simple, as the two filters follow the same logic. The case of the low pass filter only allows frequencies to pass that are below a certain threshold. Same for the High pass filter but the other way around as only high frequencies get allowed to pass through.

The last step is to apply the filters to the respective measurements. As the accelerometers come with the issue of jitters the high pass filter is ideal as it will only allow passing the high frequencies, despite the aformention, when applying a high enough threshold information can end up being removed. Lastly, the low pass filter can help mitigate the drift of the gyroscope removing any high frequency.

A previous study internally concluded that the information from the gyroscope did not contribute any relevant information and that only the accelerometer should be used to detect exterior vehicle impacts. With this knowledge, an approach of fusing an accelerometer with a gyroscope is deemed unfit for this problem.

3.4 Data Preparation

The step to guarantee the data is to standards consists of a thorough study of the data provided. By performing the aforementioned study, we can ensure that any problems with performance and evaluation in the final steps of ML will not likely be related to the data.

3.4.1 Data standardization for analysis

This section details all the work necessary to transform the data so that later sensor fusion methods can be performed. The main pre-processing steps implemented are to apply the necessary rotation matrix and time alignment to ensure the data from all the device locations are standardized.

3.4.1.1 Accelerometer rotation matrix

One of the first necessary steps when performing the analysis of the data was the need to perform a rotation of the data coming from each device location. The need for a rotation happens due to the position of the devices when placed inside the vehicle.

In order to find the rotation matrix to apply for each sensor location, a calculation for the angles based on the desired angles was performed. The rotation matrix was also necessary because there was no previous information on the angle at which the devices were installed.

Figure 16 shows the raw information coming from an accelerometer placed in the windshield of a vehicle. It also shows all axis that represents the acceleration of the device in a given sample. Because the earth has a gravity force of 9.8 ml/s or 1G, the standard angle for all the accelerometers was to have the axis x and y at zero and the z at 1, this only happens if the car is still on a horizontal plane. As shown in the image the three axis do not match the standard.

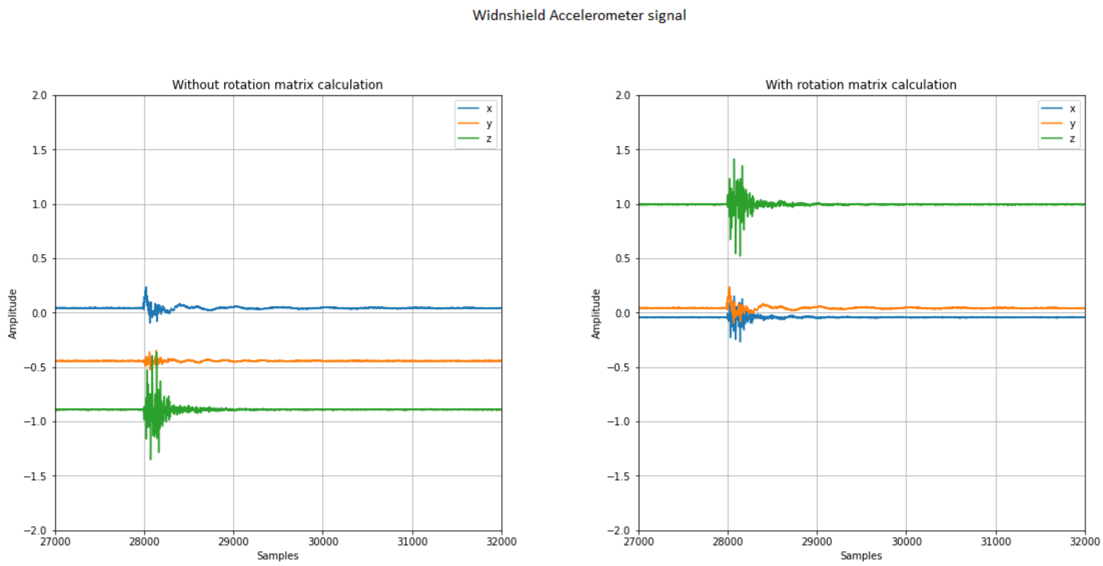


Figure 16: Example of the windshield device signal recorded with and without the rotation matrix applied.

After the right angles found and the rotation matrix created for each of the sensor's locations, the first step deemed completed as we can see in the Figure 11. The rotation matrix is a crucial part of the pre-processing phase. With the angles in the wrong rotation, fusion results in incorrect data deteriorating the model performance.

3.4.1.2 Data alignment

As with the previous step, data alignment is necessary to apply fusion methods to the data. Data alignment happens due to the need to combine data or features from the various sensors. When the sensors are miss-aligned fusing information results in incorrect or outmatched fused data.

First, we need to consider why there is a misalignment in the sensors. This misalignment occurs due to the positions of the devices in the vehicle. For example, if the impact is registered on the right side of the vehicle, the closest device will capture the data first and take longer to be captured by other devices.

To resolve the misalignment, the device placed in the windshield serves as the anchor to align the other two device sensors. As the data of each measurement from the sensors stores the timestamp recorded we can use that detail to shift the data from the other device sensors to match the anchor, which in this case is the device placed on the windshield.

