School of Engineering, Discipline of Mechanical Engineering and Mechatronics, pp. 1–73

The effect of supplementary data on acoustic event classification through machine learning

Final Year Project Report - MECH4841 Part B

JUNE 2019

Christopher Caelli [1]

[1] *Student of Mechatronics Engineering,*
The University of Newcastle, Callaghan, NSW 2308, AUSTRALIA
E-mail: Christopher.Caelli@uon.edu.au

Mandatory Dot Point Summary

[FLAGGED FOR REVIEW]Put in a screenshot of the bloody app! As per the FYP Handbook:

I did:

- I studied and prototyped two machine learning solutions to identify if audio data paired with supplementary physical data (GPS, accelerometer, gyroscope) can improve the classification of an acoustic event, as measured by the F1 score.
- I learnt and prototyped a pipeline to record, process, classify, and output data.
- I used my mechatronics degree as a basis for machine learning in this problem. This includes Mechatronics Design (trade off and evaluations from MCHA3000), sound pre-processing, pre-processing data (filtering, analysing), and statistics from MECH2450/MCHA3900.
- I identified and applied appropriate system engineering techniques to elicit requirements from my university and work stakeholders.
- I managed the development and implementation of a proof of concept app to record data as an input for this project. I applied my software engineering skills and project management skills to supervise another employee, create a schema, develop the android app, and present it to the relevant stakeholders.

Executive Summary

The project investigated whether supplementary data (Accelerometer, Gyroscope, GPS) will improve a classifier's ability to classify an " acoustic event " in an acoustic event classification / detection (AEC/D) problem. The motivation behind this was to improve AEC/D without onerous microphone requirements, to enable more widespread commercial use of AEC/D.

Section 2 reviews current research. The AEC/D field is expanding in line with advances in the wider machine learning community. However AEC/D still remains a non-trivial problem with few commercial products, and few real-world applications. Research in this area remains small compared to the wider machine learning community, but is expanding. At the time of publication, the question of the effect supplementary data on audio classification had not been answered.

Section 3 analyses the problem and elicits the design requirements through system engineering principles.

Section 4 details the method of selecting the desired output, the required data input, a metric to score by, a classifier, optimising hyperparameters, and training and validating the data. For this project:

1. The desired output was a label (or labels) detailing the acoustic event in the sample (e.g. The labels found in Appendix G like 'Walking').
2. The required data was the ExtraSensory dataset, or a custom dataset as recorded by the project's App.
3. The score selected was the F1 score. It is an industry standard.
4. The classifier chosen was the MLPClassifier which is a feedforward neural network, trained via backpropagation.
5. Hyperparameters of the MLPClassifier were chosen via a Random Search Cross Validation, against the F1 score.
6. Training and Validation were run with early stopping using a 10% validation set, using 59 out of the 60 ExtraSensory users. A final test is done with the 60$^{\text{th}}$ user.

Section 5 shows the results are a 10.5% and 11.9% relative improvement in F1 score for a binary and multi-label classifier respectively. This was calculated through the analysis of F1 scores of a binary MLPClassifier, and a multi-label MLPClassifier, both trained on the ExtraSensory dataset using the method described in Section 4.

Section 6 evaluates the results and discusses their limitations. It will discuss how multi-sensor AEC/D is effective, why the result will be valid in other projects and the significance of these results in the wider field.

Section 7 concludes and reflects on the report, and expands on the significance of the thesis.

Section 8 will recommend future work including; investigations into new datasets with a greater number of labels, and further detailed analysis into the cause of the result.

Acknowledgements

I'd like to thank my partner Brigid for encouraging and supporting me throughout this thesis.

I'd also like to thank each and every mechatronics Academic staff member at the University of Newcastle. Their tireless efforts made it clear they care about their students, and advocate for the Mechatronics program at every step. I hope to see the degree continue to grow thanks to their efforts.

Contents

1 Introduction 7
- 1.1 Machine Learning Field . . . 7
- 1.2 Classification . . . 8
- 1.3 The Problem . . . 9

2 Review of Literature 10
- 2.1 Acoustic Event Classification and Acoustic Event Detection . . . 10
- 2.2 Detection and Classification of Acoustic Scenes and Events Competitions (DCASE) . . . 11
- 2.3 Leading Research, DCASE2018 Task 4 Winner . . . 11
- 2.4 Summary of Literature Review . . . 12

3 Problem Analysis 13
- 3.1 Current 18 Words . . . 13
- 3.2 Tree Diagram . . . 13
- 3.3 Quad of Aims . . . 14
- 3.4 Input Output Analysis . . . 14
- 3.5 Affinity Diagram . . . 15
- 3.6 Systems Map . . . 16
- 3.7 Sequence Diagram . . . 17
- 3.8 N^2 Analysis . . . 18
- 3.9 Spray Diagram . . . 19
- 3.10 Matrix Diagram . . . 19
- 3.11 Summary of Problem Analysis . . . 20

4 Method 21
- 4.1 Introduction to the Method . . . 21
- 4.2 Required Data . . . 22
- 4.3 Data Collection . . . 22
  - 4.3.1 Project Android App . . . 22
  - 4.3.2 ExtraSensory . . . 25
  - 4.3.3 Project Android App vs ExtraSensory dataset . . . 25
- 4.4 Data Preprocessing . . . 26
- 4.5 Classifier . . . 29
  - 4.5.1 Model Selection . . . 30
- 4.6 Model Validation . . . 31
  - 4.6.1 Hyperparameter Optimisation . . . 34
- 4.7 Method Summary . . . 40

5 Results 41
- 5.1 Binary MLPClassifier . . . 41
- 5.2 Multilabel MLPClassifier . . . 42
- 5.3 Summary of Results . . . 43

6 Discussion 44

7 Conclusion 47

8 Future Work **48**

9 References and Citations **49**

A Full Extracts of Journals **52**

B Reflection **53**

C Full Result Output **55**
C.1 Binary Classifier . . . 55
C.2 Multi-label Classifier . . . 58

D Dataset Feature Importance **67**

E Trade-Review of "Online vs Offline" processing **69**

F GridSearch Attempt **71**

G Table of ExtraSensory Labels **72**

1 Introduction

The mobile phone is an omnipresent force in modern society. The number of unique mobile phone users was 5.1 billion at the end of 2018 [1]. Every phone carries a microphone, and in a data intensive field such as machine learning, this presents an untapped potential. One application of these devices could be to document and understand the environment around them.

1.1 Machine Learning Field

The machine learning field dates back to the first reproduction of a neural network [2] in an attempt to explain the physical decision making process of the nervous system. There were seminal scientific breakthroughs and papers on machine learning from mathematicians and early computer scientists throughout the mid to late 20th century. The advent of cheaper, faster processing power in the last decade has revived the idea of machine learning, and the widespread availability of data has again accelerated this area. Machine learning encompasses any application of a computer learning to complete a task.

Learning Computers Machine learning is based on the concept of having a computer " learn ". This broad term has been used to describe a plethora of operations including fitting a function to historical data, iterative output optimisation based on previous hypotheses, and 'genetic algorithms'[1]. All of these strategies have a common theme; machine learning rewards progress and punishes mistakes. Often this can be reduced down to a cost/constraint model.

A Formal Definition A definition for Machine learning is taken from Tom M. Mitchell's 1997 textbook, " Machine Learning ":

> " A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E [3]. "

This definition of machine learning can be applied to this project: the task T (AEC/D), the E (Audio, IMU, GPS), and P (a F1 score).

Machine Learning from a Mechatronics Perspective Machine learning is the focus of this report. Machine learning is a field of engineering and computer science concerned with automating the discovery and modelling of a process. This process is usually a classification or regression function. In a nutshell, it is about training a computer algorithm to learn from data. It's application in this report has parallels to a traditional optimisation problem in mechatronics: Given a " plant " *(or, a neural network)*, how can someone " optimise " *(train)* the plant's " parameters " *(both weightings and hyperparameters)* as to " minimise an error " *(or, maximise a classification/regression score)*?

Model Based Approach The traditional approach to machine learning has been to fit a function to a collection of historical samples, called a dataset. This fitted function then becomes an estimation of the desired process sought to be learnt. This is called the model.

[1] Genetic Algorithms are an optimisation approach that relies on narrowing a random search, using previous success to shape the bounds of the next random guess

Dataset: The Truth of Classification The process of training a model requires samples of data that have been pre-classified. This classification is known as a " ground truth ". This is what is more broadly known as supervised learning. A key concept for this project is to compare the classifier produced, against the ground truth of the samples.

Signal Processing Signal processing is a field of mathematics focused on the use, interpretation, creation and modification of continuous structures of information across the time domain. The broad definition of a signal could be anything from a wave of light, to a graph of house prices. This project was concerned with the physical phenomenon of vibrations of varying amplitude (sound!).

Audio Signal Processing Audio (or acoustics) signal processing is a sub-field of signal processing. Audio signal processing pre-dates the digitalisation of signal processing, and looks to understand the science behind an audio signal, and the many applications of it. One of its many applications is to understand the content of the audio signal.

Music Audio signal processing has been a key element of technological development in music. The medium of music ranges from ancient human use of musical instruments, to a range of mechanical analogue devices to record and reproduce sound, to a discretised digital signal representing music. All of these methods represent an audio signal, but the advent of the digital signal has allowed for the widespread use of digital signal processing (DSP). There are a range of applications including new musical effects, compression and storage, sound recognition, noise cancellation, and equalisation. The history of music is worth mentioning as it highlights how audio signal processing has been used in this field for quite some time, but only now are we recognising the diverse applications to other industries.

Acoustic Event Audio signals can contain wide range of data. This can span from music to language, but can also capture certain " acoustic events ". This could be any event that is measurable through audio, an example of which could be wheel screeching and metal clashing. In some cases, an event could even be inferred from the audio signal, such as a car crash.

New Opportunities Audio digital signal processing has allowed widespread application of actionable audio data. In the last 3 decades, applications such as speech recognition, speech synthesis, and acoustic pattern detection have matured into commercial products, and see wide spread use in society. These products contribute to the growing audio datasets that enable the opportunity for AEC/D and other machine learning applications.

1.2 Classification

Classification is the process of determining, based on observations, what categories an item belongs in. It is possible for categories to be mutually-exclusive, such as that of the an animal's genus, or it could be non-unique, such as a song's genre. It can also be the process of labelling a signal, such as the type of acoustic event detected in an audio signal.

Classification is a major topic of research within machine learning, and the classification of acoustic events with machine learning is another expanding subset of research.

Current State of Acoustic Event Classification The latest results within the field can be found in the Detection and Classification of Acoustic Scenes and Events (DCASE) competitions. At the time of publication, preliminary results from DCASE 2019's Task A challenge demonstrates an accuracy score of 86.5% [4]. ($\frac{Correctly\ Classified\ Samples}{Total\ Samples}$)

1.3 The Problem

This project aims to answer the question: What is the effect of supplementary data on acoustic event classification through machine learning? This thesis will analyse the difference between two classifiers, one with supplementary data, and one without. Additionally, the output of the classifiers will be analysed for " validity ". This will be determined by whether the classifier can successfully detect and classify an acoustic event.

Motivation for the Problem The capacity to identify a physical event in the real world using sound has many potential implications and applications. The hypothetical applications are limitless: assistance and reality closed captioning for the hearing impaired, improved security closed loop TV, improved ride sharing trip documentation, and early detection of potential accidents or medical emergencies are some examples of how it could span multiple industries. The primary focus of this report is to review AEC/D as a potential new method of debriefing.

Debreifing (or After Action Report) Debriefing is an educational tool for providing performance feedback to teams or an individual. The formal debrief originated from World War 2, with SLA Marshell's "interviews after combat" [5]. Since then, the "After Action Report" was developed by the US Army as ...

> " a professional discussion of an event, focused on performance standards, that enables soldiers to discover for themselves what happened, why it happened and how to sustain strengths and improve on weaknesses [6]. "

Debriefing is a key tool in training after any activity, and has been extended to many industries, most notably in the training of doctors [7].

Traditionally, debriefing is done through a collection of primary and secondary sources[8]. In the information age, these primary sources have expanded to include large datasets, recordings and other digital forensics, as seen in the post-simulation debriefing of a surgeon. This new and increasing range of primary sources provides the potential to significantly improve the steps of debriefing, and ultimately improve debriefing outcomes.

2 Review of Literature

A literature review of the current research was conducted with guidance from University of Queensland's guide to Literature reviews [9]. There are 3 nested areas of research that are of interest to this thesis:

- Classification and Regression techniques
- Machine learning
- The specific field of acoustic event detection and classification (AEC/D)

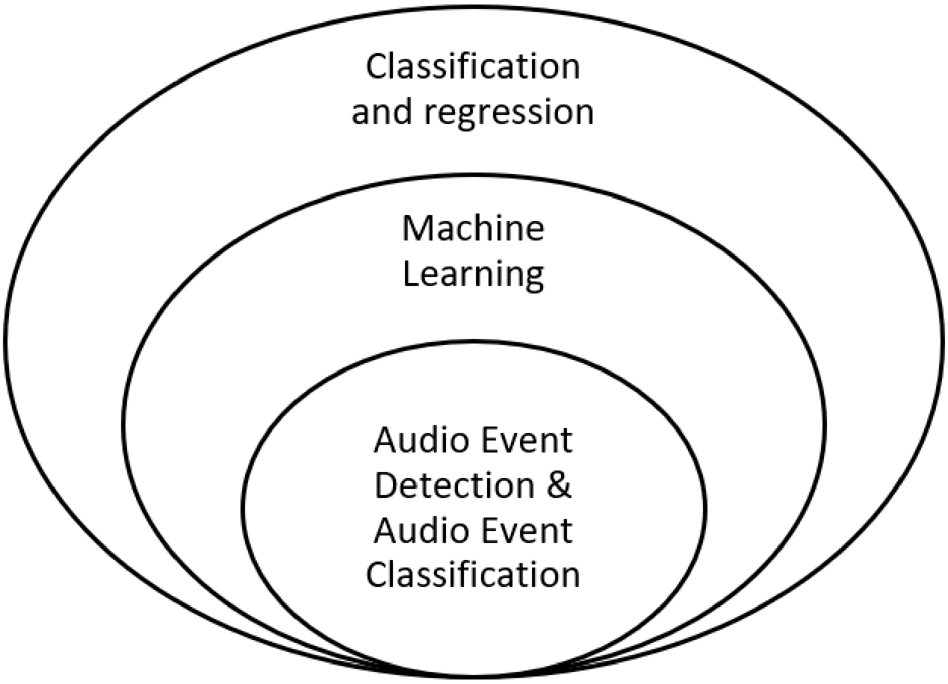

Figure 1: The literature for this project is a niche topic in the machine learning area of research

2.1 Acoustic Event Classification and Acoustic Event Detection

Historically the technologies for AEC/D have been support vector machines (SVMs), Hidden Markov Models (HHMs), and more generally, digital signal processing (DSP) threshold classifiers. In the last 8 years, AEC/D research has been a popular emerging topic of research. Research has focused on the two key tasks: detection (when did an acoustic event occur in the audio sample, and when did it stop?) and classification (what sound occurred?). New research papers are often the result of Detection and Classification of Acoustic Scenes and Events (DCASE) competitions. DCASE is an official Institute of Electrical and Electronics Engineers (IEEE) Audio and Acoustic Signal Processing (AASP) competition. Figure 2 depicts the increasing F1 scores of Toni Heittola's classifiers since 2010. This highlights the technological change the field has undergone.

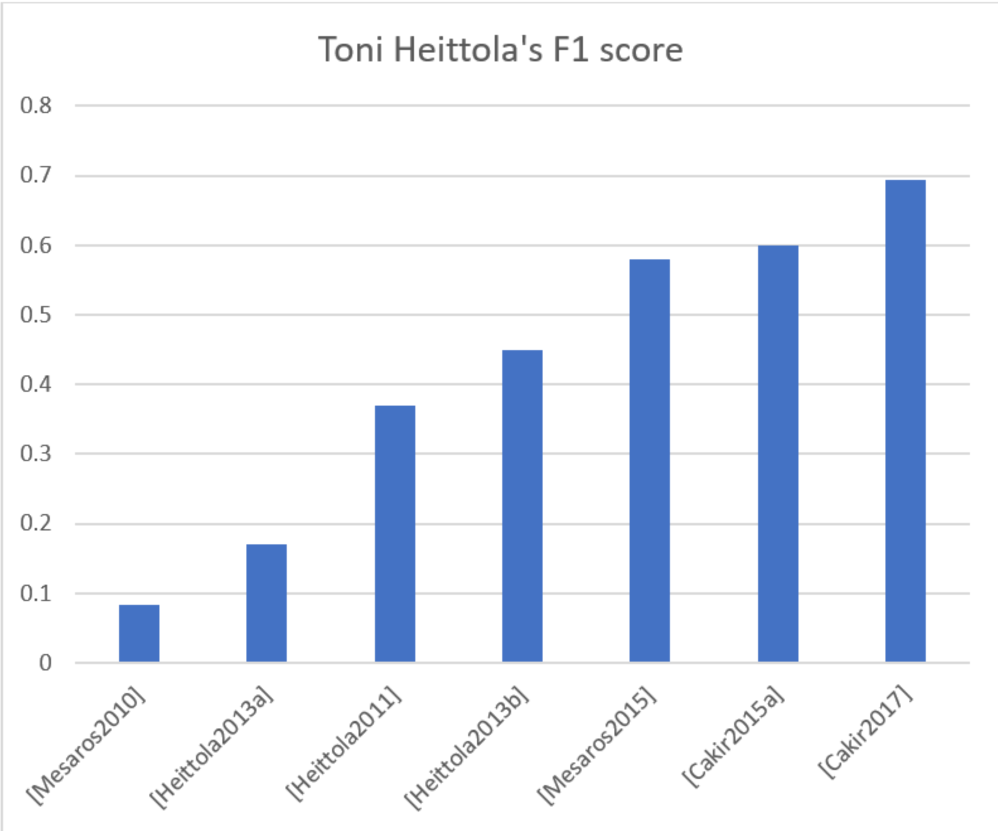

Figure 2: Tampere University of Technology researcher Toni Heittola's F1 results reflect the AEC/D field of research with drastic improvements over the last 8 years [10].