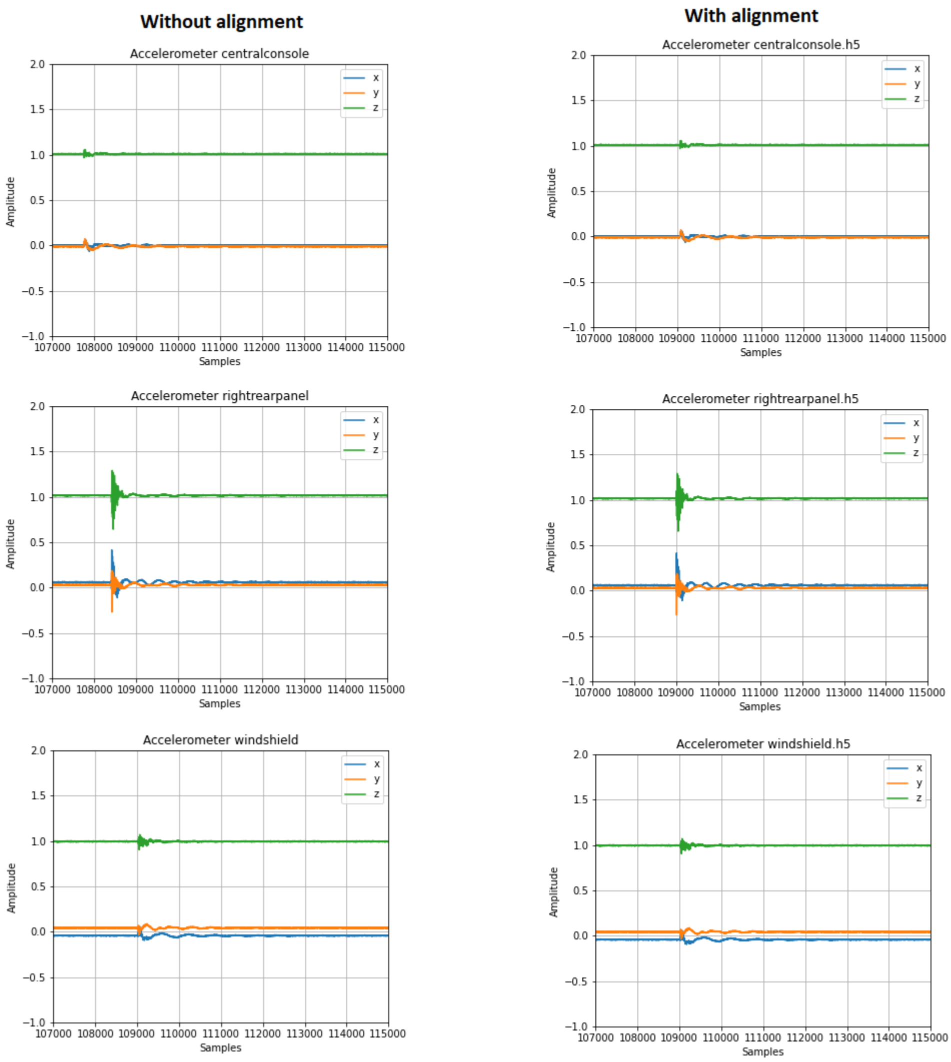


Figure 17: Example of the alignment result.

3.5 Model implementation

This section details all the planning and work needed to implement the damage prediction model. This segment represents an essential part of the thesis as it allows the test and comparison of the results that sensor fusion may bring.

3.5.1 XG-Boost

Although there are many different models, ranging from random forest to complex neural networks, the model chosen for this work was XG-Boost. This decision was made due to previous work and testing internally, which proved for this particular problem that XG-Boost held a better performance when compared to other solutions. Also, because the main focus of this thesis is the improvement that sensor fusion can bring, most of the time was spent on implementing and testing sensor fusion methods.

Now for a brief explanation of the XG-Boost model and the benefits it brings to machine learning and sensor fusion. XG-Boost is part of the gradient-boosted trees algorithm family, a supervised learning algorithm that attempt to predict the target variable by converging the estimates of a set of simpler models.

3.5.2 Augmentation

Augmentations present an important process in machine learning since it can give a different perspective or view for the algorithm, allowing more information. Even tho the augmentation can seem to lose some information, the algorithm has a different way of comprehending the data. For example an image had a rotation applied to it, this augmentation can present a challenge in guessing the image label with the naked eye. This rotation can allow the model for a better understanding of the enviromnet and help improve the performance.

After all the events have been generated and are ready to be used as input for the model to train and predict, some steps are left to execute. The first one is the augmentations. In a total of four augmentations, only two were used to transform the data in some sense. The augmentations are applied to all the sensor locations measurements and only for the events in the training files as we do not want to transform the data in the testing dataset.

All the augmentations implemented are as follows, rotation augmentation, as the name implies performs a rotation to all the user-specified axis. The other augmentation used for the final phase of this work is time shift augmentation. In contrast to rotation augmentation, instead of rotating the axis, all the data from each sensor location are shifted with a user specific value.

Finally, the last three augmentations were left out due to lower performance when compared with the prior augmentations. The first augmentation utilises a Gaussian distribution to generate noise applied to the raw data. The user can define minimum, and maximum standard deviation values to generate the noise from the Gaussian distribution and apply the number of different Gaussian distributions. The final augmentation present is axis offset to offset along with the minimum, maximum and number of offsets to apply to each user defined axis.

3.6 Sensor Fusion

In this section a detailed description of the planing and implementation of each sensor fusion method used.

3.6.1 Complementary filter

The first sensor fusion method used was the complementary filter, which falls into the data fusion methods. As explained in Chapter 2 of this thesis, the complementary filter is a low-level fusion where the raw data measurements from the sensors are fused to give a more robust understanding of the events.

Of all the different sensor fusion methods, this one proved to be a good starting point due to the straightforward approach with little to no need to perform changes to the data in order to achieve fusion. The only pre-processing done to the data is presented in the chapter Data Preparation. with more details.

The first step to perform this method is to have all the data from the multiple sensors in the same standard, in this case being the same rotation and aligned with the time. If sensor fusion were to happen without these steps, the incorrect representation of the events and the surroundings would be passed to the models and lead to a much lower performance in comparison to using a single sensor.

After the first step, the implementation of the complementary filter takes place. As mentioned earlier it is one of the simpler methods to implement as long the data is with the right stantards. As the following equation shows, it consists of the addition of the axis values of each sensor measurement at a given timestamp multiplied by a user-defined weight.

$$[H]X_1 \cdot W_1 + X2 \cdot W_2 + ... = FuseX \tag{3.1}$$

With all the planing and implementation of the necessary steps completed the only thing left is to test and find the best weight for each sensor used.

Multi-Sensor

Performance

- Test score: 0.778
- Validation score: 0.905

Fusion weight values

- rightrearpanel: 0.1
- centralconsole: 0.1
- centralconsole: 0.8

Confusion Matrix

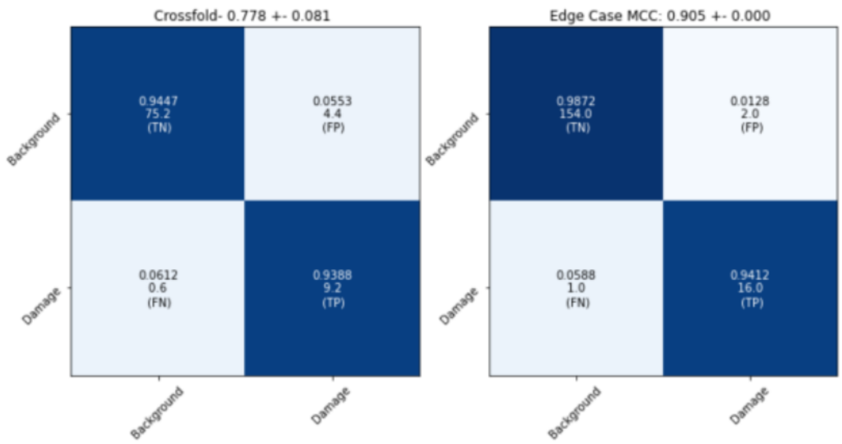


Table 6: Complementary filter MCC performance

After testing with multiple combinations of sensor locations and weights, the image above proved to be the best combination found. Here all three sensor locations were included and the majority of the weight was from the windshield location. This combination also points to the fact that the windshield is more reliable than the other locations, but still benefits from information the other two locations provide.

This point marks the end of the first objective setup for this thesis. As mentioned early the first objective of this thesis consisted in proving that a multi-sensor setup improved the performance of a single-sensor setup. When comparing the MCC score of the best sensor location combination found for the complementary filter and the best single sensor setup, we can see an improvement of 0.117 when using the complementary filter.

3.6.2 Feature Fusion

With the first objective completed and sensor fusion proved to work and improve the performance of the model, the next step of sensor fusion was to test feature fusion and compare if this method could improve results from the complementary filter.

Much of the pre-processing applied to the datasets had already been done when testing and implementing the complementary filter. Because of that much of the time spent in the pre-processing could be now spent on testing and developing the feature fusion method. The method that ultimately ended up being used to fuse the sensor locations, consists in 3 parts.

The big step was generating events with a time window interval so that later feature extraction and calculation could be applied. An event in this case does not represent necessarily an impact on the car. The algorithm generates events based on the magnitude of the accelerometer measurements.

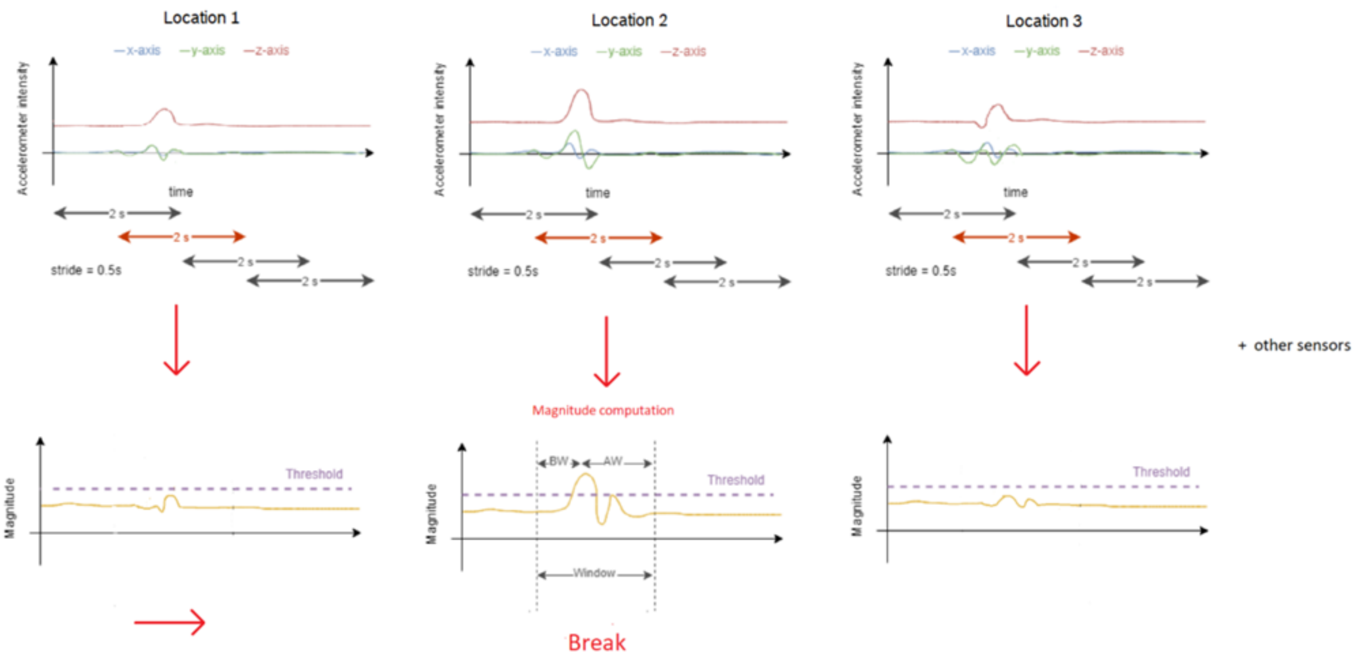


Figure 18: Feature fusion framework.

The image Figure 16 represents the workflow of the first step in feature fusion. The first segment involves looping through all the locations accelerometer data in a two-second time frame. In a cycle containing all the specified locations, for each one, the magnitude of the data is calculated in the two-second interval set before. After acquiring the magnitude, a threshold is passed to check if there is a point where the magnitude surpasses it. In the positive case, the magnitude transcends the threshold a window is created with a predefined size, and the cycle breaks for that two-second window. Note that the window created has a length before the perceived point with the notation $BW(Before\ window)$ and length after

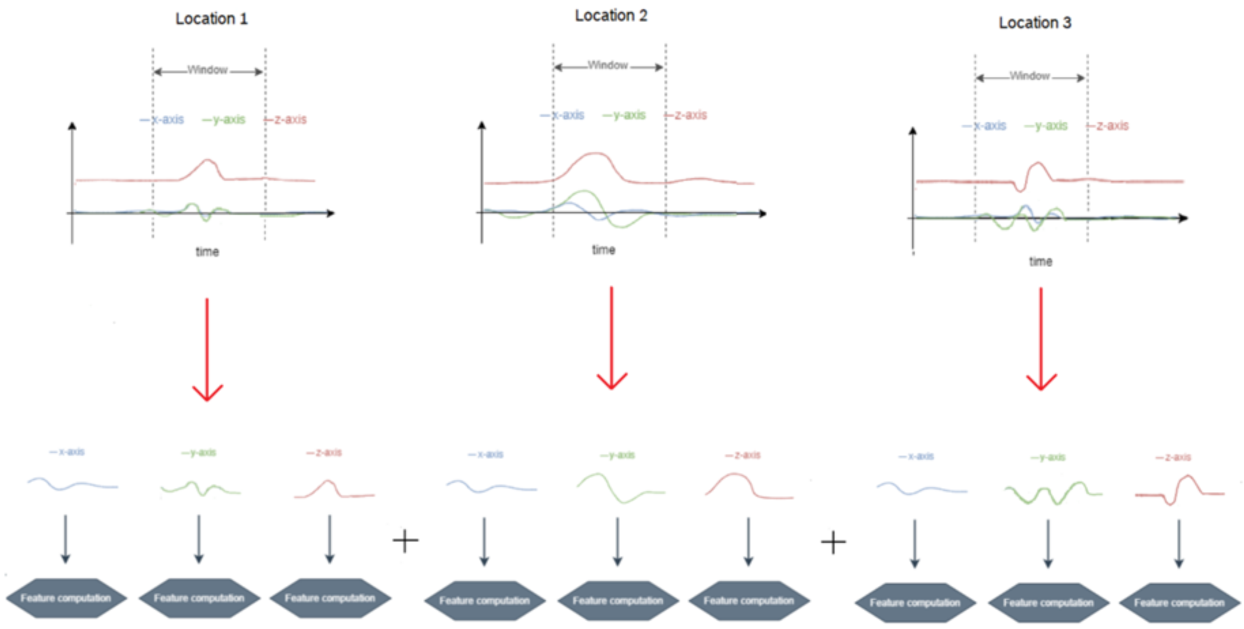


Figure 19: Feature computation framework.