2.2 Detection and Classification of Acoustic Scenes and Events Competitions (DCASE)

The DCASE competitions started in 2013, had its second competition in 2016. It has since become a yearly event. Since 2017, there has been a new format where the competition starts in March, formal results are released by September, and in November a workshop for participants regarding the best results from that year [11]. 2016 was the first to feature machine learning (winning teams incorporated machine learning into ensemble model classifiers). By 2017 all entries utilised machine learning. The results from DCASE2018 indicate that progress in the AEC/D field had allowed more sophisticated, real-world applications to be evaluated. This includes DCASE2018 Challenge Task #4 [12]. This task focused on using a subset of Google's AudioSet to determine the audio class and the timestamps of when that class occurred within a sample. The complexity of the dataset, the number of acoustic classes, and fidelity of output were unprecedented in the AEC field. The winning results for this level of complexity did not score as high a F1 score compared to the DCASE2017 competition (2017 had a 41.7% F1 score [13] vs 2018's 32.4% [12]). DCASE2017 featured more controlled environments/datasets across the 4 tasks which likely resulted in the disparity. However DCASE2018 that presented the best precedent for the problem addressed in this thesis.

2.3 Leading Research, DCASE2018 Task 4 Winner

The winning model used in the DCASE2018 Task 4 challenge used a the following features:

1. "Mean-teacher" model for classification
2. a convolutional neural network (CNN) for context gating[2][14]
3. a bidirectional recurrent neural network (RNN) to improve the utilisation of unlabelled, unbalanced training datasets [15].

2.4 Summary of Literature Review

Previously, AEC/D research has not utilised machine learning, as can be seen in the music industry. It is only recently with competitions like DCASE that machine learning has started to be incorporated into the analysis of AED/C. Even in these situations the datasets used are often unique and not widely applicable. At the time of submission, no papers have been identified that discuss the use of audio with physical sensor data, and as yet the question of the effect of supplementary data on audio classification has not been answered.

[2]a pre-classifier step to improve flaws in training methodologies for some machine learning models [14]

3 Problem Analysis

This project approached the problem through a system engineering perspective, a process which "is a structured and systematic methodology providing greater visibility and control over . . . new system[s]" [16]. This perspective was informed by ISO/IEC/IEEE 15288 [17] and Burge Hughes Walsh's system engineering Toolbox [18]. The tools adapted and applied throughout this section are primarily derived from this " System Engineering Toolbox ".

The purpose of Problem Analysis is to define the scope of this project to be achievable, measurable and practical to implement. This took 2-4 weeks through June 2018, and remains an ongoing task.

The following system engineering tools have been adapted and implemented to develop this paper and the author's understanding of the problem.

3.1 Current 18 Words

A tool called " 18 Words " was used to constantly refine and maintain a description on the scope as it changed throughout problem analysis, with the intent to succinctly describe the project in " close to 18 words ". The current "18 Words" is the following:
"The project aims to create a portable, trackfile recording system. This is coupled with offline processing to record specific physical events that occurred during recording."

3.2 Tree Diagram

With the understanding of what the project was intended to be, a Tree Diagram was drafted to explore missed requirements, hidden modules and other aspects of the project not yet considered. Figure 3 shows a Tree Diagram breakdown for this FYP.

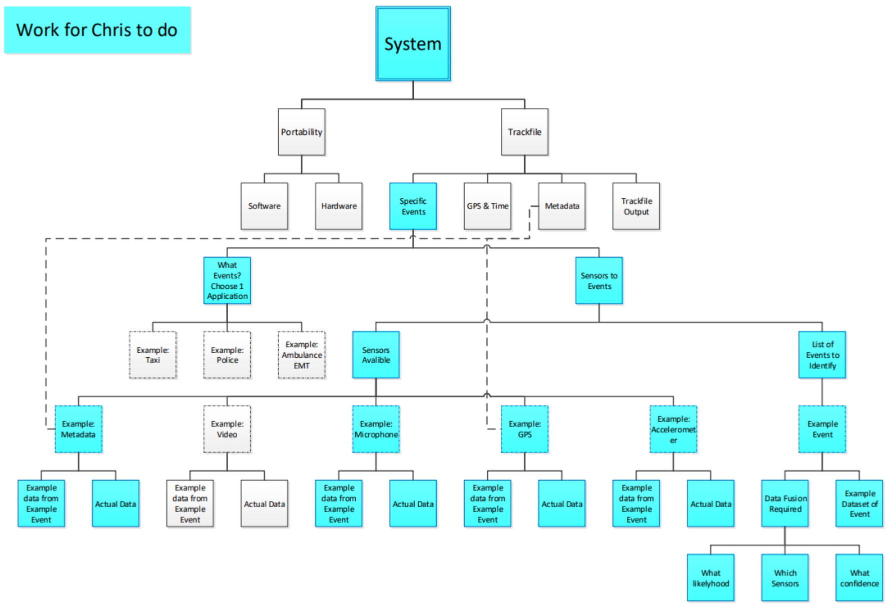

Figure 3: A tree diagram for the project, used as a method of allocating work

3.3 Quad of Aims

The Quad of Aims is a tool used to explore 4 critical, high level aspects of the project as explained in Table 1, and shown in Figure 4.

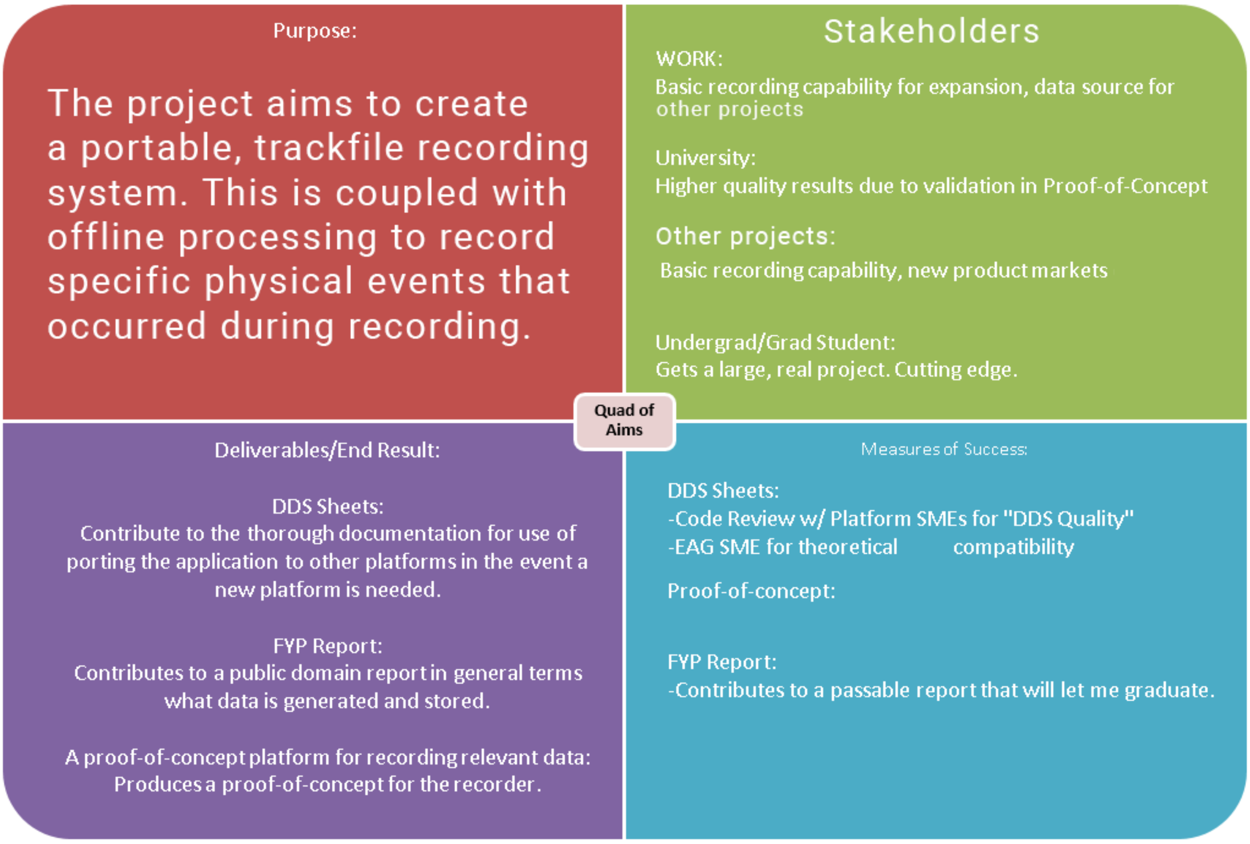

Figure 4: A Quad of Aims for the project might be A3 in size and would have the relevant information embedded. It may also be completed on a whiteboard.

Table 1: Quad of Aims

Label	Description
Purpose	This is our "18 words"
Stakeholders	University, Author's Work
Deliverables	Documentation, recommendations, FYP report, proof-of-concept
Measure of Success	Review of Documentation by SMEs, review of FYP report, dry and wet run of proof-of-concept

This was done to evaluate any early risks, and the begin scope reduction on the project.

3.4 Input Output Analysis

The input output analysis of the system informed the bounds and requirements to operate the system. In this situation, it helped define the full scope of the project; including the technical and non-technical aspects of undertaking the MECH4841 Project as shown in Figure 5.

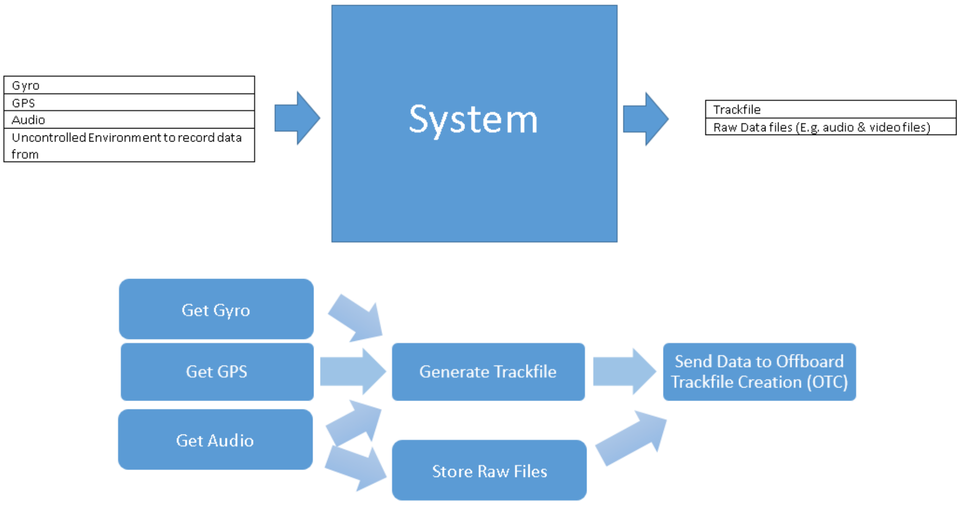

Figure 5: The input output analysis for the system

3.5 Affinity Diagram

An Affinity Diagram helped to add detail to the high level concepts in the project. There was a focus on putting tangible measurements onto requirements. During this process, system architecture decisions were made such as a trade review into online vs offline processing Appendix E, and subsequently splitting the system into smaller modules. The primary system had 2 key sub-systems: the recording system, and the offboard processing systems. An Affinity Diagram of the top-level system was created, and and this informed the creation of 2 affinity diagrams of the sub-systems. After this was complete, previous work was updated to reflect these changes. Figure 6.

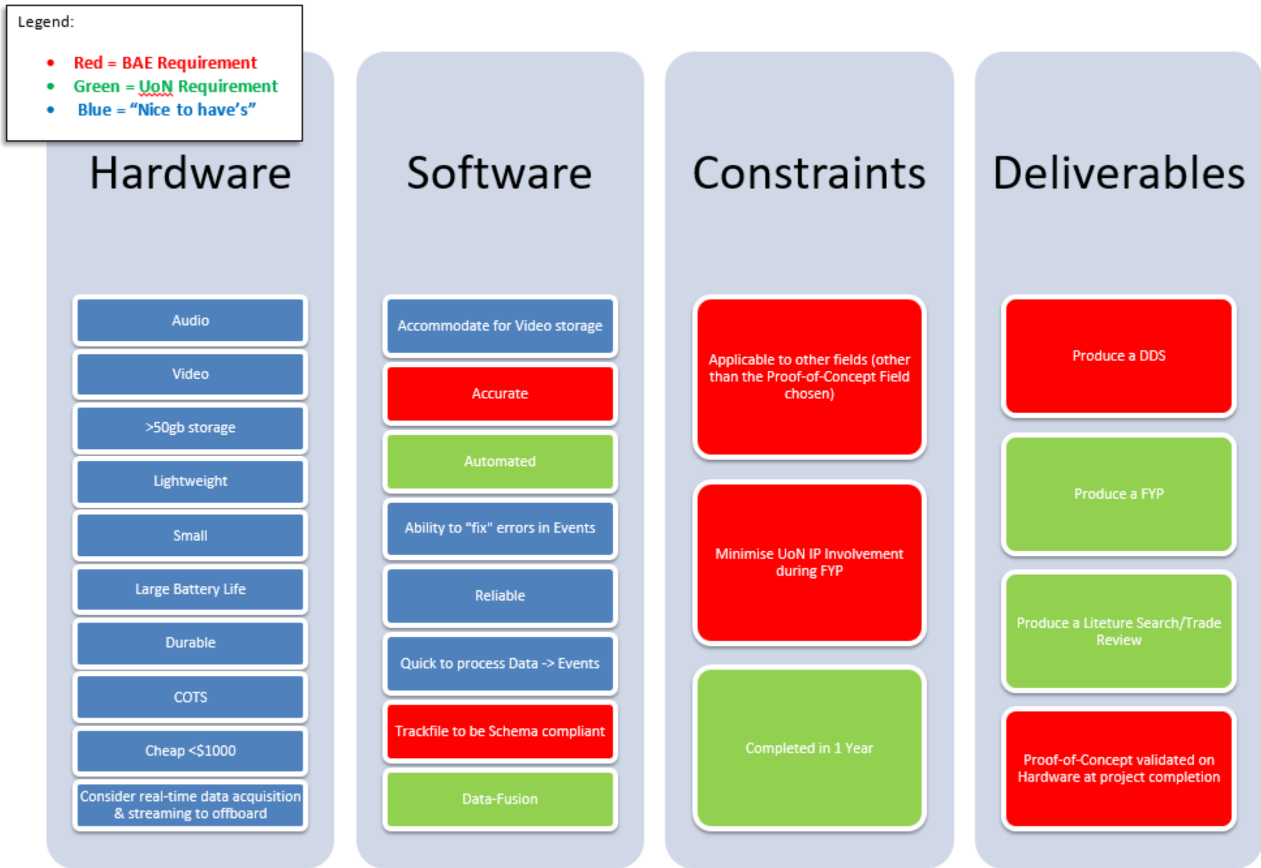

Figure 6: The Affinity Diagrams for the full system architecture

3.6 Systems Map

A systems map uses the affinity diagram, input output analysis, and tree-diagram to identify the processes inside sub-modules that are needed to design the system. This was used to measure and estimate the workload necessary to implement data fusion alongside machine learning of acoustic classification as shown in Figure 7.

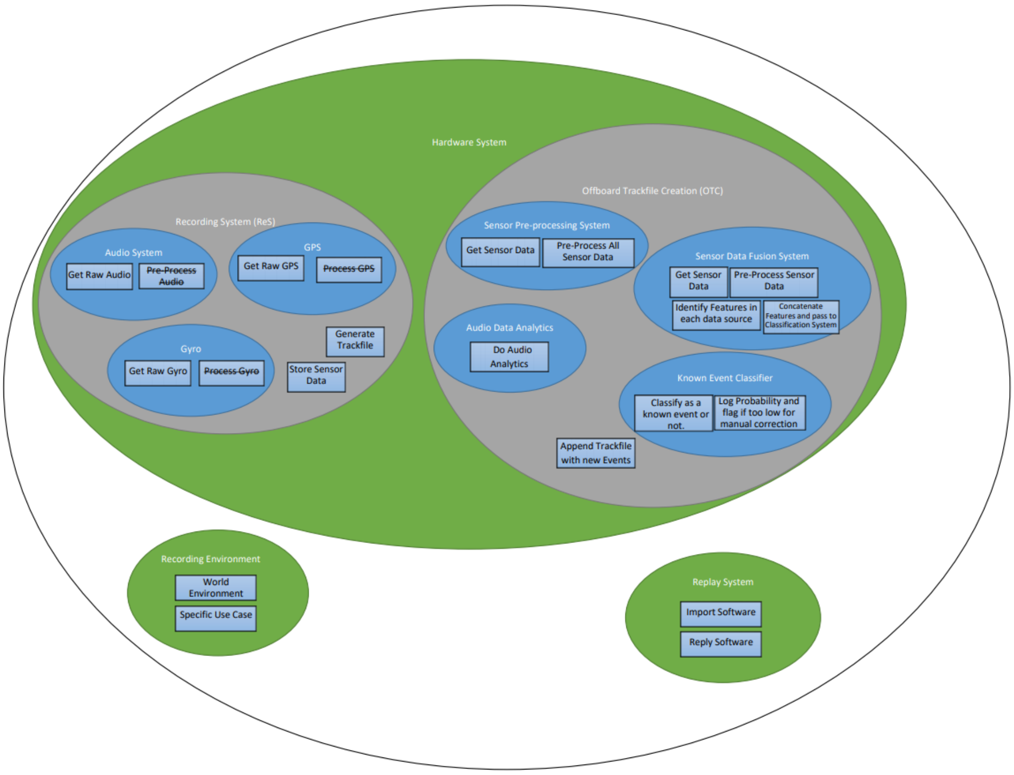

Figure 7: A systems map for the project

3.7 Sequence Diagram

A Sequence Diagram was developed to analyse the flow of data through the processes identified in the System Map. This is shown in Figure 8. This helped to predict and manage any potential complexities and logistics related to the specific needs of each process. Originally, This Sequence Diagram was used to justify removing data fusion from the scope due to the large workload required to implement alongside machine learning. Later in the project, it was brought back into the scope of the project to help compare and evaluate its effectiveness in improving the overall system.

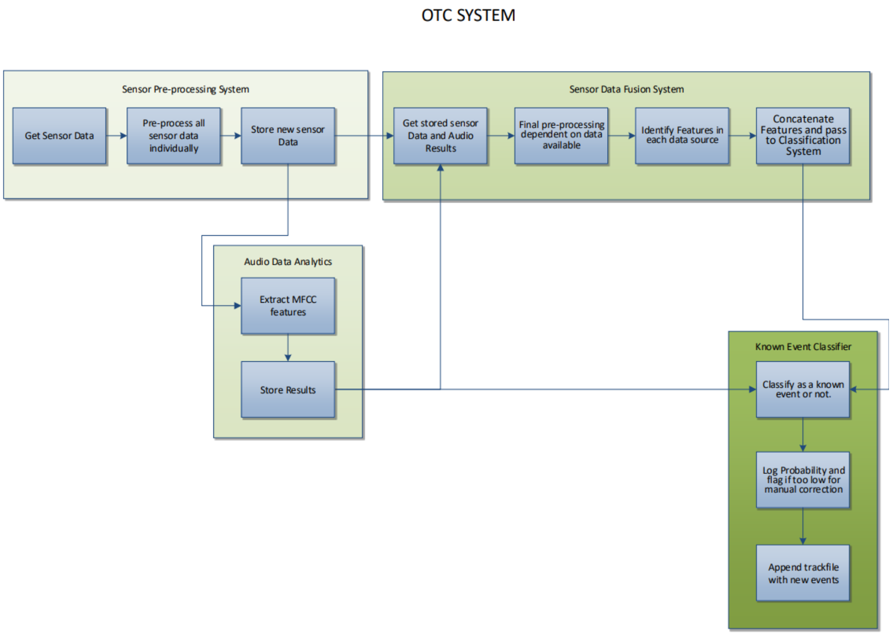

Figure 8: A basic Sequence Diagram for the project

3.8 N^2 Analysis

A N^2 analysis methodically expands on the data flow within the system. This complimented the complexities highlighted in the sequence diagram by documenting what data was expected. A example of this is shown in Figure 9

Sensor Pre-processing	Audio pre-processing gets passed onto the Audio Analytics module	Sensors Pre-processing gets passed onto the sensor data fusion module		
	Audio Data Analytics	Processed audio gets passed onto the sensor data fusion system Feedback Loop	Audio get passed onto the known event classifier	
	Possibility of training data coming back from Known Event Classifier helping train the audio data analytics Feedback Loop	Sensor Data Fusion System	Fused data gets passed into the known event classifier	
Lessons learnt should inform better sensor pre-processing design	Possibility of training data coming back from Known Event Classifier helping train the audio data analytics		Known Event Classifier	

Figure 9: A N^2 Analysis for the project

3.9 Spray Diagram

The spray diagram shown in Figure 10 shows how details of the system could have multiple effects in design requirements, outcomes and operational use. Of interest in the diagram was the reciprocal relationship between high-quality output and high-quality input. The requirement of a high-quality output indicated the potential increase of production and design costs due to the necessitation of a high-quality input.

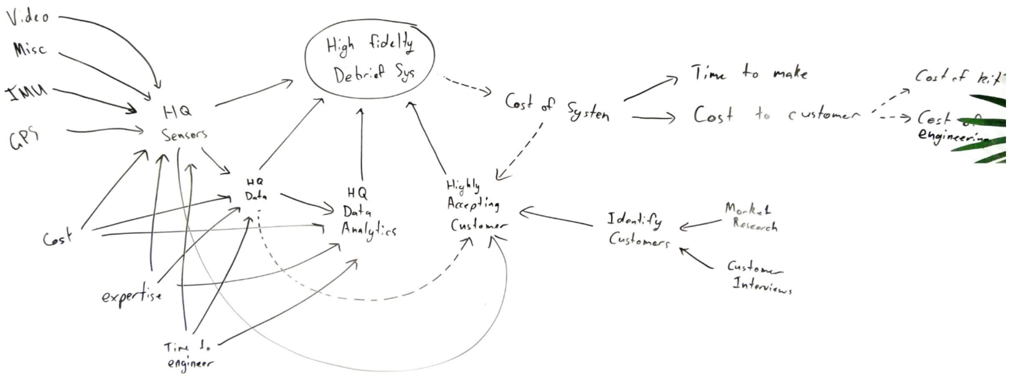

Figure 10: A Spray Diagram for the project

3.10 Matrix Diagram

A matrix diagram was made to review reduced scope as shown in Figure 11. The numbers indicate " packages of work ". By using a "strong", "weak" or "none" indicator for each aspect of the project, each package of work was evaluated against it's likelihood to fulfil to the requirements of the project. It highlighted the difficulty in balancing the needs of both major stakeholders.

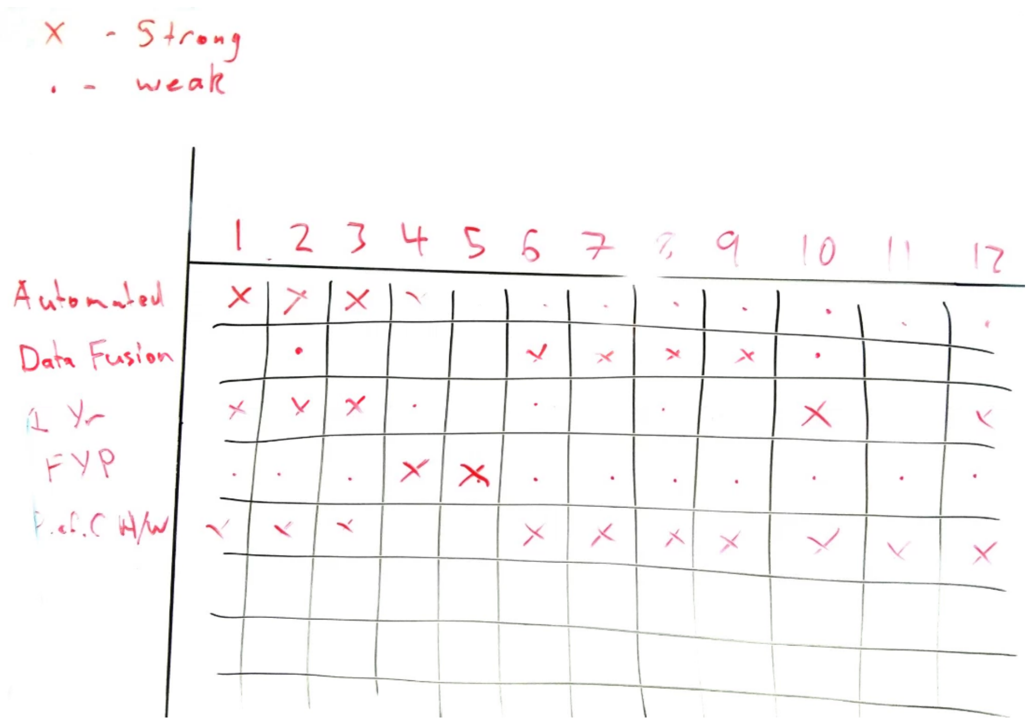

Figure 11: A matrix diagram for the project

3.11 Summary of Problem Analysis

This section described the motivation for the project was applied AEC/D and this required scope reduction to ensure it was feasible as a Final Year Project. It also included a proof-of-concept machine learning solution within the scope. This should also provide an excellent learning opportunity for the author.

4 Method

4.1 Introduction to the Method

In a traditional machine learning problem there are 5 stages to the method:

1. Data
2. Preprocessing
3. Model
4. Fitting/Training
5. Evaluation

Stage 1 Stage 1 is choosing what raw data is available and what information the model must produce. These are the inputs and the outputs that is wanted from the system, and the expected result.

Stage 2 Stage 2 is preprocessing. The purpose of preprocessing is to transform the available data into a format most appropriate for the intended analysis. This is inclusive of processes such as optimising, filtering, feature building[3] and many other options. Preliminary features are also evaluated in this stage to ascertain their quality and effectiveness in a preliminary model.

Stage 3 Stage 3 is selecting the model to be used. The selection of the model involves an understanding of what is best practice in the chosen topic area, a review of recent publications and their methods, and investigating if the chosen topic necessitates any special requirements. Once a model is elected, the parameters of the model must be determined (or evaluated) before a final model can be selected. This is done in the following two stages.

Stage 4 Stage 4 is fitting the model. The selected algorithm may need to be trained or fitted. This is the process of taking labelled input / output data (denoted as X, y) and supervising the model as it attempts to reproduce y from X. A common method to achieve this is through Stochastic Gradient Descent (SGD) and back propagation [19]. In summary, it is the process of changing the neuron weights proportionally to their contribution to the error. Equation 4.1 adapted from Lecun(2012) demonstrates this, where W is the machine learning weights for a given topology, η is the proportional factor or step size, and the partial represents how each weight contributes to error [19].

$$W(t) = W(t-1) - \eta \frac{\partial E}{\partial W} \tag{4.1}$$

Once trained on a training set, the next stage is validating whether the model is sufficiently fit.

Stage 5 Stage 5 is validation. Validation is the process of checking the performance of the model against certain criteria. For instance, a model may be checked for overfitting, underfitting, bias, sensitivity, specificity, accuracy, precision, AUC (area under the (ROC) curve), and F1 score. F1 score was used as the primary measure in this project because the F1 has been used as a research industry

[3] Producing refined data of interest to the model. For example, Mel-frequency cepstral coefficients (MFCCs) from raw Audio

standard for evaluating algorithms.

It is beneficial to apply a methodical approach when implementing machine learning. As each section will significantly affect the next section, care should be taken to optimise each stage before progressing in implementation. External to the method of building the machine learning classifier is the work involved in developing the pipeline associated with it. The main tasks to implement a machine learning classifier as a part of a larger solution are as follows:

- Develop the use-case, and identify the output
- Investigate what data is available, and whether further data/datasets would be beneficial
- Choose a framework/technology to implement the machine learning algorithm
- Build the machine learning classifier as described above, and test its effectiveness
- Validate the whole system by using the classifier in the pipeline

4.2 Required Data

The machine learning solution required GPS, accelerometer, gyroscope, and audio. A Google Pixel 2 was selected as the target data acquisition unit. Table 2 below shows the Pixel 2 sensors associated with the required data.

Table 2: Data Available [20]

Sensor	Description
GPS	The Pixel 2 uses the Snapdragon 835 System on Chip GPS receiver
Accelerometer / Gyroscope	The Pixel 2 uses a Bosch BMI160
Audio	The Pixel 2 uses a Qualcom WCD9395 Audio Codec chip and a NXP TFA9891UK Audio Amplifier

4.3 Data Collection

The decision of what data to collect and utilise was identified through the problem analysis in Section 3. As the goal was to maximise the accuracy of the classifier, the data chosen for use and collection was GPS, accelerometer, gyroscope, and audio. As described in the problem analysis phase, the best platform for this would be a mobile phone as shown in Section 3 and Appendix E.

4.3.1 Project Android App

Android was chosen as the platform as it had significant support for reading from the on-board sensors. The data available from the Android API can be seen in Table 3. This was the source for accelerometer, gyroscope and GPS data. It was also selected due to its support for C/C++ through the Native Development Kit (NDK). The programming language C is required knowledge for many courses in the University of Newcastle Bachelor of Mechatronics degree.

The app was developed over five weeks. It was able to record audio to a .wav file and record raw GPS data[4], raw acceleration, and raw gyroscopic data. An example of this is shown in Listing 1. The

[4] Timestamps, latitude, longitude and altitude are recorded from the GPS sensor

Table 3: Data Available[21]

Sensor	Description
TYPE ACCELEROMETER	Acceleration force along the x axis (including gravity).
	Acceleration force along the y axis (including gravity).
	Acceleration force along the z axis (including gravity).
TYPE ACCELEROMETER UNCALIBRATED	Measured acceleration along the X axis without any bias compensation.
	Measured acceleration along the Y axis without any bias compensation.
	Measured acceleration along the Z axis without any bias compensation.
	Measured acceleration along the X axis with estimated bias compensation.
	Measured acceleration along the Y axis with estimated bias compensation.
	Measured acceleration along the Z axis with estimated bias compensation.
TYPE GRAVITY	Force of gravity along the x axis.
	Force of gravity along the y axis.
	Force of gravity along the z axis.
TYPE GYROSCOPE	Rate of rotation around the x axis.
	Rate of rotation around the y axis.
	Rate of rotation around the z axis.
TYPE GYROSCOPE UNCALIBRATED	Rate of rotation (without drift compensation) around the x axis.
	Rate of rotation (without drift compensation) around the y axis.
	Rate of rotation (without drift compensation) around the z axis.
	Estimated drift around the x axis.
	Estimated drift around the y axis.
	Estimated drift around the z axis.
TYPE LINEAR ACCELERATION	Acceleration force along the x axis (excluding gravity).
	Acceleration force along the y axis (excluding gravity).
	Acceleration force along the z axis (excluding gravity).
TYPE ROTATION VECTOR	Rotation vector component along the x axis (x sin(o/2)).
	Rotation vector component along the y axis (y sin(o/2)).
	Rotation vector component along the z axis (z sin(o/2)).
	Scalar component of the rotation vector ((cos(o/2)).1
TYPE SIGNIFICANT MOTION	N/A
TYPE STEP COUNTER	Number of steps taken by the user since the last reboot while the sensor was acti
TYPE STEP DETECTOR	N/A

output was an audio .wav file, and a Comma Separated Value (csv) file, as shown in Table 4. Figure 12 shows a screenshot of the app made.

Figure 12: A screenshot of the app produced in this project

During this time, an Atlassian JIRA environment was set up to facilitate management (using concepts from ENG3500 - Project Management). The Atlassian “ Git-Flow ” methodology was also learnt and used.

I assisted with an estimated 30% of the overall programming, and a code review of the product along with other colleagues. The majority of development work was attributed to an intern I was supervising at my place of employment.