with the notation $AW(After\ window)$. This window is later used to define the time interval feature computation that will take place and is explained later on in this chapter.

In the case the magnitude fails to reach the threshold, the next location gets cycled in and goes throw the same process. Given the event that none of the locations magnitudes surpass the threshold, the next two-second interval is looped and no window is created. The final segment in this process is composed of checking the label for the specified window and setting the label as a $background$ or a $damage$.

The next step implemented begins once a window has been created as mentioned earlier. As presented in the image Figure 20, the same window is used in all location accelerometer data to delimit the interval where feature computation happens. The most important part of feature fusion, as the name suggests, instead of raw fusion data from the measurements we fuse features extracted from them. As presented above for each axis multiple features are calculated and stored. Finally, all the features from each of the locations are stored together and later used as input for the model to predict if the vehicle suffered damage or not.

Before the features get used as input for the damage classification model, there's still a final step implemented. This step involves applying a correlation method to remove features with a high correlation between them. The method chosen for this process is the Pearson Correlation method.

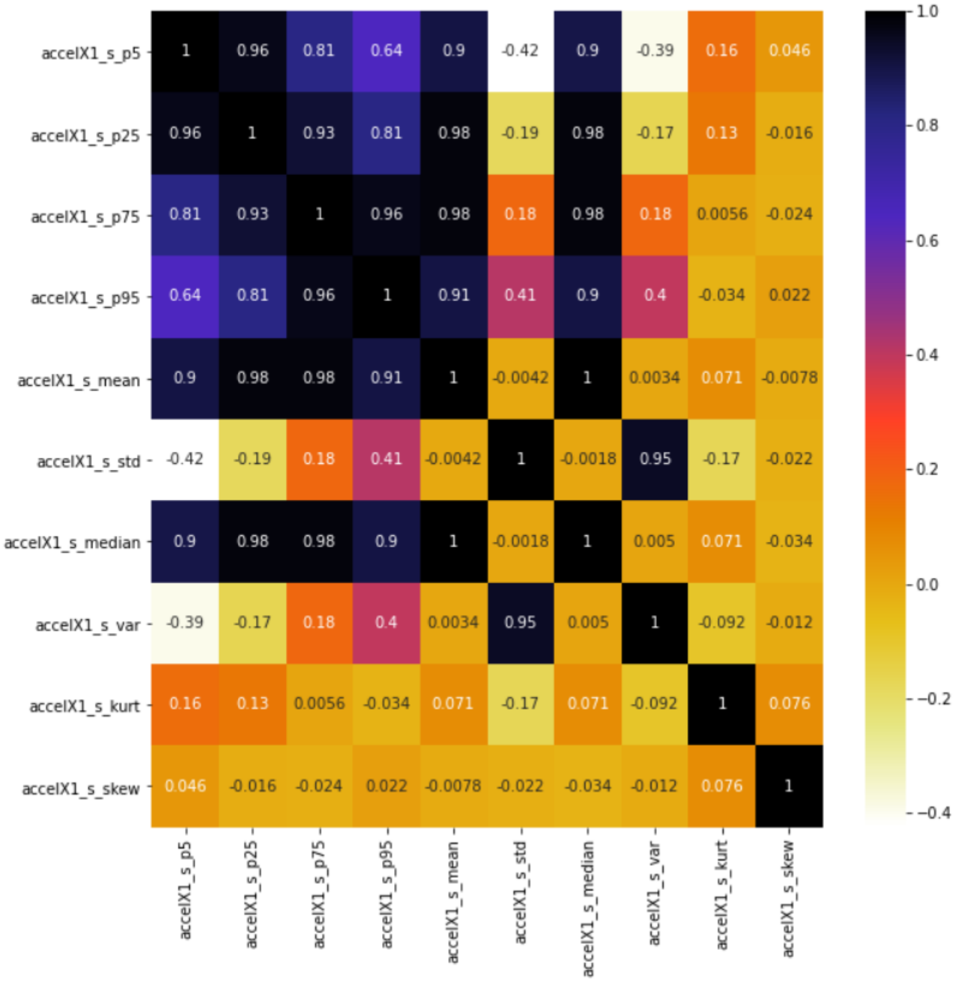


Figure 20: Correlation between features.

This method calculates the correlation between every feature and passes a threshold to evaluate which feature to remove. As shown in the figure Figure 21 the correlation can vary in a range o -1 to 1. These values can help us evaluate if a feature adds any value or if it can be deleted to reduce redundancy. For this purpose, a threshold of 0.94 was defined if a value is higher than 0.94, this also works if it is less than -0.94, and one of the features is discarded.

3.7 Results

This section represents the final step in the development process. Here lies all the performance for all the models and methods used. Due to the nature of this work, this section holds substantial importance as it compares the performance of a single-device setup and a multi-device setup.

Concerning the evaluation process, a metric must define as best suited to this thesis work. If the only element to guide the changes and iteration of the model were the accuracy, it would prove difficult as almost no information could be helpful to know whether the model is performing poorly. For this same reason, the metric chosen was Matthews Correlation Coefficient (MCC). This choice was made since this exact metric takes into consideration every element of the confusion matrix, which contains the True Positive (TP), False Positive (FP), True Negative (TN), False Negative (FN). Another benefit of using this metric is regarding unbalanced datasets. Because the algorithm generates events based on the magnitude of the accelerometer measurements, a condition can happen where there are more events in the test dataset than in the training dataset.