Listing 1: Code snippet from [21] for getting Accelerometer data from the Android System.

```
public void onSensorChanged(SensorEvent event) {
    // Get X,Y,Z values
    accX = event.values[0];
    accY = event.values[1];
    accZ = event.values[2];

    ...
 }
```

Table 4: An example of data recorded from the App

id	attr_time	time	attr_x	attr_y	attr_z
1	1547182324433	2018-01-11 14:52	-0.229843	3.1795	8.52336
2	1547182324442	2018-01-11 14:52	-0.257614	2.93185	8.61444
3	1547182324451	2018-01-11 14:52	-0.179626	2.81464	8.81348
4	1547182324467	2018-01-11 14:52	-0.118927	2.82674	8.70142
5	1547182324498	2018-01-11 14:52	-0.191544	2.95287	8.66815

4.3.2 ExtraSensory

The ExtraSensory dataset contains over 300,000 samples across 60 Users (in a CSV format with 278 feature columns), as shown in Table 6. Each row contains an a 'snapshot' of the sensors at a sampling rate of $\frac{1}{60}$ Hz. There is a lack of continuity within each user's CSV. The dataset is unbalanced, and has had its labels cleaned by the researchers involved. Further details about the users themselves can be found in Table 5, or in Vaizman (2017) [22].

Table 5: ExtraSensory Data Breakdown [22]

	Range	Average	(standard deviation)
Age	18-42	24.7	-5.6
Height	145-188	171	-9
Weight	50-93	66	-11
Body	18-32	23	
Labeled	685-9,706	5,139	-2,332
Additional	2-6,218	1,150	-1,246
Average	1.1-9.7	3.8	-1.4
Days	2.9-28.1	7.6	-3.2

4.3.3 Project Android App vs ExtraSensory dataset

Table 4 shows a portion of the Comma-Separated Values (csv) data produced from the app. The id is a unique count, the attr_time is a timestamp from the Android system, and attr_x, attr_y, and attr_z columns show measured acceleration.

Table 6 shows a section of the csv data from the ExtraSensory dataset.

Both datasets share the same timestamp format. A " time " column has been added to the table for illustration purposes only.

ExtraSensory Dataset The ExtraSensory dataset is compiled from 60 users with a diverse range of phones, with " 34 iPhone users, and 26 Android users. " [22]. A further breakdown can be found in Table 7. This is in contrast to this project's phone range, which was limited to 2 Android devices, a Samsung S6 and a Lenovo Zuk Z2. The Samsung uses a InvenSense MPU-6500, the Lenovo Zuk Z2 unknown. A section of data can be found in Table 6. Further details can be found on page 27.

Table 6: Example of data from the ExtraSensory dataset

timestamp	Time	raw_acc:mean	raw_acc:std	raw_acc:moment3	raw_acc:moment4
1464129912	2016-05-24 22:45	1.011438	0.012573	0.023013	0.04124
1464129950	2016-05-24 22:45	1.011233	0.009356	-0.005622	0.016687
1464130031	2016-05-24 22:47	1.013422	0.018068	-0.008593	0.039286
1464130109	2016-05-24 22:48	1.014891	0.0164	0.021383	0.038825
1464130130	2016-05-24 22:48	1.017487	0.022632	-0.012891	0.037226

Table 7: Mobile Phone types used in the ExtraSensory dataset

Device Type								
iPhone generations	4	4S	5	5S	5C	6	6S	
Android devices	Samsung	Nexus	HTC	moto G	LG	Motorola	One Plus One	Sony

4.4 Data Preprocessing

In the machine learning pipeline for use with the Android app, the data pre-processing occurred off-board from the app. This preprocessing was done using Python. Preprocessing the data occurs before the machine learning modules of code, and Matlab was also used to prototype the algorithms.

Audio Preprocessing of the audio data was as follows:
The recorded .wav files were split into smaller sections via the sliding window method[5]. Each window then has the Fourier transform applied to it (as shown in Equation 4.2). These windows are then mapped to the Mel Log Scale (as shown in Listing 2). This in turn is both " Feature Engineering ", as well as preprocessing to be compatible with training the model.

$$WindowSample(t) \xrightarrow{\mathscr{F}} WindowSample(\omega) \tag{4.2}$$

Listing 2: Matlab trial of MFCC extraction

```
[sample,Fs] = audioread('C:/FYP/report/Code/Audio recording 2019-01-11 16-26-00.wav');
Freq = 44100 % Hz, https://developer.android.com/reference/android/media/AudioFormat.
    html
[coeffs,delta,deltaDelta,loc] = mfcc(sample,Freq) % Audio Toolbox: https://www.
    mathworks.com/help/audio/ref/mfcc.html
```

For the ExtraSensory dataset, this had already been processed.

GPS The Global Positioning System (GPS) is a constellation of geosynchronous satellites. Each satellite emits a timestamp. When a GPS receiver receives multiple timestamps, it can triangulate its

[5]This is a method of sampling a data by using a fixed sample time, and " sliding " the bounds of the window across the sample. It discretises a continuous sample by known window length and overlap parameters.

position. The mobile phone in this project is the GPS receiver. A GPS signal is requested by the " Fused Location Provider ", in the Google Play services location API[6] and comes heavily pre-filtered. Further filtering is possible by a Kalman filter with the other physical sensors.

IMU For the purpose of this project, the data collected through the Android App had a Low Pass Filter applied to it (however was not used). An example of this filtering is shown in Figure 13, and the code used in Listing 3. A Kalman Filter was considered, but not progressed when the ExtraSensory dataset was selected for use.

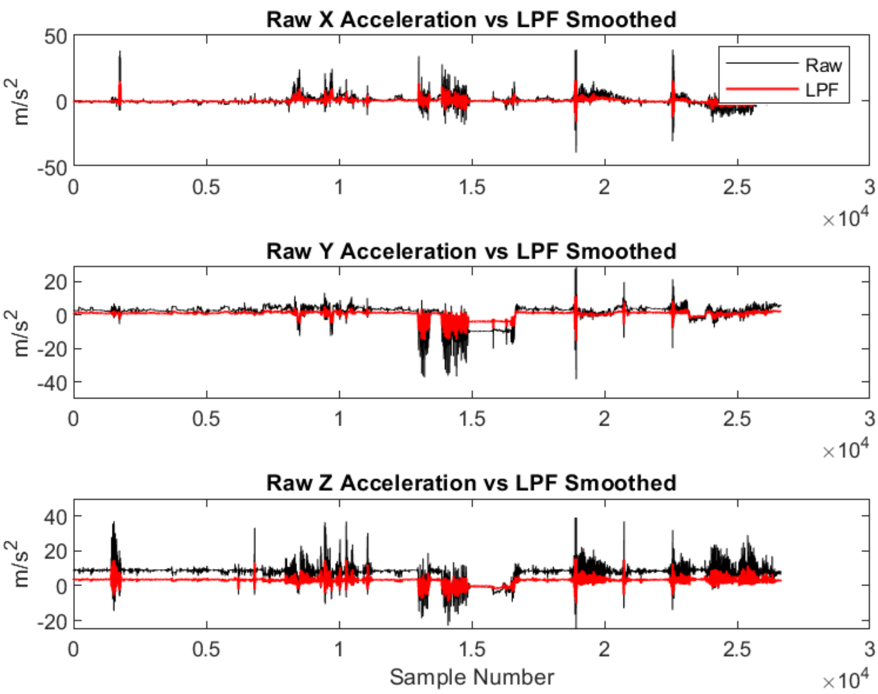

Figure 13: Results of Low Pass Filter vs Raw accelerometer data

Data Training, Testing and Evaluation Set The ExtraSensory dataset[7] was considered for use in 2 ways: training on a per user instance, or training on a per sample instance. The benefits of per user was to isolate recording hardware, and other plant noise. However this was undesirable, as this could result in a low variance in ϵ (bias), which increases the likelihood of the learnt function ($\hat{f}$) to overfit the signal of the training users ($\hat{f} = f(x) + \epsilon$). This was the benefit of instead training on a per sample basis. By introducing other user noise into a training set, overfitting was minimised " for free ". Finally, as a " kfold " (or other cross-validation methods) operation is applied to the whole training dataset for training and validation purposes, the likelihood of the random selection incorporating a larger variance of noise increases. Subsequently, the data is used on as a per sample basis.

[6] Google's closed source " Fused Location Provider " is chosen over the Android Platform's location API for performance reasons. The Android Platform is a part of the Android open source project. One could replicate this using the open source equivalent but one would need to consider refining the location data themselves; perhaps with a Kalman Filter?

[7] The same approach was considered for the Android app

Listing 3: Smoothing of Acceleration data

```
importURL = "C:\FYP\report\Code\2019-01-11T16-26-00_androidAccData.csv";
AccData = csvread(importURL,1,1);
SmoothedAccData = zeros(size(AccData,1),4);
t = 1:size(AccData,1);
lowpassAlpha = 0.40; % 80%
for i = 1:size(AccData,1)
   for j = 2:4
    % Low Pass Filter on
    SmoothedAccData(i,j) = SmoothedAccData(i,j) ...
        + lowpassAlpha * (AccData(i,j) - SmoothedAccData(i,j));
   end
end
for plotNum = 1:3
    subplot(3,1,plotNum);
    plot(t,AccData(:,plotNum+1),'k');
    hold on
    plot(t,SmoothedAccData(:,plotNum+1), 'r', 'LineWidth',1);
    ylabel('m/s^2');
    switch(plotNum)
        case(1)
            title('Raw X Acceleration vs LPF Smoothed');
            legend('Raw', 'LPF');
        case(2)
            title('Raw Y Acceleration vs LPF Smoothed');
        case(3)
            title('Raw Z Acceleration vs LPF Smoothed');
            xlabel('Sample Number');
    end
end
set(gcf,'color','white')
saveas(gcf,'AndroidDataExample.png')
```

“ kfold ” Cross validation James (2017) describes kfold cross validation as:

> “ kFold involves randomly dividing the set of observations into k groups, or folds, of approximately equal size. The first fold is treated as a validation set, and the method is fit on the remaining k - 1 folds. [23] ”

A key element expanding on this is the ability to cycle through ‘k folds’, changing which group is the validation, or test set, and training on the rest [8]. This was important for this project as it maximises training data whilst minimising the potential to overfit.

Balanced vs Unbalanced The meaning of a “ balanced ” vs “ unbalanced ” dataset is whether positive samples are “ balanced ” by an equal amount of negative samples. In a hypothetical scenario, this might equate to an equal amount of benign vs malignant cancers in a cancer classification dataset. The reason why a balanced dataset is important is to minimise any bias in the network. If a majority of cases are benign, then a bias towards all cases being benign would be a valid strategy, but not a

[8] kFold implementation occurs within the RandomCV, as discussed on page 34

desired one.

The ExtraSensory dataset contains positive samples across 51 classes. These 51 classes do not occur at the same frequency as each other. In any given sample, one would not expect every class to be equally represented. To minimise bias, the Python library " imblearn " was used to resample the dataset, as shown in Listing 4.

Listing 4: Code for resampling of the ExtraSensory dataset

```
sampling_strategy = 1.0; #Ratio of resample occurance
RanUSampler = RandomUnderSampler(sampling_strategy=sampling_strategy); #Define the
     Resampling
X_res, y_res = RanUSampler.fit_resample(X_original,  y_original); #Resample via
    Random Under Sampling
resampleIndices = RanUSampler.sample_indices_ #Get the indicides that correspond
    to the new undersampled, more balanced dataset
M_res = M[resampleIndices]; #duplicate this sample to our validity matrix
timestamps_res = timestamps[resampleIndices]; #and to our timestamps
```

4.5 Classifier

A classifier by definition classifies (or, categorises) things. A more thorough understanding is that a classifier takes some input data and maps it to an output category, or label. In a machine learning context, this mapping is done via the $\hat{f}$ learnt function. For this project the classifier is a learnt function that takes audio, accelerometer, gyroscope and GPS data and maps that data to a label one of the labels found in Appendix G, Table 11.

Binary, Multi-class, and Multi-Label classifiers If a classifier is to map data to a label, it stands to reason that the label must be well defined. A binary classifier as the name suggests will categorise input as 1, or 0 against a label. An example of a result could be " Car " or " Not Car ", but it can also be extended to multiple classes, for example " Not Car ", " Holden ", " Ford " etc. This can be done via an ensemble of binary classifiers (combining the output of a " Ford " classifier and a " Holden " classifier, that have both been parsed the same sample etc.), or more recently through a " One Hot " [24] encoded output array as seen in Table 8. A key element of multi-class classifiers are that the classes are mutually exclusive. The car brands in Table 8 are mutually exclusive.

Table 8: Example Hot One Encoding

	Holden	Ford	Other
Ford Fiesta	0	1	0
VW Golf	0	0	1
Holden Captiva	1	0	0

There is another type of classifier called the multi-label classifier. These labels are not always mutually exclusive and can occur at the same time. For example, talking and eating would occur in the same setting at a restaurant. This thesis requires a multi-label classifier as it aims to label whether the signals provided in the data represent different acoustic events such as jogging, walking, laughing, computer work, etc as found in Table 11.

For this project, a binary classifier and a multi-label classifier were used and evaluated. The reason for this is because a binary classifier is a valid and succinct way to answer the intent of the project, which was to answer the question does supplementary data assist in an acoustic event detection and classification (AEC/D). However for most 'realistic' use cases, the validity of the result is important. For this reason, a multi-label classifier would be an appropriate classifier to test with as well.

The methodology section was used to create a multi-label variant of the network. This used the same hyperparameter optimisation as the binary classifier and is discussed in the results.

4.5.1 Model Selection

Model selection is an important aspect of machine learning because of the " No Free Lunch " theorem, described by Magdon (2000) as:

> " No Free Lunch theorems have shown that learning algorithms cannot be universally good. [25] "

This theorem shows that selection of a model is non-trivial, and the use case must be factored in.

The model required for the project will have to take into consideration several factors; the type, format and quality of data in use, the type and format of the desired information, as well as some unique factors. There is a desire to investigate applicability for onboard processing and a desire to have the methodology and results generalise well (across different use cases).

The standard approach is to either a) try many model options or b) investigate current best practice in academia. Below several models have been reviewed.

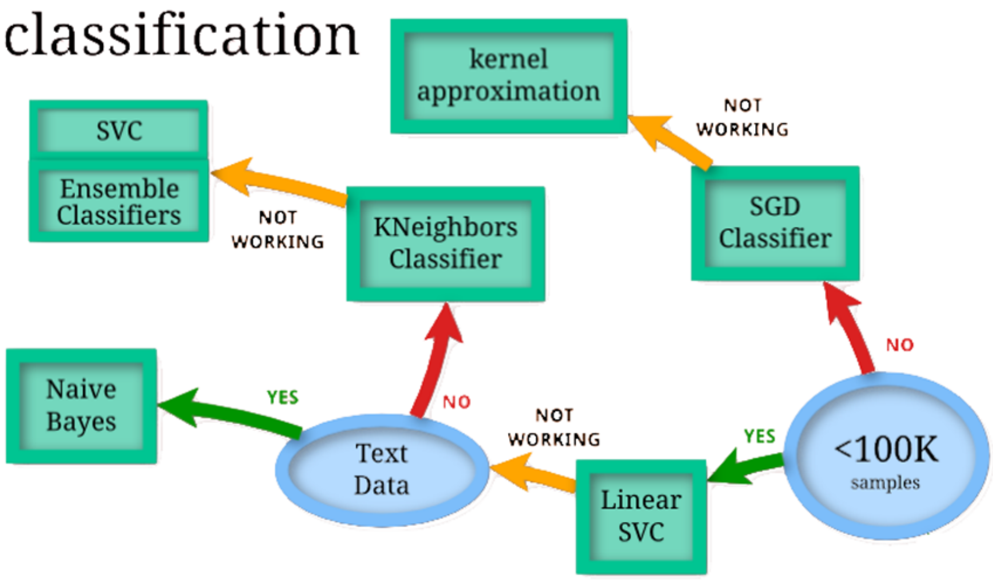

Figure 14: Scikit-learn's suggested model selection process[26]

Convolutional Neural Networks Convolutional neural networks (CNN) were primarily designed for, and received the best results for image-based classification problems. This is because the architecture has been designed for computer vision. This would have be an excellent choice if audio spectrograms were to be used.

Recursive Neural Networks Recursive neural networks (RNN) are not well suited to tabular data. This is because RNNs (and deeper learning methods) utilise the Manifold Hypothesis. This suggests that very high dimensional data can have features represented on a low dimensional manifold. As this manifold is much lower in dimension (and consequently requires computation power), this is what deep learning models attempt to learn and map to. This was tested in Fefferman (2016) [27]. As such, this approach is not best suited to the small dimensionality of the preprocessed, tabular features found in the ExtraSensory dataset. A recursive neural network could be used to generate a probability output and could also be used in conjunction with a soft Max layer to act as a classifier but as stated it is not the best tool.

Random Forests Another option is a random forest classifier which has been used to great success by Phan (2015) [28]. However a random forest classifier is not well suited to unbalanced data sets and would perform worse as the number of classes in the problem scaled up. For feature engineering purposes, a random forest classifier was used to evaluate and compare the individual feature's importance in the ExtraSensory dataset.

Multilayer Perceptron Feed Forward Neural Network A multilayer perceptron (MLP) classifier is a feedforward neural network, trained via backpropagation. It was selected for this thesis. For this project, tabular time-invariant snapshots of data were used. The simple MLP classifier trained faster than other neural networks, and was theorised to produce better results. Due to the MLP's simplicity, it was also expected to compress better for any potential application of onboard processing.

The details of the models use (hyperparameters, results) are discussed in the Section 4.6 and Section 5.

4.6 Model Validation

Validation is the process of optimising the hyperparameters of a model, and evaluating each iteration on the hold-out validation set of data. This is an important aspect of the machine learning process because model validation cannot be learnt from data. That is to say, hyperparameter optimisation is out of the scope of training via traditional gradient decent and back propagation. Due to the " No Free Lunch " theorem, a model will be chosen and optimised before training and evaluation begins. This is a significant aspect of identifying the optimal model for the solution.

Model Performance Metrics Before the hyperparameters could be optimised, a metric had to be selected to optimise the network about. There are many metrics that will be discussed below.

Terminology Before evaluating metrics, the terminology of the following formula must be formalised. The following terms help describe what a " correct " classification is. These terms are:

- True Positives (TP) - A correctly classified positive sample

- True Negatives (TN) - A correctly classified negative sample
- False Positives (FP) - A negative example incorrectly classified as positive
- False Negatives (FN) - A positive example incorrectly classified as negative

These four terms are the main parameters in the following formulae.

Accuracy Accuracy is the measure of how many samples the classifier correctly classified across its operation, as shown in Equation 4.3.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{4.3}$$

Accuracy is used when a dataset is balanced (or close to balanced), and when there's no specific weighting on False Positives or False Negatives. Situations where this may have specific weightings on False Positive and False Negative rates could be the medical industry. In this scenario, a false positive will have a cost associated with it (the cost of further tests, or screenings) where as a false negative would affect a patient's health outcomes (as they could be severely impacted by the incorrect diagnosis).

In this project, accuracy was a very poor metric for the machine learning measure because the ExtraSensory dataset has 51 labels. When performing a binary classification on any individual label the majority class will be blank/negative. As such, a classifier which would classify every sample as blank/negative would still achieve a high accuracy rate (due to achieving a high True Negative rate). This was discussed as a key element of an unbalanced dataset, and accuracy can reinforce a classification bias.

Confusion Matrix A confusion matrix is a 4 cell table that lists TP, FP, FN, TN. This is a great visual tool to aid human evaluation of the model, as shown in Figure 15. The matrix can be normalised across the rows, as to show what portion of samples were correctly classified. The capacity to directly penalising (and optimise about) the False Positives, and False Negatives, could be an essential tool for a project.

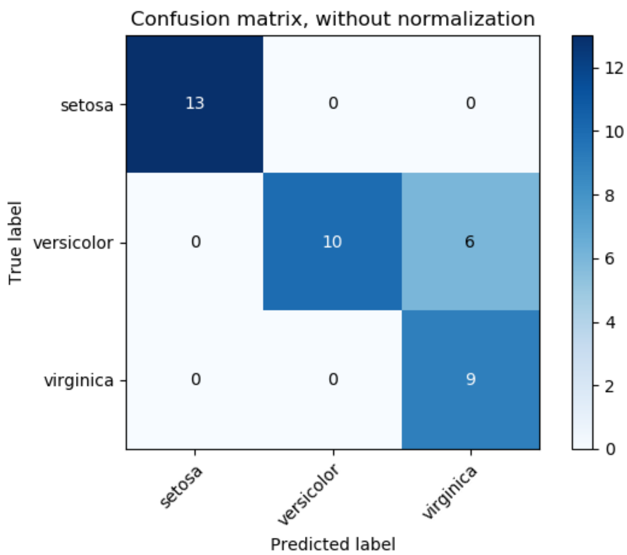

Figure 15: A confusion matrix of the Iris Database, sourced from scikitlearn's website [29].

In this project, the Confusion Matrix was generated to check and contextualise other performance measures. It is an important tool for weighing up the validity of classifier before accepting it for real world use.

Recall (or sensitivity) Building upon the direct use of the 4 factors, TP, FP, FN, TN, one can measure the ability of a classifier to " Recall " the true positive rate of a dataset as defined in Equation 4.4. This is the ratio of predicted positives vs the " True " positive rate.

$$Recall = \frac{TP}{TP + FN} \tag{4.4}$$

In this project, the Recall rate was of importance as the user case is to document an acoustic event, upon which any False Positives could be disregarded by the user, or provide interesting new insight into the behaviour observed.

Precision In contrast to Recall, precision is the measure of the error in Positives classification. That is, out of the samples classified as Positive, how many were suppose to be positive, or the ratio of correct predictions over all predictions.

$$Precision = \frac{TP}{TP + FP} \tag{4.5}$$

In this project, this was less important as discussed above. Precision may be important when a high degree of statistical confidence is needed for a decision, but missed opportunities may not be important.

Specificity Specificity is the compliment of to Recall. It is the ratio of correctly negative classifications vs the total number of negative classifications. This is another important metric for training models that are required to be highly " specific " it the model's predictions.

$$Precision = \frac{TN}{TN + FP} \tag{4.6}$$

In this project, the specificity was ideally to be be traded off against the Recall rate. For the reasons already discussed, the risk from false positives are negligible.

F1 Score An F1 score is an industry standard approach to evaluating a model. It is formally defined as the harmonic mean of the Precision and Recall, as shown in Equation 4.7. This score is a method of evaluating a model's Recall and Precision with equal weighing, but punishing an imbalance between the two scores.

$$F1Score = \frac{2 \times (Precision \times Recall)}{Precision + Recall} \tag{4.7}$$

This project used the F1 score as its prime metric for evaluating the model, and the effect of supplementary data on the acoustic event detection/classification task in question. A further extension of the F1 score was required to apply to a multi class or multilabel problem. This is done through a sum of F1 scores across each class or label. For best precision, a weight could be applied during the summation of F1 scores to best reflect (and optimise) the intent of the classifier.

Area Under the Receiver Operating Characteristics (AUROC) Curve Lastly, the Area Under the Receiver Operating Characteristics (AUROC) is an important measure of the a classifiers overall performance. The AUROC demonstrates how confident a classifier is across all classification decision boundary[9] thresholds.

In a more formal sense, the Receiver Operating Characteristics (ROC) is the relation between the True Positive rate against the False Positive rate, and how it changes when the classification decision threshold is changed. To calculate the ROC, a sample of TP and FP is taken for a varying threshold (the threshold needed for the classifier to classify a sample as positive or negative).

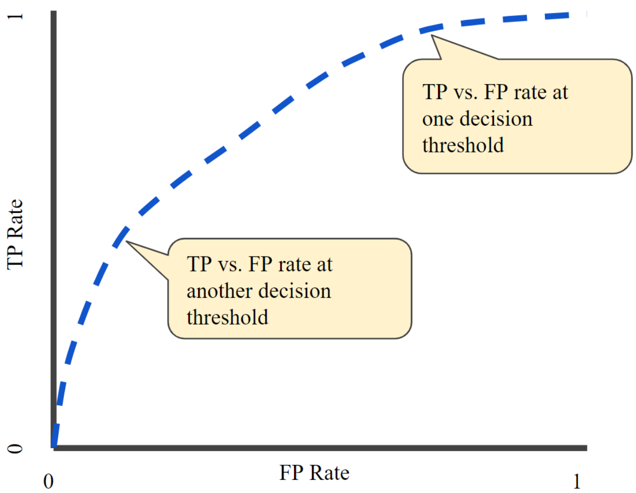

Figure 16: A ROC Curve example from Google's machine learning shortcourse [30]

In this project, the AUROC was not extensively used, except for educational purposes. The purpose of a AUROC is to demonstrate, and optimise the confidence in classification based on thresholds applied. The AUROC can be used to demonstrate a quantitative assessment of any particular sample (similar to a Regression problem). An important limitation is the ROC is plotted against each class, or label. Thus 51 AUROCs would need to be calculated to evaluate the performance of the multilabel classifier.

4.6.1 Hyperparameter Optimisation

Machine learning is about learning, and adjusting, the internal parameters of a classifier. A hyperparameter is a parameter that cannot be learnt from this process (such as backpropagation), as hyperparameters are external to the classifier. These are variables such as hidden layer size[10], training rates, or classifier specific strategies etc. The way to choose these hyperparameters is therefore an optimisation

[9] A decision boundary is the limit which separates a positive class from a negative class.
[10] A hidden layer is any layer of a MLP that sits between the input layer and the output layer

problem. The problem is to maximise the score selected from the metrics above, by trialling different hyperparameters. The following describe methods of finding the optimal hyperparameters.

Manual Search Hyperparameters could be manually selected and adjusted, in a search for a more optimal solution. This may be a valid option for interrogating what each hyperparameter does, and the impact on the classification score. This however would not be a valid solution for finding the optimal hyperparameters.

This was tried and yielded mediocre results compared to default hyperparameter settings on various algorithms.

Brute Force Search To improve upon the Manual Search, one could attempt to try every option available in the parameter space. For example,

```
for hyperparameter1 = 0 to 100 {
    for hyperparameter1 = 0 to 100 {
        TrainNetwork(hyperparameter1,hyperparameter2)
    }
}
```

This improves on the Manual Search, but is very inefficient ($O(n^2)$), and thus extremely slow. This was not tried during this project, but highlights the importance of the following algorithms.

Grid Search (Manual Grid Search) If the Brute Force Search method attempts to try every combination, then an improvement on this is to better specify the set from which a combination is formed. This is called a Grid Search. A Grid Search iterates through every combination provided. A 2D example shown in Figure 17 shows a grid is generated based on the hyperparameters intervals selected. A powerful aspect of Grid Search is the capacity to select non-equidistant, arbitrary intervals.

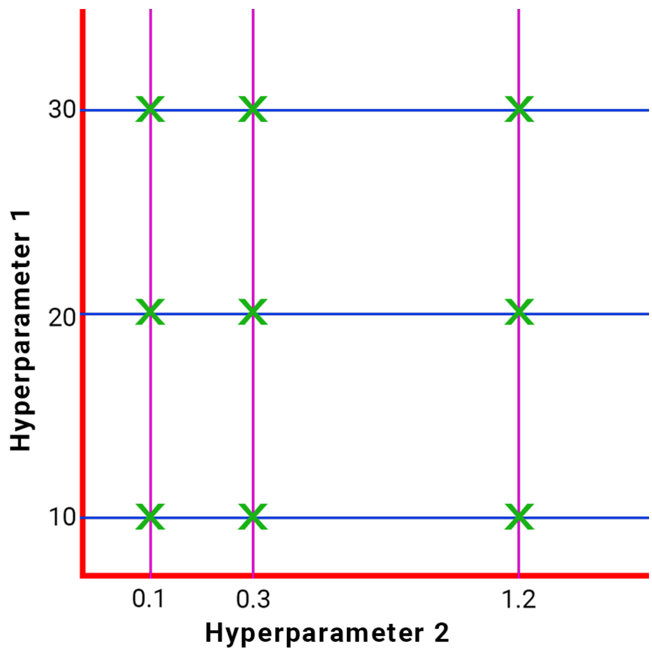

Figure 17: 3 options for Hyperparameter 1 and 2 have been selected. Each combination will be evaluated, as denoted by the green 'x'.

Grid Search was used for early implementations across both TensorFlow + Keras, and Scikit-learn.

Random Search (Random Grid Search) A Random Search builds upon the Grid Search, but aims to reduce the computation time required to find an optimal set of hyperparameters. This is especially true in higher dimensions[11]. The key difference between the two searches, is Random Search will not attempt to try every combination. It will sample random number of combinations, which has been " shown that empirically and theoretically, randomly chosen trials are more efficient for hyper-parameter optimization than trials on a grid " [31]. In part, this is due to a principle known as " Embarrassingly parallel ".
" Embarrassingly Parallel " Embarrassingly Parallel is a critical concept in machine learning that allows the Random and Grid searches to be run in parallel [32]. The concept is, because running a trial of 1 point does not have a dependence on any other point, they can be run in parallel. This is where mass parallel computing (such as the operations of a Graphic Processing Units, or GPU) allows very large hyperparameter optimisation searches to be run in parallel.

Due to the simplicity of the multilayer perceptron model selected, the random search was primarily used for hyperparameter optimisation.

Bayesian Hyperparameter Optimisation Bayesian Hyperparameter Optimisation is the principle of allowing a search to be influenced by previous experience. This is applied when the cost of the sample, such as a model's F1-score from a particular combination of hyperparameters, is computationally expensive to calculate. A more formal description can be found in Brochu's 2010 paper, "A Tutorial on Bayesian Optimization ...":

> " Bayesian optimization employs the Bayesian technique of setting a prior over the objective function and combining it with evidence to get a posterior function [33]. "

Bayesian Optimisation for hyperparameter selection was not tested, as it is suggested it is best suited for optimising more complex networks [34] [33]. It is however an excellent tie in to the MCH3900 Bayesian lectures, and could constitute a future package of work.

Final Hyperparameter output Below are snippets of the an implemented Random Search for an MLP classifier. Of note is the parameters included are listed in the Table 9 below. The hyperparameter options listed are a part of an iterative process where the range was manually narrowed after successive RandomSearches. This is a method to increase precision without high initial ranges that would be inefficient.

These parameters are then placed into a python Dict, and passed into the scikit RandomizedSearchCV function, as shown in Listing 5.

After Listing 5 has run, these were the the results:

Grid Search Implementation A Grid Search was also trialed, and details can be found in Appendix F.

[11] This is know as the " Curse of Dimensionality "

Table 9: Random Search Hyperparameters

Hyperparameters	Description	Options				
'learning_rate'	How ΔWeight changes wrt loss gradient	"constant"		"invscaling"		"adaptive"
'hidden_layer_sizes'	$(H_1, ..., H_{n-1}, H_n)$ where H is is number of nodes, and n is the number of layers	(145,)	(147,)	(149,)	(151,)	(153),
'alpha'	L2 Regularisation rate, for penalising feature weights. Helps prevent overfitting	0.09	0.11	0.13	0.15	0.17
'activation'	The model function $f(x)$ to fit to the problem function, where $y = f(\sum(W \times x)) + bias$	"logistic"		"relu"		"Tanh"

Listing 5: Model validation in python for the project

```
parameters={'learning_rate': ["constant", "invscaling", "adaptive"],
            'hidden_layer_sizes': [(145,), (147,), (149,), (151,), (153)],
            'alpha': [0.09, 0.11, 0.13, 0.15, 0.17, 0.19],
            'activation': ["logistic", "relu", "Tanh"]}
mlpc = MLPClassifier(verbose=False, early_stopping=True, learning_rate='adaptive',
    max_iter=1000)
clf = RandomizedSearchCV(estimator=mlpc, scoring='f1',param_distributions=parameters,
                         n_jobs=-1,verbose=1,cv=3);
clf.fit(X_train,y);
print("F-Score:␣%.2f" % clf.score(X_train, y))
print(clf.best_params_)
```

Machine Learning Libraries Used The classification libraries used were Google's TensorFlow and scikit-learn. Both libraries are python based (supporting both Python 3.0, and Python 2.7) [35][36]

TensorFlow TensorFlow is an opensource machine learning framework, primarily developed by Google. " It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications " [?]. TensorFlow in conjunction with with Keras was used to prototype the initial pipeline. According to the textbook, " Hands on machine learning with Scikit, Keras & TensorFlow ", a pipeline is:

> " A sequence of data processing components is called a data pipeline. Pipelines are very common in machine learning systems, since there is a lot of data to manipulate and many data transformations to apply [37]. "

TensorFlow and Keras were selected to initially design the classifier due its ability to run on the TensorFlow Lite. This was inline with the motivation of the project to allow widespread use of an AEC/D solution. The code snippet Listing 6 shows how a model converts.

The original prototyping for audio only classification with TensorFlow and Keras was to use matplotlib to plot the spectrogram of each audio sample, and use a sequential CNN to process the spectrograms.

```
F-Score: 0.92
{'solver': 'adam', 'learning_rate': 'invscaling', 'hidden_layer_sizes': (150,),
 'alpha': 0.17, 'activation': 'relu'}
```

Listing 6: Code snippet for converting the classifier model to TensorFlow Lite for mobile operation [38]

```
import tensorflow as tf

converter = tf.lite.TFLiteConverter.from_saved_model(saved_model_dir)
tflite_model = converter.convert()
open("converted_model.tflite", "wb").write(tflite_model)
```

A code snippet for that design is shown in Listing 7. This was influenced by Keras documenation, tutorials, and best practice for guidance.