This metric gives values within a range of -1 and 1. These values have each a representation associated with each of them, the closest the number is to 1 the closer the model is to a perfect prediction, on the contrary, closer to -1 represents an inverse prediction. Lastly, the closer the MCC score is to 0 the more random the prediction is and consequently can tell us that the model is failing to learn and picks the prediction at random.

$$MCC = \frac{(TP \cdot TN - FP \cdot FN)}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{3.2}$$

Another performance evaluation to be had in mind in this stage of the development was the number of FPs found in the test dataset confusion matrix. The element FP, represents the events that were incorrectly predicted to result in damage to the exterior of the vehicle where truthfully no damage had been done. Here the priority is to keep the FP to the lowest number possible as it could trigger any false alarm or system in place in the vehicle for the eventual circumstance the vehicle suffers damage.

Ultimately the test dataset needed to be untouched when comparing device setups performance. This ensures that the comparison between the two setups is based on the same test events.

3.7.1 Complementary Filter

The following table demonstrates, side by side, the performance based on the MCC metric obtained on the best performing models for the best single-device setup and multi-device setup. Apart from the MCC score, the hyperparameters and confusion matrix are also shown. This process was only done to the AudiA5 dataset as the complementary filter was used as a proof of concept in order for feature fusion to take place.

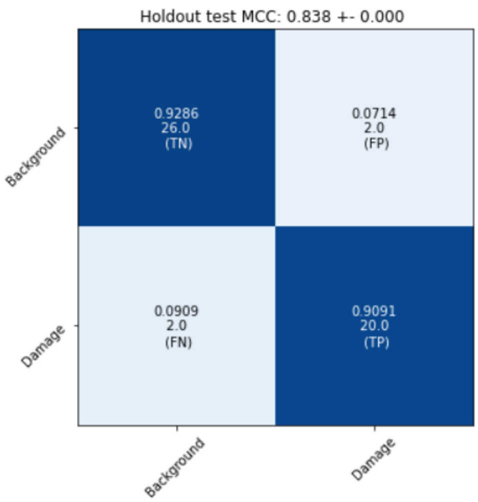
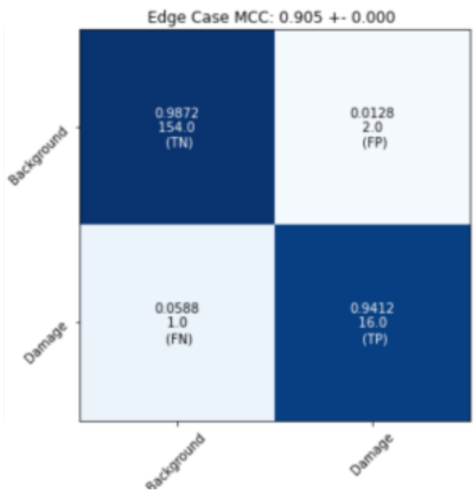
Single-Sensor	**Multi-Sensor**
Performance	
• Test score: 0.782	• Test score: 0.778
• Validation score: 0.838	• Validation score: 0.905
Hyperparamters	
• n_estimators: 1000	• n_estimators: 1000
• max_depth: 6	• max_depth: 6
• eta: 0.05	• eta: 0.05
• colsample_bytree: 1	• colsample_bytree: 1
• scale_pos_weight: 1	• scale_pos_weight: 1
Confusion Matrix	
	

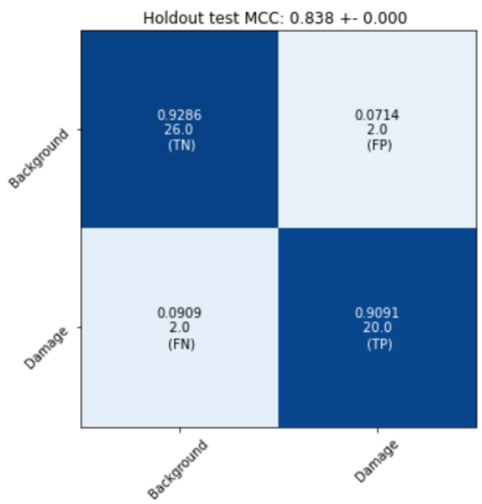
Table 7: Complementary filter Single VS Multi sensor scores

As shown in the previous table, when using a multi-sensor setup, the detection model improves up to 12%. The complementary filter setup uses all three sensor locations, helping the model understand the environment. Regarding the false positives, as demonstrated in the confusion matrix, the complementary filter can maintain a low number of cases, thus lowering the possibility of a false alarm.

3.7.2 Feature fusion

Similar to the complementary filter, the following tables portray the performance, side by side, from both single and multi-device setups. Note that this time for feature fusion, instead of only showing the performance comparison on the AudiA5, it also presents the performance obtained on the Tiguan dataset. In order to validate the results acquired previously, by applying the same model and process to a new dataset, any flaws and irregularities can be detected and improved upon.

Single-Sensor	**Multi-Sensor**
Performance	
• Test score: 0.782	• Test score: 0.811
• Validation score: 0.838	• Validation score: 0.970
Hyperparamters	
• n_estimators: 1000	• n_estimators: 1000
• max_depth: 6	• max_depth: 6
• eta: 0.05	• eta: 0.05
• colsample_bytree: 1	• colsample_bytree: 1
• scale_pos_weight: 1	• scale_pos_weight: 1
Confusion Matrix	



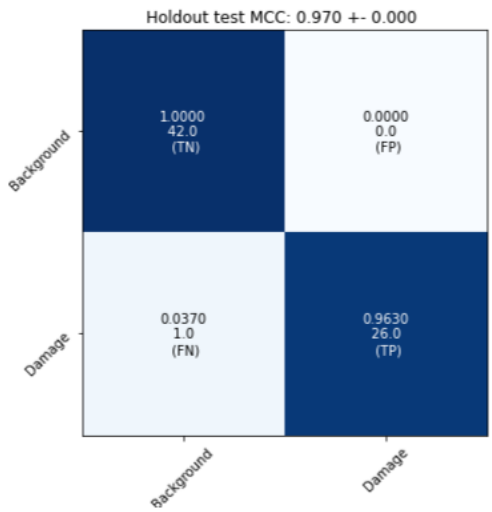
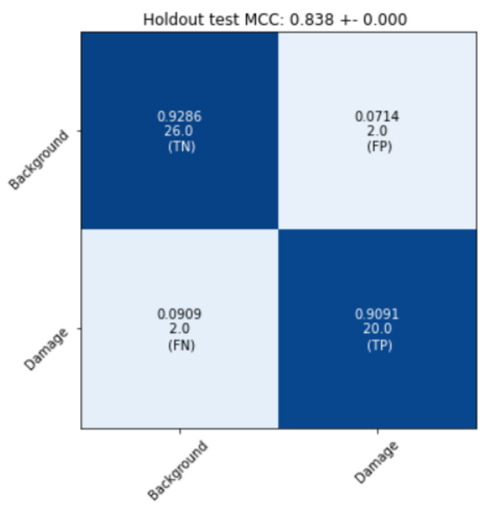


Table 8: Feature fusion Single VS Multi sensor scores for AudiA5

As mentioned before, the following table displays the performance comparison between a single-device setup and the feature fusion method used to enable a multi-device setup.