Due to the size of the ExtraSensory dataset (60 users), and the use of the supplementary GPS, acceleration and gyroscope data, the classifier could have been prone to overfitting. Dropout layers are method of regularisation, which is used to help prevent overfitting in machine learning. Dropout layers work by "randomly dropping units (along with their connections) from the neural network during training" [?]. The purpose of dropout layers in this classifier is to prevent the classifier from overfitting by reducing the over reliance on any individual neuron, but also to reduce the over reliance on any individual data source. This is important because the Audio feature has the highest fidelity, and the classifier could have a heavy bias towards it and as a result overfit to th audio. Dropout layers will help minimise this overfit. To test this hypothesis, a Random Forest classifier (estimators = 300) was trained on all users to weigh the features, to help identify the design choices. Appendix D shows each feature importance breakdown, where Figure 18 shows the scikit Random Trees feature importance of the set of inputs (audio, accelerometer, gyroscope, GPS).

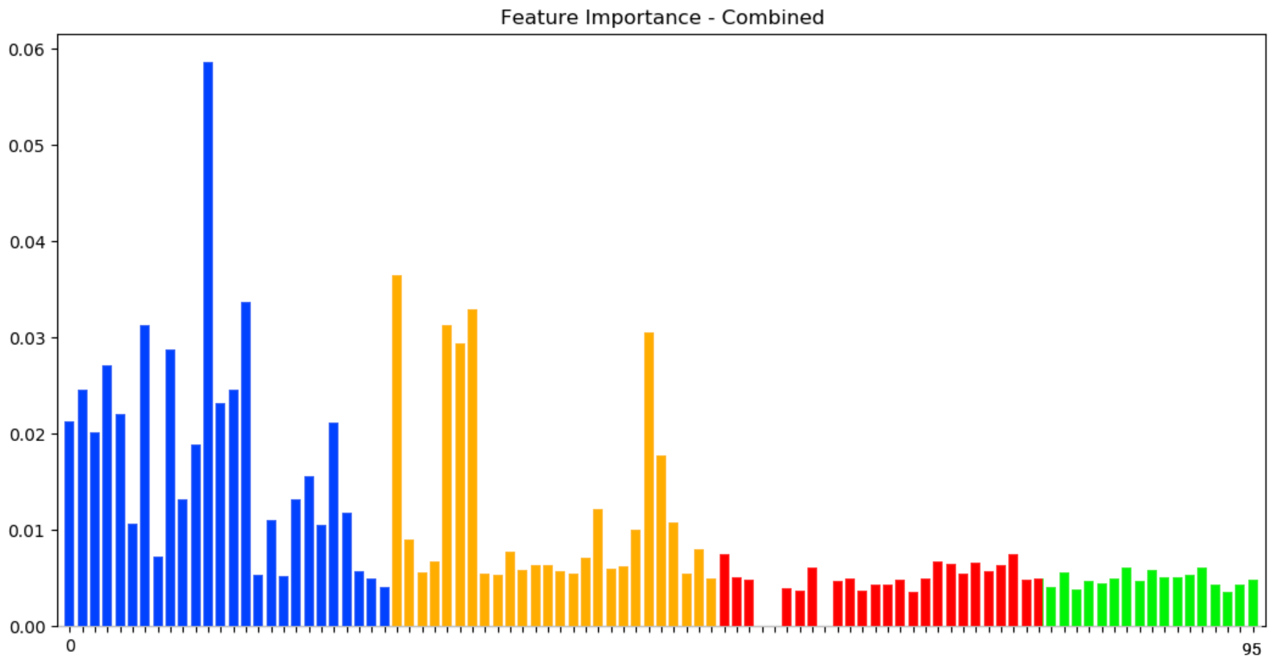

Figure 18: A Random Tree Classifier was used to evaluate feature importance on a the ExtraSensory after random undersampled dataset, during the design phase of the main classifier. Features were Audio (Blue, 26 bins), Accelerometer (Orange, 26 features), Gyroscope (Red, 26 features), GPS (Green, 17 features)

Listing 7: Code snippet for the early CNN model

```
model = Sequential()                                  #Start a sequential NN
model.add(Conv2D(32, (2), input_shape=X_Train.shape[1:])) #Add the first Convolution layer
model.add(Activation('relu'))                         #Use a relu activation function (consider logistic)
model.add(MaxPooling2D(pool_size=(2)))
model.add(Dropout(0.3))                              # Dropout critical, should help stop audio dominating over other features. Consider experimenting with dropout (30% recommended from stackoverflow)
model.add(Conv2D(64, (2)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2)))
model.add(Dropout(0.3))
model.add(Flatten())
model.add(Dropout(0.5))
model.add(Dense(51))
model.add(Activation('softmax'))
```

scikit-learn Scikit-learn is an extensive machine learning Python library. The purpose of the library is to remove much of the “ boilerplate ” code associated with implementing machine learning networks and algorithms. It also provides an extensive list of machine learning support functions. When combined with other Python libraries such as numpy, matlibplot and others, it allows for succinct and consistent reproduction of the results this project aims to deduce. It is for this reason that libraries for machine learning were utilised in contrast with writing a framework from scratch.

4.7 Method Summary

This section details the method of selecting the desired output, the required data input, a metric to score by, a classifier, optimising hyperparameters against the previous options, and finally training and validating the data.

For this project

1. The desired output was a label (or labels) detailing the acoustic event in the sample (e.g. The labels found in Appendix G like 'Walking').
2. The required data was the ExtraSensory dataset, or a custom dataset as recorded by the project's App.
3. The score selected was the F1 score. It is an industry standard.
4. The classifier chosen was the MLPClassifier which is a feedforward neural network, trained via backpropagation.
5. Hyperparameters of the MLPClassifier were chosen via a Random Search Cross Validation.
6. Training and Validation were run with early stopping[12] using a 10% validation set, using 59 out of the 60 ExtraSensory users.
7. Final testing is done with the 60$^{\text{th}}$ user.

[12] Early Stopping is the practice of training a neural network, and measuring the score of the classifier on the validation set. The error of the classifier on the testing set will tend to 0, but at a certain point the error on the validation set will reacha a global minimum. This represents the early stopping point to prevent overfitting.

5 Results

The purpose of this report was to examine the effect of supplementary data on acoustic event classification through machine learning. It sought to identify a demonstratable difference (or lack thereof) in the F1 score after comparing a classifier with and without supplementary data. In line with this, the results will analyse the F1 scores of a binary MLPClassifier, and a multi-label MLPClassifier, both trained on the ExtraSensory dataset using the method described in Section 4.

5.1 Binary MLPClassifier

A Binary MLP Classifier was run using the following settings:

```
{'solver': 'adam', 'learning_rate': 'adaptive', 'hidden_layer_sizes': (150, 100, 50), '
    alpha': 0.03, 'activation': 'tanh'}
```

It was trained to classify label 2[13]. Average training time for audio only was 42.9 minutes, whilst audio+extra was 53.6 minutes. This does not reflect the full training time, but rather the final optimising and training time after manual refinement of the Random Search Cross Validation hyperparameter range.

Results from Audio Only The results of the audio binary MLPClassifier are as follows:

```
--- Training Metrics ---
Samples: 8766
Sensors: ['Aud']
--- Performance Metrics ---
Accuracy: 0.51
Recall(sensitivity): 0.61
Precision: 0.56
Specificity: 0.39
F-Score: 0.76
--- Confusion Matrix ---
[[2366 1138]
 [ 706 3688]]
```

Results from Audio, Accelerometer, Gyroscope, GPS The results of the audio+extra binary MLPClassifier are as follows:

```
--- Training Metrics ---
Samples: 8766
Sensors: ['Aud', 'Acc', 'Gyro', 'Loc']
--- Performance Metrics ---
Accuracy: 0.51
Recall(sensitivity): 0.55
Precision: 0.55
Specificity: 0.45
F-Score: 0.84
--- Confusion Matrix ---
```

[13] Sitting

```
[[2892  657]
 [ 609 3707]]
```

Area Under Receiver Operating Characteristics (AUROC) The AUROC was calculated and plotted for the binary MLP classifier.

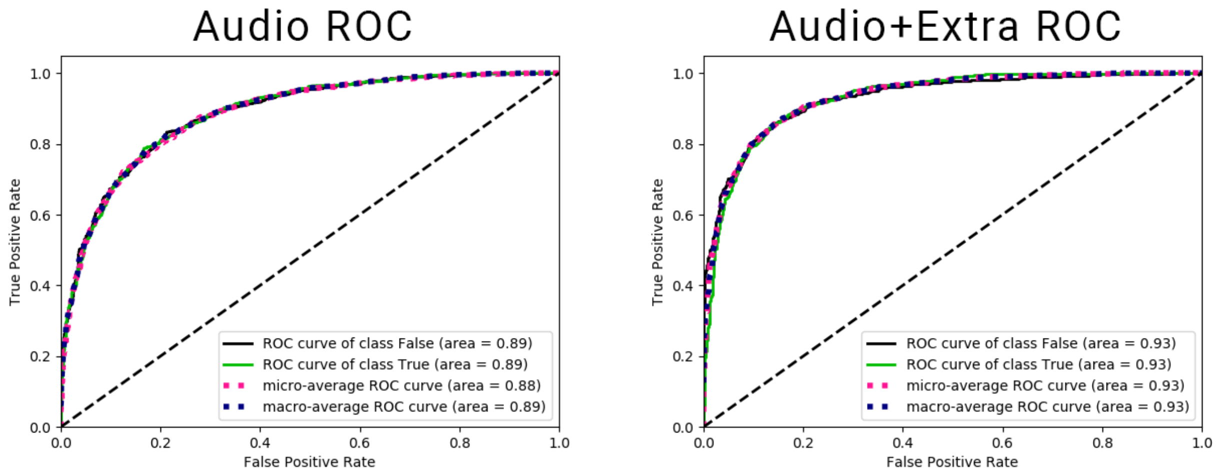

Figure 19: ROCs for Audio, Audio+Extra

The AUROC was not calculated for the multilabel, as the AUROC is a comparison on a per-class classification basis, and would be represented by the 20 labels evaluated.

5.2 Multilabel MLPClassifier

A Multilabel MLP Classifier was run using the following settings:

```
{'solver': 'adam', 'learning_rate': 'invscaling', 'hidden_layer_sizes': (150, 200), '
    alpha': 0.07, 'activation': 'relu'}
```

It was trained to classify labels 2 through 22 inclusive. Average training time for Audio only was 60.6 minutes, whilst Audio+Extra was 73.1 minutes. This does not reflect the full optimising and training time, but rather the final training time after manual refinement of the Random Search Cross Validation hyperparameter range.

Results from Audio Only The results of the audio+extra multilabel MLPClassifier are as follows:

```
--- Training Metrics ---
Samples: 7903
Sensors: ['Aud']
--- Performance Metrics ---
Accuracy: 0.95
Recall(sensitivity): 0.63
Precision: 0.76
Specificity: 0.98
F-Score: 0.67
```

Results from Audio, Accelerometer, Gyroscope, GPS The results of the audio+extra multilabel MLPClassifier are as follows:

```
--- Training Metrics ---
Samples: 2158
Sensors: ['Aud', 'Acc', 'Gyro', 'Loc']
--- Performance Metrics ---
Accuracy: 0.95
Recall(sensitivity): 0.73
Precision: 0.79
Specificity: 0.98
F-Score: 0.75
```

5.3 Summary of Results

The results achieved (shown in Table 10) demonstrate that the supplementary data used in accordance with the method described in Section 4, contribute to an 8% increase to the F1 score in both the binary MLP Classifier, and the multilabel classifier when compared with audio only input. This a relative increase of 10.5% and 11.9% respectively.

Table 10: F1 Score Results

	Audio	Audio+Extra	F1 Increase	% Improvement
Binary	0.76	0.84	8%	10.5%
Multilabel	0.67	0.75	8%	11.9%

6 Discussion

The results above aim to demonstrate a quantitative difference in F1 score across the two classifiers with and without Accelerometer, Gyroscope and GPS data. The results achieved did produce a quantitative relative improvement of 10.5% and 11.9% for the Binary and Multilabel result respectively. The relative improvements of 10.5% and 11.9% are significant in a field that frequently publishes about improvements in the single digits, as seen in Figure 2.

Assumptions The results are constrained by the assumptions made in Section 4. One assumption is that the " No Free Lunch " theorem infers that no individual classifier is theoretically better than another pre-training. This assumption implies that the results of this project will affect all models, but the quantitative impact may vary post-training. Similarly, the dataset and training may significantly affect the strength of this result. Consequently, the strength of the findings in this thesis is unknown in the context of application to other projects.

Causation The increase in F1 score could occur in two areas. The first is during hyperparameter optimisation, in which the hyperparameters are fitted to the activation model using audio and/or the supplementary data, optimised about the F1 score. The second is in the fitting itself, where the supplementary data may contain a signal that could be benefiting the F1 score of the model. If this hypothesis is correct, then these results will hold valid across other classifiers that require hyperparameter optimisation and fitting of training data.

Confidence and Qualitative Evaluation of Result The result had been analysed using the Area Under Receiver Operating Characteristics (AUROC) plot, to determine how the classifier would operate over different decision boundaries. The AUROC plots shown in Figure 19 show the consistency of the model. This is demonstrated via the smooth continuity of the ROC curve (no major jumps in TP performance). This ensures that the results produced were not the result of a classifier optimised about an artefact of the post-training model.

Changing Hyperparameters The results assume that using static hyperparameters across the audio, and audio+extra classifiers would be an invalid strategy of testing the hypothesis. It would be expected that optimising the hyperparameters for one classifier and not the other would result in the other classifier performing worse. This would result in a convolution of the effect from the extra data, and a classifier with poorly optimised hyperparameters. As such, there is an inverse assumption that optimising both classifiers separately, but using the same methodology, is a fair strategy to minimise the model and its hyperparameter's effect on F1 score.

Complexity The complexity of the classification task was low compared to similar studies [39][40] due to the limited and broad labels available in the ExtraSensory dataset. However, the number of classes are comparative to previous studies (61 classes, Virtanen (2010) [40]). To improve these results, the complexity of the labels could be improved. A promising new technique is to train a network on weakly supervised data [39], which could extend to generating/classifying new labels on the ExtraSensory dataset. This would allow a more complex dataset, such as AudioSet [41] to reprocess the ExtraSensory database to have a larger number of classes.

Acoustic Event Detection The validity of the result holds true for acoustic event classification, however, no acoustic event " detection " was carried out. That is, a sound sample was classified as containing an acoustic event rather than identifying when the acoustic event occurred in the sound sample. Phan (2016) [42] discusses the differences and complexities involved in AEC vs AED, and evaluates the use of classifiers as a verification stage in an acoustic event detection pipeline. Phan's results suggested that the AEC is a crucial element of the detection pipeline. Given this, the results in this thesis may then be valid for acoustic event detection, given AEC's role in AED.

Quality of Data ExtraSensory was used as the dataset for this project. As a result of using a public dataset, there was no investigation into the quality of the data within the dataset beyond reviewing the original study. The quality of the recording device, any preproccessing, or whether they generalised well, have not been checked. This is a key limitation with analysing the quantitative result. However, this limitation does not invalidate the result because the dataset was collected with mobile phones. Mobile phones would likely be the main datasource for any future dataset involving audio, accelerometer, gyroscope and GPS data.

Furthermore, no effort was made to audit the labels provided in the dataset, as it was assumed that the dataset was 100% accurate. If this was not the case, this assumption would affect the expected relative improvement in the result. The labels were self-reported and cleaned up by the researchers [22]. Whilst this introduces human error into the dataset, all datasets will require a decision boundary that may be controversial. To ensure the quality of the decision boundary, future studies should take guidance from their use case in real world use, or would take guidance from similar studies.

Choice in Frameworks The choice in machine learning frameworks (and their implementation) could affect the result. For this project, an assumption was made that the choice in frameworks would be of minimal impact, and the results would remain valid across other frameworks.

Details of the Audio Data A key assumption was that the audio data provided (mono-channel recorded from a mobile phone) was representative of all audio data for use in machine learning. Furthermore, the audio was processed into Mel-frequency cepstral coefficients (MFCCs), a leading approach to processing audio data. This report did not attempt to test the results against different types of audio data (such as multi channel, higher dynamic ranges, etc). The report did not attempt to review the results with other approaches of processing audio data (such as spectrograms with computer vision / machine learning), which may significantly affect the result.

Types of Supplementary Data This report had defined " supplementary data " as accelerometer, gyroscope, and GPS data. An assumption was that the AEC/D field would accept this definition within the context of the results. A limitation of this result was whether the result would transfer and compare with different supplementary data sources. For example, would similar benefits be gained from inclusion of video, metadata, multiple audio sources etc.

Model Optimisation A key assumption was that the model selected had been close to fully optimised to the extent Section 4 allowed. This assumption infers that any improvement in F1 score with the addition of supplementary data can be determined to be from the introduction of the data itself, and not a result of better hyperparameter optimisation.

F1 Score as a Metric This project sought to measure a discernible difference in F1 score by the introduction of supplementary data. An assumption that F1 score is the most appropriate indicator of a positive result is made. It could be noted that F1 score is the harmonic mean of the True Positive and True Negative rate. Had the project selected a different metric from Section 4.6, the intent of the question would remain but the relative increase in performance would be vastly different. An assumption was that an improvement in F1 score throughout hyperparameter optimisation and backpropagation/training of the model is desirable in other projects.