Single-Sensor	**Multi-Sensor**
Performance	
• Test score: 0.782	• Test score: 0.811
• Validation score: 0.909	• Validation score: 0.913
Hyperparamters	
• n_estimators: 1000	• n_estimators: 1000
• max_depth: 6	• max_depth: 6
• eta: 0.05	• eta: 0.05
• colsample_bytree: 1	• colsample_bytree: 1
• scale_pos_weight: 1	• scale_pos_weight: 1
Confusion Matrix	



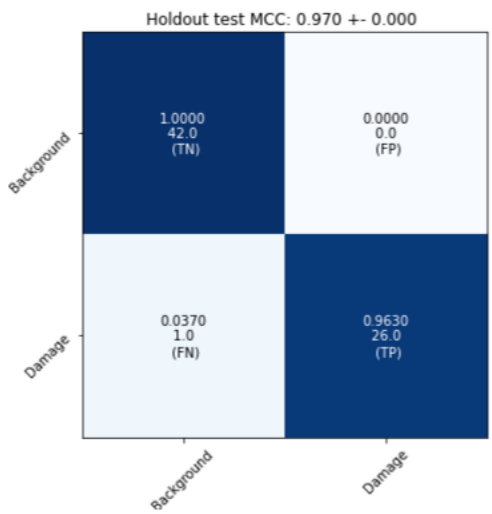


Table 9: Feature fusion Single VS Multi sensor scores for Tiguan

As described in the state of the art section of this thesis, sensor fusion can improve the performance of machine learning solutions by combining information from multiple sensor measurements. With the use of complementary filter and feature fusion methods, the model performance improved compared to single sensor performance. Although the accuracy can give some information about how the model is performing it doesn't take in consideration the TP and FP. With this in mind the performance is evaluated by two metrics one being the mcc score that takes in consideration all the elements in the confusion matrix and the number of FP as this is a algorithm to detect impacts there is a need to minimize the percentage of FP occurrences thus not to set any potential alarms.

In conclusion, the use of sensor fusion looks promising in a vehicle damage detection environment, opening more possibilities and developments in the sensor fusion area to improve detection performance even further. Although sensor fusion proved to work, many improvements can still be applied to all phases of this project. Some solutions and modifications are presented in the future work section.

4

Conclusion and Future work

The purpose of this chapter splits into two parts. The first one describes the work's final observations and thoughts. Moreover, the second part is reserved for presenting an alternative approach and future work related to sensor fusion.

4.1 Conclusion

As mentioned earlier, this main objective of this thesis is to combine multiple sources of information by using sensor fusion methods and consequently improve the performance of the already existing damage detection model. First, a data-level fusion approach was proposed to prove if sensor fusion could improve the model performance. Before applying any modification to the sensor data, a detailed analysis of the already implemented system was required. After the analysis, the processing step began by applying the necessary steps so sensor fusion could occur. The first modification was finding the rotation matrix for two locations so that all the sensors had the same rotation. as a The second transformation required was aligning the sensor's measurements so that the events match all the sensor's locations.

After all the required steps were established, the first implementation of sensor fusion started. As mentioned before, to test and validate the benefits of sensor fusion, the method of choice was the complementary filter, as it proved simple to implement. As was foreseeable, using multiple sensors improved the model performance, in this case, by up to 7% while keeping the false positives to a low occurrence. Completing the first objective opens the doors to testing other methods to help improve the performance even further.

The next step consisted of implementing a medium fusion approach known as feature fusion. As the name implies, the fusion does not happen at the data level but after a set of features extracted from the sensors' measurements. For this reason, the already implemented model had to be changed, and the pipeline had to consider multiple sensors. Also, a more dynamic approach was developed to help test and compare multiple combinations of sensors. All details regarding the work to change the model pipeline are described in the section Feature fusion.

By this point, the only usable dataset was the AudiA5, as the Tiguan was still in the recording and labelling phases. Sens fusion can be validated and proved to work in different environments by testing and comparing results from both datasets. The last phase of this project consisted of experimenting and comparing different sensor fusion combinations with both datasets combined.

Concerning the objectives of this thesis, even though the main objective of proving the benefits of sensor fusion in a damage detection context was completed, one objective failed. Due to time and hardware constraints, a multi-sensor fusion setup in a vehicle with multiple devices could not be implemented and is described in the future work section of this thesis. Regarding the primary goal of this project, as expected from the state-of-the-art section, the results proved promising and open the possibility for diverse experiments in the area of sensor fusion.

4.2 Future Work

Due to the possibilities offered by the context of this thesis theme and promising results from using sensor fusion, a different approach for some of the phases of development and next steps can be presented. Initially, by recording more diverse events, the detection model's performance could validate the use of sensor fusion even further. Concerning sensor fusion methods, a new high-level fusion approach can help improve the model prediction and increase the understanding of the environment. The work presented in this thesis can be used with a high-level fusion method, allowing multiple experiments with different fusion strategies.

Although the complementary filter was used and compared with a single sensor setup, the main objective consisted in proving that sensor fusion can improve the model performance. The next step is combining the features from feature fusion in conjunction with the result from the complementary filter to take advantage of both methods' benefits. Regarding the augmentations, the next step would be adding and testing new augmentations found in the literature phase and combining them with existing ones to improve performance. Besides using Pearson correlation to remove highly correlated features, the method can again be used to remove features with a significant correlation with the prediction class, consequently removing redundancy.

The last two future work set for this work involves the implementation of a multi-sensor setup in a vehicle containing additional device locations than the previous experiments. By adding five to seven more device locations, many combinations can be tested and later compared to find the best setup by trying to maintain the lowest possible number of devices and good performance. Finally, more sensors can be used to perform sensor fusion, enhancing the understanding of the environment.

Bibliography

[1] J. M. Lourenço. *The NOVAthesis LaTeX Template User's Manual*. NOVA University Lisbon. 2021. url: https://github.com/joaomlourenco/novathesis/raw/master/template.pdf (cit. on p. ii).

[2] J. Esteban et al. "A Review of data fusion models and architectures: towards engineering guidelines". en. In: *Neural Computing & Applications* 14.4 (Dec. 2005), pp. 273–281. issn: 1433-3058. doi: 10.1007/s00521-004-0463-7. url: https://link.springer.com/article/10.1007/s00521-004-0463-7 (visited on 01/20/2022) (cit. on p. 3).

[3] F. E. White. *Data Fusion Lexicon*. en. Tech. rep. JOINT DIRECTORS OF LABS WASHINGTON DC, Oct. 1991. url: https://apps.dtic.mil/sti/citations/ADA529661 (visited on 01/20/2022) (cit. on p. 3).

[4] J. Z. Sasiadek. "Sensor fusion". en. In: *Annual Reviews in Control* 26.2 (Jan. 2002), pp. 203–228. issn: 1367-5788. doi: 10.1016/S1367-5788(02)00045-7. url: https://www.sciencedirect.com/science/article/pii/S1367578802000457 (cit. on p. 3).

[5] D. Hall and J. Llinas. "An introduction to multisensor data fusion". In: *Proceedings of the IEEE* 85.1 (Jan. 1997), pp. 6–23. issn: 1558-2256. doi: 10.1109/5.554205 (cit. on pp. 3, 5).

[6] W. Elmenreich. "An introduction to sensor fusion". In: (), pp. 1–28. url: https://www.researchgate.net/profile/Wilfried-Elmenreich/publication/267771481_An_Introduction_to_Sensor_Fusion/links/55d2e45908ae0a3417222dd9/An-Introduction-to-Sensor-Fusion.pdf (cit. on pp. 4–8, 10).

[7] S. Ben Ayed, H. Trichili, and A. M. Alimi. "Data fusion architectures: A survey and comparison". In: *2015 15th International Conference on Intelligent Systems Design and Applications (ISDA)*. ISSN: 2164-7151. Dec. 2015, pp. 277–282. doi: 10.1109/ISDA.2015.7489238 (cit. on pp. 5, 6, 8).

[8] D. Hall and J. Llinas, eds. *Multisensor Data Fusion*. DOI: 10.1201/9781420038545. Boca Raton: CRC Press, June 2001. isbn: 978-0-429-12747-2 (cit. on pp. 5, 6).

[9] A. Gad and M. Farooq. *Data fusion architecture for Maritime Surveillance*. url: https://ieeexplore.ieee.org/abstract/document/1021189 (visited on 01/24/2022) (cit. on p. 6).

[10] B. Chandrasekaran, S. Gangadhar, and J. M. Conrad. "A survey of multisensor fusion techniques, architectures and methodologies". In: *SoutheastCon 2017*. ISSN: 1558-058X. Mar. 2017, pp. 1–8. doi: 10.1109/SECON.2017.7925311 (cit. on p. 7).

[11] S. Beddar-Wiesing and M. Bieshaar. "Multi-Sensor Data and Knowledge Fusion - A Proposal for a Terminology Definition". en. In: *undefined* (2020). url: https://www.semanticscholar.org/paper/Multi-Sensor-Data-and-Knowledge-Fusion-A-Proposal-a-Beddar-Wiesing-Bieshaar/41e8c9757e950aadea79d947176cab44776a71b2 (visited on 01/31/2022) (cit. on pp. 9, 10).

[12] B. Dasarathy. "Sensor fusion potential exploitation-innovative architectures and illustrative applications". In: *Proceedings of the IEEE* 85.1 (Jan. 1997), pp. 24–38. issn: 1558-2256. doi: 10.1109/5.554206 (cit. on pp. 9, 10).

[13] H. F. Nweke et al. "Data fusion and multiple classifier systems for human activity detection and health monitoring: Review and open research directions". en. In: *Information Fusion* 46 (Mar. 2019), pp. 147–170. issn: 1566-2535. doi: 10.1016/j.inffus.2018.06.002. url: https://www.sciencedirect.com/science/article/pii/S1566253518304135 (cit. on p. 11).

[14] A. Assa and F. Janabi-Sharifi. "A Kalman Filter-Based Framework for Enhanced Sensor Fusion". In: *IEEE Sensors Journal* 15.6 (June 2015), pp. 3281–3292. issn: 1558-1748. doi: 10.1109/JSEN.2014.2388153 (cit. on pp. 11, 12).

[15] M. H. Haider et al. "Comparison of Complementary and Kalman Filter Based Data Fusion for Attitude Heading Reference System". In: vol. 1919. Feb. 2017. doi: 10.1063/1.5018520 (cit. on pp. 12, 13).

[16] R. C. King et al. "Application of data fusion techniques and technologies for wearable health monitoring". en. In: *Medical Engineering & Physics* 42 (Apr. 2017), pp. 1–12. issn: 1350-4533. doi: 10.1016/j.medengphy.2016.12.011. url: https://www.sciencedirect.com/science/article/pii/S1350453317300152 (cit. on p. 12).

[17] A. A. Aguileta et al. "Multi-Sensor Fusion for Activity Recognition—A Survey". en. In: *Sensors* 19.17 (Jan. 2019), p. 3808. issn: 1424-8220. doi: 10.3390/s19173808. url: https://www.mdpi.com/1424-8220/19/17/3808 (visited on 01/31/2022) (cit. on p. 12).

[18] M. D. *Complementary Filter - an overview | ScienceDirect Topics*. url: https://www.sciencedirect.com/topics/computer-science/complementary-filter (visited on 01/20/2022) (cit. on p. 13).

[19] N. A. Semary et al. “Fruit-Based Tomato Grading System Using Features Fusion and Support Vector Machine”. en. In: *Intelligent Systems’2014*. Advances in Intelligent Systems and Computing. DOI: 10.1007/978-3-319-11310-4_35. Springer, Cham, 2015, pp. 401–410. isbn: 978-3-319-11309-8 978-3-319-11310-4. url: https://link.springer.com/chapter/10.1007/978-3-319-11310-4_35 (visited on 01/17/2022) (cit. on p. 13).

[20] A. Wang et al. “Accelerating wrapper-based feature selection with K-nearest-neighbor”. en. In: *Knowledge-Based Systems* 83 (July 2015), pp. 81–91. issn: 0950-7051. doi: 10.1016/j.knosys.2015.03.009. url: https://www.sciencedirect.com/science/article/pii/S0950705115001033 (visited on 01/17/2022) (cit. on p. 14).

[21] D. R. Scott, G. M. Flachs, and P. T. Gaughan. “Sensor fusion using K-nearest neighbor concepts”. In: *SPIE Digital Library*. Vol. 1383. SPIE, Apr. 1991, pp. 367–378. doi: 10.1117/12.25272. url: https://www.spiedigitallibrary.org/conference-proceedings-of-spie/1383/0000/Sensor-fusion-using-K-nearest-neighbor-concepts/10.1117/12.25272.full (visited on 01/14/2022) (cit. on p. 14).