Limiting Model Choice to Avoid Excessive Feature Some machine learning models are sensitive to the number of features. Dimensionality reduction is critical to keep only the features that are of high importance. As the result didn't dicern which features were critical, the result may need to be duplicated after a dimension reduction.

Temporal Features This classifier did not make use of any temporal aspects that would be available to other datasets. This was a limitation of the dataset. As the supplementary data is a measurement in the time domain, it would be expected that their inclusion to a AEC/D pipeline would benefit, however it would be left for future work to determine the exact nature of this benefit.

Significance This hypothesis presents the impact of the supplementary datasources of an accelerometer, gyroscope and GPS on AEC/D. This impact was of 10.5% and 11.9% for their respective models. The qualitative aspects of these results will be applicable to model implementation in other projects. Classifiers are a fundamental building block of the AEC/D field. This thesis will be relevant to future research that makes use of classification as it will aid researchers in the selection of data inputs.

The decision to define " supplementary data " as accelerometer, gyroscope and GPS was a calculated decision as these datasources are not only prevalent in modern society, but commonly found together in devices such as mobile phones. Subsequently, the research this thesis contributes to will have a widespread, real-world impact across varying industries and products.

This thesis is significant because it is a key contributer to the real world impact of the AEC/D field.

7 Conclusion

The report demonstrated that introducing accelerometer, gyroscope, and GPS data contributed to an 8% increase in the F1 score in both a binary, and a multilabel MLPClassifier. This a relative increase of 10.5% and 11.9% respectively, compared to a baseline without the extra data. The results had significant limitations and assumptions about typical AEC/D problems. These limitations do not affect the significance and qualitative result of this report, but will affect the quantifiable improvement expected when implementing these findings in other projects. The result was discussed and found to be valid. This thesis contributes to the field by specifying an estimate improvement for the inclusion of the supplementary data. At the time of publication, the question of the effect supplementary data on audio classification had not been previously answered.

This report focused on investigating and analysing an AEC/D problem. A heavy focus was placed on creating a reviewable machine learning implementation, to best facilitate transferring results to other projects.
The methodical process applied in this project showed that the result can be isolated. Whilst the performance cannot be attributed to hyperparameter optimisation or backpropagation analytically, it was evident that the effect was present in these areas.
The report looks at how the AEC/D pipeline would be created (the creation of the app, preprocessing, etc.), much of which is not reflected in the final results but were critical in the overall Final Year Project. Highlighting these aspects are also important in converying a contextual understanding of the design choices being made in the result. This context aims to help reproducablity.
The results achieved showed audio as a sole input for determining physical actions to be viable, but the extra available data did improve the classification. The work presented does demonstrates a contribution to the audio processing space and explores how mechatronics domain knowledge impacts and extends machine learning.

Based on the work presented, it is clear that audio data processing will be a major tool and research topic into the future. Without large datasets of audio paired with physical sensor data, the machine learning tools explored in this Final Year Project may not be able to be used. As such, it is recommended that due to the rise of well lablled audio datasets, audio only should remain the key basis for further research into this space. An ensemble of classifiers with a well trained AEC/D based on audio only classifier, could make use of the extra data to form a more effective classifier.

A recommendation would be to apply the outcome to a relevant project when the supplementary data is available, but consider the extra training time required. Also consider the quality of datasets available, and whether requiring the inclusion of supplementary data would exclude higher quality datasets. For example, an AEC/D classifier trained on the ExtraSensory dataset (with accelerometer, gyroscope, and GPS data) may perform worse than an audio only classifier of the same design trained on the Google AudioSet dataset.

In conclusion the question of “ what is the effect of supplementary data on acoustic event classification through machine learning ” has been answered. There is an 8% increase in F1 score, or 10.5/11.9% relative increase in a binary MLPClassifier and multilabel MLPClassifier respectively. The implications of this report on potential applications are well defined; any improvement on classification brings ideas like reality closed captioning for the hearing impaired, improved security closed loop TV, improved ride sharing trip documenation, early detection of potential accidents or medical emergencies, and a method of debriefing across multiple industries closer to reality.

8 Future Work

Individual effect A package of future work would be to discern what impact each additional data-source had on the result. It could be inferred from Appendix D that accelerometer was likely the most impactful element. An investigation to discern the specific impact each sensor had, and the cost of including " less important " features vs training time would be of critical value.

Review Preprocessing of Data A package of future work would be to analyse whether the method of preprocessing chosen affects the results. For example, this project used MFCCs, but another common practice is to use a sliding window spectrograms.

Temporal Feature Engineering A package of future work would be to discern how this result might affect classifiers that use temporal elements for AEC/D.

Replicate Across Different Datasets The result should hold valid across other datasets, but it would be important to peer review these findings with a different dataset. The results should also be replicated across varying detail levels of labels (more details, more focused, or more broad, more cross domain).

Use Another AEC/D to Label This Dataset A key limitation to this field of search is the lack of datasets with both audio and IMU data. As audio classifiers trained on AudioSet improve, a classifer could be trained and optimised about minimising the False Positive rate. This could then be realiably used to append new labels to the ExtraSensory dataset. This would enable higher quality research to be produced with the dataset.

Bayesian Optimization Whilst not essential to the results of this paper, it would be an interesting report to evaluated how a completely different hyperparameter optimisation technique would effect the results. This is the study of metaheuristics. It would be an excellent tie in for this project from the MCH3900 Bayesian lectures.

Novel Use of Bogosort as Optimisation Strategy A final suggestion for future work is to adapte the Bogosort to machine learning, as it is an increasingly popular area of research [43][44][45]. The suggested method for using the Bogosort (or Bogosearch) would be to iterate over the hyperparameters of the network, then iterate over the neuron weights. If successfully applied, it could be the first instance of the Bogosort being applied in the machine learning sphere.

9 References and Citations

References

[1] G. Association, "The mobile economy 2019," 2019. [Online]. Available: https://www.gsmaintelligence.com/research/?file=b9a6e6202ee1d5f787cfebb95d3639c5&download

[2] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[3] T. M. Mitchell, *Machine Learning.* McGraw-Hill Education, 1997. [Online]. Available: https://www.amazon.com/Machine-Learning-Tom-M-Mitchell/dp/0070428077?SubscriptionId=AKIAIOBINVZYXZQZ2U3A&tag=chimbori05-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=0070428077

[4] A. Mesaros, T. Heittola, and T. Virtanen, "A multi-device dataset for urban acoustic scene classification," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2018 Workshop (DCASE2018)*, November 2018, pp. 9–13. [Online]. Available: https://arxiv.org/abs/1807.09840

[5] L. L. Morrison, John E. ; Meliza, "Foundations of the after action review process," INSTITUTE FOR DEFENSE ANALYSES ALEXANDRIA VA, Tech. Rep., 1999.

[6] D. o. t. A. HEADQUARTERS, "Circular (tc) 25-20: A leader's guide to after action reviews," Department of the Army, Tech. Rep., 1993.

[7] C. R. Johnson Pivec, "Debriefing after simulation: Guidelines for faculty and students," 2011.

[8] J. Kaliner, "When will we ever learn? the after action review, lessons learned and the next steps in training and educating the homeland security enterprise for the 21st century," Ph.D. dissertation, NAVAL POSTGRADUATE SCHOOL MONTEREY, CALIFORNIA, 2013.

[9] T. U. of Queensland, "Literature review," The University of Queensland, Tech. Rep., 2018.

[10] T. Heittola, "Research: Sound event detection," 2018.

[11] DCASE., *(DCASE 2018 Schedule).* Schedule, 2018. [Online]. Available: http://dcase.community/

[12] *(n.* d.), 2018.

[13] *Results.* (n. d.). DCASE2017, 2017. [Online]. Available: http://www.cs.tut.fi/sgn/arg/dcase2017/challenge/task-sound-event-detection-in-real-life-audio#results

[14] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," *CoRR*, 2016.

[15] L. JiaKai, "Mean teacher convolution system for dcase 2018 task 4," DCASE2018 Challenge, Tech. Rep., September 2018.

[16] B. H. W. Limited, *Systems Engineering: Design and deliver better products and processes.* [Online], 2018.

[17] I. O. for Standardization, *IEEE 15288: Systems and software engineering – System life cycle processes*, ieee 15288 ed., International Organization for Standardization, 2015, international Organization for Standardization. (n.d.). Systems and software engineering – System life cycle processes. Retrieved from https://www.iso.org/standard/63711.html.

[18] B. H. W. Limited, *The Systems Thinking Tool Box*. [Online], 2015.

[19] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Muller, "Efficient backprop," in *Neural networks: Tricks of the trade*. Springer, 2012, pp. 9–48.

[20] S. Havard, "Google pixel xl teardown," Oct. 2016, https://www.electronicproducts.com/Google_Pixel-whatsinside_text-205.aspx.

[21] Google, "Sensors motion," 2019.

[22] Y. Vaizman, K. Ellis, and G. Lanckriet, "Recognizing detailed human context in the wild from smartphones and smartwatches," *IEEE Pervasive Computing*, vol. 16, no. 4, pp. 62–74, oct 2017.

[23] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning*. Springer-Verlag GmbH, 2017. [Online]. Available: https://www.ebook.de/de/product/20292548/gareth_james_daniela_witten_trevor_hastie_robert_tibshirani_an_introduction_to_statistical_learning.html

[24] D. A. Huffman, "The synthesis of sequential switching circuits," *J. Franklin Institute, vol. 257, no*, vol. 257, no. 3, pp. 161–190, Mar. 1954.

[25] M. Magdon-Ismail, "No free lunch for noise prediction," *Neural Computation*, vol. 12, no. 3, pp. 547–564, 2000. [Online]. Available: https://doi.org/10.1162/089976600300015709

[26] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[27] C. Fefferman, S. Mitter, and H. Narayanan, "Testing the manifold hypothesis," *Journal of the American Mathematical Society*, vol. 29, no. 4, pp. 983–1049, feb 2016.

[28] H. Phan, M. Maas, R. Mazur, and A. Mertins, "Random regression forests for acoustic event detection and classification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 1, pp. 20–31, jan 2015.

[29] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Confusion matrix — scikit-learn 0.21.2 documentation," 2019.

[30] Google Brain, "Classification: Roc curve and auc — machine learning crash course — google developers," 2019. [Online]. Available: https://www.tensorflow.org/install/pip

[31] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research*, vol. 13, no. Feb, pp. 281–305, 2012.

[32] N. S. Maurice Herlihy, *The Art of Multiprocessor Programming, Revised Reprint*. Elsevier Science, 2012. [Online]. Available: https://www.ebook.de/de/product/19187472/maurice_herlihy_nir_shavit_the_art_of_multiprocessor_programming_revised_reprint.html

[33] E. Brochu, V. M. Cora, and N. de Freitas, “A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning,” 2010.

[34] J. Snoek, H. Larochelle, and R. P. Adams, “Practical bayesian optimization of machine learning algorithms,” 2012.

[35] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Release history — scikit-learn 0.21.2 documentation,” 2019. [Online]. Available: https://scikit-learn.org/stable/whats_new.html

[36] Google Brain, “Install tensorflow with pip — tensorflow,” 2019. [Online]. Available: https://www.tensorflow.org/lite/guide/get_started#2_convert_the_model_format

[37] A. Geron, *Hands on Machine Learning with Scikit, Keras & Tensorflow*. O’Reilly, 2017.

[38] Google Brain, “Get started with tensorflow lite,” 2019. [Online]. Available: https://www.tensorflow.org/install/pip

[39] S.-Y. Tseng, J. Li, Y. Wang, J. Szurley, F. Metze, and S. Das, “Multiple instance deep learning for weakly supervised small-footprint audio event detection,” 2017.

[40] T. Virtanen, “Acoustic event detection in real life recordings,” 2010.

[41] Google, *AudioSet: An Ontology and Human-labeled dataset for audio events*. Google, 2018.

[42] H. Phan, P. Koch, F. Katzberg, M. Maass, R. Mazur, I. McLoughlin, and A. Mertins, “What makes audio event detection harder than classification?” 2016.

[43] H. Gruber, M. Holzer, and O. Ruepp, “Sorting the slow way: An analysis of perversely awful randomized sorting algorithms,” in *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2007, pp. 183–197.

[44] M. Sherman, “Bogo-sort is sort of slow,” jun 2013.

[45] M. Holzer and J.-T. Maurer, “Selection via the bogo-method - more on the analysis of perversely awful randomized algorithms,” 2018.

A Full Extracts of Journals

N/A, email correspondence available on request. aasdasda

B Reflection

I'll write this from the first person perspective. This project was a whirlwind. Taken over the 2018 S2, 2019 S1 period, including the Christmas Holidays, the project was a learning experience. The results I produced were somewhat in line with a common sense guess, but I can take pride in knowing the outcome is quantified, based on hard work, and flexed a lot of Engineers Australia's comptencies (for professional engineers). I learnt a lot starting with...

Lessons Learnt

- The Industry - FYP was a squandered opportunity on my part, but almost by my choice in FYP topic. Work was able to provide significant support for implementation, and even the prospect of data collection through engaging with potential customers and stakeholders for different use-cases. Unfortunately, I felt uncomfortable pursing that due to how heavy the time spent in working on implementation as a practitioner vs theoretical engineering as a 4th year engineering student. The lesson learnt is to have not done an Industry FYP without understanding how it'll affect the pathway to a final thesis. I could have elected to conduct research into a topic better suited to the thesis format.
- Implementation drove some technology choices I probably would have been better off without. E.g. I went to a Matlab seminar on how to develop machine learning in Matlab. Had I used this, the project could have been more homogenous and focused on the big question around "acoustic event detection/classification". But to do so means I don't develop the python ML skills needed for 'real world' ML tasks. A lesson learnt is to focus on what is assessable, and to probably have used Matlab to demonstrate I understood the theory, not to learn and demonstrate the actual skills required to practice in Industry.
- The FYP is estimated to be a 10 week full time commitment to meet the 360 hours demanded of a 30 unit subject. I potentially " wasted " 6 weeks just identifying the question I was trying to answer in the report, but ultimately learnt a lot from the process. Having said that, if I hadn't elected to do a Industry FYP, the work of understanding what problem I would tackle potentially could be better shared between the supervisor and myself.
- A tale as old as time: smaller scope, get it done in Part A, expand and write in Part B.

Engineers Australia Comptency 14: Problem Analysis This project was the first time that there was very little constraints in the solution I chose. Other courses I have done allowed system engineering to determine a solution, but they were always limited and restricted to the task at hand. This was the first task I felt like I could fully engage with the 14th EA comptency.

Engineers Australia Comptency 16: Evaluation The core of this task was to use the entirety of my engineering skills developed at UoN over the last 4 years, and ultimately evaluate the result. I felt this was a success.

Engineers Australia Comptency 12: Advanced Engineering The FYP has made me appreciate Subject Matter Expertise. In MCHA3900, I focused heavily on understanding the physical hardware involved in the Hexapod - Matlab, RPi, i2c, Openservo v3, and the physical bot itself. I felt I knew that system well, but I recognised my limited understanding. In comparison to that, I reflect on *how much* I've learnt during this FYP, and to what depth of detail compared to MCHA3900. I think I already knew that when I read it is ‘MCHA3900 - INTRO TO ROBOTICS’, but seeing how that was already a good 200 hours I’m surprised just how much more I’ve had to delve into this.

On that note, I lastly reflect on the skills I’ve learnt, and the perspective I’ve gained in how complex the field is, and how much more there is to learn.