[22] Y. Li, W. Dai, and W. Zhang. “Bearing Fault Feature Selection Method Based on Weighted Multidimensional Feature Fusion”. In: *IEEE Access* 8 (2020), pp. 19008–19025. issn: 2169-3536. doi: 10.1109/ACCESS.2020.2967537 (cit. on p. 14).

[23] D. Ruta and B. Gabrys. “An Overview of Classifier Fusion Methods”. en. In: *Computing and Information Systems* 7.1 (Feb. 2000), pp. 1–10. issn: 1352-9404. url: https://eprints.bournemouth.ac.uk/9649/ (visited on 01/31/2022) (cit. on p. 14).

[24] A. D. Dongare, R. R. Kharde, and A. D. Kachare. *Introduction to Artificial Neural Network* (cit. on p. 15).

[25] R. Uhrig. “Introduction to artificial neural networks”. In: *Proceedings of IECON ’95 - 21st Annual Conference on IEEE Industrial Electronics*. Vol. 1. Nov. 1995, 33–37 vol.1. doi: 10.1109/IECON.1995.483329 (cit. on p. 15).

[26] R. Gravina et al. “Multi-sensor fusion in body sensor networks: State-of-the-art and research challenges”. en. In: *Information Fusion* 35 (May 2017), pp. 68–80. issn: 1566-2535. doi: 10.1016/j.inffus.2016.09.005. url: https://www.sciencedirect.com/science/article/pii/S156625351630077X (cit. on p. 16).

[27] D. Freedman. “Overview of decision level fusion techniques for identification and their application”. In: *Proceedings of 1994 American Control Conference - ACC ’94*. Vol. 2. June 1994, 1299–1303 vol.2. doi: 10.1109/ACC.1994.752269 (cit. on p. 16).

[28] F. Castanedo. “A Review of Data Fusion Techniques”. en. In: *The Scientific World Journal* 2013 (Oct. 2013), e704504. issn: 2356-6140. doi: 10.1155/2013/704504. url: https://www.hindawi.com/journals/tswj/2013/704504/ (visited on 01/28/2022) (cit. on pp. 16, 17).

[29] C. Chen, R. Jafari, and N. Kehtarnavaz. "Improving Human Action Recognition Using Fusion of Depth Camera and Inertial Sensors". In: *IEEE Transactions on Human-Machine Systems* 45.1 (Feb. 2015), pp. 51–61. issn: 2168-2305. doi: 10.1109/THMS.2014.2362520 (cit. on p. 16).

[30] L. Zadeh. "Fuzzy logic". In: *Computer* 21.4 (Apr. 1988), pp. 83–93. issn: 1558-0814. doi: 10.1109/2.53 (cit. on p. 17).

[31] R. M. Abdelmoneem, E. Shaaban, and A. Benslimane. "A Survey on Multi-Sensor Fusion Techniques in IoT for Healthcare". In: *2018 13th International Conference on Computer Engineering and Systems (ICCES)*. Dec. 2018, pp. 157–162. doi: 10.1109/ICCES.2018.8639188 (cit. on p. 17).

[32] J.-S. Jang. "ANFIS: adaptive-network-based fuzzy inference system". In: *IEEE Transactions on Systems, Man, and Cybernetics* 23.3 (May 1993), pp. 665–685. issn: 2168-2909. doi: 10.1109/21.256541 (cit. on p. 17).

[33] N. Sabri. *(PDF) Fuzzy inference system: Short review and design*. en. url: https://www.researchgate.net/publication/280739444_Fuzzy_inference_system_Short_review_and_design (visited on 01/28/2022) (cit. on p. 18).

[34] U. Koneru, S. Redkar, and A. Razdan. "Fuzzy Logic Based Sensor Fusion for Accurate Tracking". en. In: *Advances in Visual Computing*. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, Sept. 2011, pp. 209–218. isbn: 978-3-642-24030-0 978-3-642-24031-7. doi: 10.1007/978-3-642-24031-7_21. url: https://link.springer.com/chapter/10.1007/978-3-642-24031-7_21 (visited on 01/20/2022) (cit. on pp. 19, 21).

[35] L. Liu et al. "Fall detection using doppler radar and classifier fusion". In: *Proceedings of 2012 IEEE-EMBS International Conference on Biomedical and Health Informatics*. ISSN: 2168-2208. Jan. 2012, pp. 180–183. doi: 10.1109/BHI.2012.6211539 (cit. on pp. 19, 21).

[36] S. O'Regan, S. Faul, and W. Marnane. "Automatic detection of EEG artefacts arising from head movements using EEG and gyroscope signals". en. In: *Medical Engineering & Physics* 35.7 (July 2013), pp. 867–874. issn: 1350-4533. doi: 10.1016/j.medengphy.2012.08.017. url: https://www.sciencedirect.com/science/article/pii/S1350453312002445 (cit. on pp. 19, 21).

[37] A. Ramachandran, A. Ramesh, and A. Karuppiah. "Evaluation of Feature Engineering on Wearable Sensor-based Fall Detection". In: *2020 International Conference on Information Networking (ICOIN)*. ISSN: 1976-7684. Jan. 2020, pp. 110–114. doi: 10.1109/ICOIN48656.2020.9016479 (cit. on pp. 19, 21).

[38] P. Pierleoni et al. "A Wearable Fall Detector for Elderly People Based on AHRS and Barometric Sensor". In: *IEEE Sensors Journal* 16.17 (Sept. 2016), pp. 6733–6744. issn: 1558-1748. doi: 10.1109/JSEN.2016.2585667 (cit. on pp. 20, 21).

[39] N. Kumar, D. Acharya, and D. Lohani. “An IoT-Based Vehicle Accident Detection and Classification System Using Sensor Fusion”. In: *IEEE Internet of Things Journal* 8.2 (Jan. 2021), pp. 869–880. issn: 2327-4662. doi: 10.1109/JIOT.2020.3008896 (cit. on pp. 20, 21).

[40] F. Felisberto, F. Fdez. -Riverola, and A. Pereira. “A Ubiquitous and Low-Cost Solution for Movement Monitoring and Accident Detection Based on Sensor Fusion”. en. In: *Sensors* 14.5 (May 2014), pp. 8961–8983. doi: 10.3390/s140508961. url: https://www.mdpi.com/1424-8220/14/5/8961 (visited on 01/19/2022) (cit. on pp. 20, 21).