C Full Result Output

C.1 Binary Classifier

```
PS C:\FYP\2019-05-11\extra sensor> .\culled_python.py
00EABED2-271D-49D8-B599-1D4A09240601.features_labels.csv.gz |X2287-Y2287 / timestamps
    2287 / feature_names 225 / label_names 51
098A72A5-E3E5-4F54-A152-BBDA0DF7B694.features_labels.csv.gz |X9100-Y9100 / timestamps
    9100 / feature_names 225 / label_names 51
0A986513-7828-4D53-AA1F-E02D6DF9561B.features_labels.csv.gz |X13060-Y13060 / timestamps
    13060 / feature_names 225 / label_names 51
0BFC35E2-4817-4865-BFA7-764742302A2D.features_labels.csv.gz |X16168-Y16168 / timestamps
    16168 / feature_names 225 / label_names 51
0E6184E1-90C0-48EE-B25A-F1ECB7B9714E.features_labels.csv.gz |X23689-Y23689 / timestamps
    23689 / feature_names 225 / label_names 51
1155FF54-63D3-4AB2-9863-8385D0BD0A13.features_labels.csv.gz |X26374-Y26374 / timestamps
    26374 / feature_names 225 / label_names 51
11B5EC4D-4133-4289-B475-4E737182A406.features_labels.csv.gz |X35219-Y35219 / timestamps
    35219 / feature_names 225 / label_names 51
136562B6-95B2-483D-88DC-065F28409FD2.features_labels.csv.gz |X41437-Y41437 / timestamps
    41437 / feature_names 225 / label_names 51
1538C99F-BA1E-4EFB-A949-6C7C47701B20.features_labels.csv.gz |X47986-Y47986 / timestamps
    47986 / feature_names 225 / label_names 51
1DBB0F6F-1F81-4A50-9DF4-CD62ACFA4842.features_labels.csv.gz |X55361-Y55361 / timestamps
    55361 / feature_names 225 / label_names 51
24E40C4C-A349-4F9F-93AB-01D00FB994AF.features_labels.csv.gz |X60132-Y60132 / timestamps
    60132 / feature_names 225 / label_names 51
27E04243-B138-4F40-A164-F40B60165CF3.features_labels.csv.gz |X65059-Y65059 / timestamps
    65059 / feature_names 225 / label_names 51
2C32C23E-E30C-498A-8DD2-0EFB9150A02E.features_labels.csv.gz |X73575-Y73575 / timestamps
    73575 / feature_names 225 / label_names 51
33A85C34-CFE4-4732-9E73-0A7AC861B27A.features_labels.csv.gz |X79747-Y79747 / timestamps
    79747 / feature_names 225 / label_names 51
3600D531-0C55-44A7-AE95-A7A38519464E.features_labels.csv.gz |X84950-Y84950 / timestamps
    84950 / feature_names 225 / label_names 51
40E170A7-607B-4578-AF04-F021C3B0384A.features_labels.csv.gz |X92599-Y92599 / timestamps
    92599 / feature_names 225 / label_names 51
481F4DD2-7689-43B9-A2AA-C8772227162B.features_labels.csv.gz |X99290-Y99290 / timestamps
    99290 / feature_names 225 / label_names 51
4E98F91F-4654-42EF-B908-A3389443F2E7.features_labels.csv.gz |X102540-Y102540 /
    timestamps 102540 / feature_names 225 / label_names 51
4FC32141-E888-4BFF-8804-12559A491D8C.features_labels.csv.gz |X107519-Y107519 /
    timestamps 107519 / feature_names 225 / label_names 51
5119D0F8-FCA8-4184-A4EB-19421A40DE0D.features_labels.csv.gz |X114136-Y114136 /
    timestamps 114136 / feature_names 225 / label_names 51
5152A2DF-FAF3-4BA8-9CA9-E66B32671A53.features_labels.csv.gz |X120753-Y120753 /
    timestamps 120753 / feature_names 225 / label_names 51
59818CD2-24D7-4D32-B133-24C2FE3801E5.features_labels.csv.gz |X126700-Y126700 /
    timestamps 126700 / feature_names 225 / label_names 51
59EEFAE0-DEB0-4FFF-9250-54D2A03D0CF2.features_labels.csv.gz |X134242-Y134242 /
    timestamps 134242 / feature_names 225 / label_names 51
5EF64122-B513-46AE-BCF1-E62AAC285D2C.features_labels.csv.gz |X138153-Y138153 /
    timestamps 138153 / feature_names 225 / label_names 51
61359772-D8D8-480D-B623-7C636EAD0C81.features_labels.csv.gz |X144232-Y144232 /
```

```
    timestamps 144232 / feature_names 225 / label_names 51
61976C24-1C50-4355-9C49-AAE44A7D09F6.features_labels.csv.gz |X152962-Y152962 /
    timestamps 152962 / feature_names 225 / label_names 51
665514DE-49DC-421F-8DCB-145D0B2609AD.features_labels.csv.gz |X162129-Y162129 /
    timestamps 162129 / feature_names 225 / label_names 51
74B86067-5D4B-43CF-82CF-341B76BEA0F4.features_labels.csv.gz |X169427-Y169427 /
    timestamps 169427 / feature_names 225 / label_names 51
78A91A4E-4A51-4065-BDA7-94755F0BB3BB.features_labels.csv.gz |X181423-Y181423 /
    timestamps 181423 / feature_names 225 / label_names 51
797D145F-3858-4A7F-A7C2-A4EB721E133C.features_labels.csv.gz |X185016-Y185016 /
    timestamps 185016 / feature_names 225 / label_names 51
7CE37510-56D0-4120-A1CF-0E23351428D2.features_labels.csv.gz |X194777-Y194777 /
    timestamps 194777 / feature_names 225 / label_names 51
7D9BB102-A612-4E2A-8E22-3159752F55D8.features_labels.csv.gz |X196377-Y196377 /
    timestamps 196377 / feature_names 225 / label_names 51
8023FE1A-D3B0-4E2C-A57A-9321B7FC755F.features_labels.csv.gz |X205566-Y205566 /
    timestamps 205566 / feature_names 225 / label_names 51
806289BC-AD52-4CC1-806C-0CDB14D65EB6.features_labels.csv.gz |X214808-Y214808 /
    timestamps 214808 / feature_names 225 / label_names 51
81536B0A-8DBF-4D8A-AC24-9543E2E4C8E0.features_labels.csv.gz |X221215-Y221215 /
    timestamps 221215 / feature_names 225 / label_names 51
83CF687B-7CEC-434B-9FE8-00C3D5799BE6.features_labels.csv.gz |X230754-Y230754 /
    timestamps 230754 / feature_names 225 / label_names 51
86A4F379-B305-473D-9D83-FC7D800180EF.features_labels.csv.gz |X241492-Y241492 /
    timestamps 241492 / feature_names 225 / label_names 51
96A358A0-FFF2-4239-B93E-C7425B901B47.features_labels.csv.gz |X247311-Y247311 /
    timestamps 247311 / feature_names 225 / label_names 51
9759096F-1119-4E19-A0AD-6F16989C7E1C.features_labels.csv.gz |X257270-Y257270 /
    timestamps 257270 / feature_names 225 / label_names 51
99B204C0-DD5C-4BB7-83E8-A37281B8D769.features_labels.csv.gz |X263308-Y263308 /
    timestamps 263308 / feature_names 225 / label_names 51
9DC38D04-E82E-4F29-AB52-B476535226F2.features_labels.csv.gz |X272994-Y272994 /
    timestamps 272994 / feature_names 225 / label_names 51
A5A30F76-581E-4757-97A2-957553A2C6AA.features_labels.csv.gz |X274661-Y274661 /
    timestamps 274661 / feature_names 225 / label_names 51
A5CDF89D-02A2-4EC1-89F8-F534FDABDD96.features_labels.csv.gz |X280701-Y280701 /
    timestamps 280701 / feature_names 225 / label_names 51
A7599A50-24AE-46A6-8EA6-2576F1011D81.features_labels.csv.gz |X284599-Y284599 /
    timestamps 284599 / feature_names 225 / label_names 51
A76A5AF5-5A93-4CF2-A16E-62353BB70E8A.features_labels.csv.gz |X292119-Y292119 /
    timestamps 292119 / feature_names 225 / label_names 51
B09E373F-8A54-44C8-895B-0039390B859F.features_labels.csv.gz |X300253-Y300253 /
    timestamps 300253 / feature_names 225 / label_names 51
B7F9D634-263E-4A97-87F9-6FFB4DDCB36C.features_labels.csv.gz |X309636-Y309636 /
    timestamps 309636 / feature_names 225 / label_names 51
B9724848-C7E2-45F4-9B3F-A1F38D864495.features_labels.csv.gz |X317262-Y317262 /
    timestamps 317262 / feature_names 225 / label_names 51
BE3CA5A6-A561-4BBD-B7C9-5DF6805400FC.features_labels.csv.gz |X325571-Y325571 /
    timestamps 325571 / feature_names 225 / label_names 51
BEF6C611-50DA-4971-A040-87FB979F3FC1.features_labels.csv.gz |X329022-Y329022 /
    timestamps 329022 / feature_names 225 / label_names 51
C48CE857-A0DD-4DDB-BEA5-3A25449B2153.features_labels.csv.gz |X334114-Y334114 /
    timestamps 334114 / feature_names 225 / label_names 51
```

```
CA820D43-E5E2-42EF-9798-BE56F776370B.features_labels.csv.gz |X341979-Y341979 /
    timestamps 341979 / feature_names 225 / label_names 51
CCAF77F0-FABB-4F2F-9E24-D56AD0C5A82F.features_labels.csv.gz |X350451-Y350451 /
    timestamps 350451 / feature_names 225 / label_names 51
CDA3BBF7-6631-45E8-85BA-EEB416B32A3C.features_labels.csv.gz |X353311-Y353311 /
    timestamps 353311 / feature_names 225 / label_names 51
CF722AA9-2533-4E51-9FEB-9EAC84EE9AAC.features_labels.csv.gz |X356926-Y356926 /
    timestamps 356926 / feature_names 225 / label_names 51
D7D20E2E-FC78-405D-B346-DBD3FD8FC92B.features_labels.csv.gz |X363136-Y363136 /
    timestamps 363136 / feature_names 225 / label_names 51
E65577C1-8D5D-4F70-AF23-B3ADB9D3DBA3.features_labels.csv.gz |X366577-Y366577 /
    timestamps 366577 / feature_names 225 / label_names 51
ECECC2AB-D32F-4F90-B74C-E12A1C69BBE2.features_labels.csv.gz |X370107-Y370107 /
    timestamps 370107 / feature_names 225 / label_names 51
F50235E0-DD67-4F2A-B00B-1F31ADA998B9.features_labels.csv.gz |X372373-Y372373 /
    timestamps 372373 / feature_names 225 / label_names 51
end
layer sizes
Fitting 3 folds for each of 30 candidates, totalling 90 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 4 concurrent workers.
[Parallel(n_jobs=-1)]: Done  42 tasks      | elapsed: 27.3min
[Parallel(n_jobs=-1)]: Done  90 out of  90 | elapsed: 53.6min finished
F-Score: 0.92
{'solver': 'adam', 'learning_rate': 'adaptive', 'hidden_layer_sizes': (150, 100, 50), '
    alpha': 0.03, 'activation': 'tanh'}
--- Training Metrics ---
Samples: 8766
Label: Walking
TP: 18835024
TN: 12425049
Sensors: ['Aud', 'Acc', 'Gyro', 'Loc']
Features: 7865 features
--- Performance Metrics ---
Accuracy: 0.51
Recall(sensitivity): 0.55
Precision: 0.55
Specificity: 0.45
F-Score: 0.84
--- Confusion Matrix CLASS 2 ---
[[2892  657]
 [ 609 3707]]
--- Training Metrics ---
Samples: 4973
Label: Walking
TP: 167919
TN: 2718675
Sensors: ['Aud', 'Acc', 'Gyro', 'Loc']
Features: 2158 features
--- Performance Metrics ---
Accuracy: 0.62
Recall(sensitivity): 0.35
Precision: 0.10
Specificity: 0.65
```

```
F-Score: 0.80
--- Confusion Matrix CLASS 2 ---
[[1400  535]
 [   5  218]]
layer sizes
Fitting 3 folds for each of 30 candidates, totalling 90 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 4 concurrent workers.
[Parallel(n_jobs=-1)]: Done  42 tasks      | elapsed: 25.6min
[Parallel(n_jobs=-1)]: Done  90 out of  90 | elapsed: 42.9min finished
F-Score: 0.82
{'solver': 'adam', 'learning_rate': 'constant', 'hidden_layer_sizes': (300,), 'alpha':
    0.001, 'activation': 'relu'}
--- Training Metrics ---
Samples: 8766
Label: Walking
TP: 21205444
TN: 10764288
Sensors: ['Aud']
Features: 7898 features
--- Performance Metrics ---
Accuracy: 0.51
Recall(sensitivity): 0.61
Precision: 0.56
Specificity: 0.39
F-Score: 0.76
--- Confusion Matrix CLASS 2 ---
[[2366 1138]
 [ 706 3688]]
--- Training Metrics ---
Samples: 4973
Label: Walking
TP: 233481
TN: 2149785
Sensors: ['Aud']
Features: 2158 features
--- Performance Metrics ---
Accuracy: 0.51
Recall(sensitivity): 0.49
Precision: 0.10
Specificity: 0.51
F-Score: 0.69
--- Confusion Matrix CLASS 2 ---
[[1109  826]
 [   2  221]]
```

C.2 Multi-label Classifier

```
00EABED2-271D-49D8-B599-1D4A09240601.features_labels.csv.gz |X2287-Y2287 / timestamps 2287 /
     feature_names 225 / label_names 51
098A72A5-E3E5-4F54-A152-BBDA0DF7B694.features_labels.csv.gz |X9100-Y9100 / timestamps 9100 /
     feature_names 225 / label_names 51
0A986513-7828-4D53-AA1F-E02D6DF9561B.features_labels.csv.gz |X13060-Y13060 / timestamps
    13060 / feature_names 225 / label_names 51
0BFC35E2-4817-4865-BFA7-764742302A2D.features_labels.csv.gz |X16168-Y16168 / timestamps
```

```
    16168 / feature_names 225 / label_names 51
0E6184E1-90C0-48EE-B25A-F1ECB7B9714E.features_labels.csv.gz |X23689-Y23689 / timestamps
    23689 / feature_names 225 / label_names 51
1155FF54-63D3-4AB2-9863-8385D0BD0A13.features_labels.csv.gz |X26374-Y26374 / timestamps
    26374 / feature_names 225 / label_names 51
11B5EC4D-4133-4289-B475-4E737182A406.features_labels.csv.gz |X35219-Y35219 / timestamps
    35219 / feature_names 225 / label_names 51
136562B6-95B2-483D-88DC-065F28409FD2.features_labels.csv.gz |X41437-Y41437 / timestamps
    41437 / feature_names 225 / label_names 51
1538C99F-BA1E-4EFB-A949-6C7C47701B20.features_labels.csv.gz |X47986-Y47986 / timestamps
    47986 / feature_names 225 / label_names 51
1DBB0F6F-1F81-4A50-9DF4-CD62ACFA4842.features_labels.csv.gz |X55361-Y55361 / timestamps
    55361 / feature_names 225 / label_names 51
24E40C4C-A349-4F9F-93AB-01D00FB994AF.features_labels.csv.gz |X60132-Y60132 / timestamps
    60132 / feature_names 225 / label_names 51
27E04243-B138-4F40-A164-F40B60165CF3.features_labels.csv.gz |X65059-Y65059 / timestamps
    65059 / feature_names 225 / label_names 51
2C32C23E-E30C-498A-8DD2-0EFB9150A02E.features_labels.csv.gz |X73575-Y73575 / timestamps
    73575 / feature_names 225 / label_names 51
33A85C34-CFE4-4732-9E73-0A7AC861B27A.features_labels.csv.gz |X79747-Y79747 / timestamps
    79747 / feature_names 225 / label_names 51
3600D531-0C55-44A7-AE95-A7A38519464E.features_labels.csv.gz |X84950-Y84950 / timestamps
    84950 / feature_names 225 / label_names 51
40E170A7-607B-4578-AF04-F021C3B0384A.features_labels.csv.gz |X92599-Y92599 / timestamps
    92599 / feature_names 225 / label_names 51
481F4DD2-7689-43B9-A2AA-C8772227162B.features_labels.csv.gz |X99290-Y99290 / timestamps
    99290 / feature_names 225 / label_names 51
4E98F91F-4654-42EF-B908-A3389443F2E7.features_labels.csv.gz |X102540-Y102540 / timestamps
    102540 / feature_names 225 / label_names 51
4FC32141-E888-4BFF-8804-12559A491D8C.features_labels.csv.gz |X107519-Y107519 / timestamps
    107519 / feature_names 225 / label_names 51
5119D0F8-FCA8-4184-A4EB-19421A40DE0D.features_labels.csv.gz |X114136-Y114136 / timestamps
    114136 / feature_names 225 / label_names 51
5152A2DF-FAF3-4BA8-9CA9-E66B32671A53.features_labels.csv.gz |X120753-Y120753 / timestamps
    120753 / feature_names 225 / label_names 51
59818CD2-24D7-4D32-B133-24C2FE3801E5.features_labels.csv.gz |X126700-Y126700 / timestamps
    126700 / feature_names 225 / label_names 51
59EEFAE0-DEB0-4FFF-9250-54D2A03D0CF2.features_labels.csv.gz |X134242-Y134242 / timestamps
    134242 / feature_names 225 / label_names 51
5EF64122-B513-46AE-BCF1-E62AAC285D2C.features_labels.csv.gz |X138153-Y138153 / timestamps
    138153 / feature_names 225 / label_names 51
61359772-D8D8-480D-B623-7C636EAD0C81.features_labels.csv.gz |X144232-Y144232 / timestamps
    144232 / feature_names 225 / label_names 51
61976C24-1C50-4355-9C49-AAE44A7D09F6.features_labels.csv.gz |X152962-Y152962 / timestamps
    152962 / feature_names 225 / label_names 51
665514DE-49DC-421F-8DCB-145D0B2609AD.features_labels.csv.gz |X162129-Y162129 / timestamps
    162129 / feature_names 225 / label_names 51
74B86067-5D4B-43CF-82CF-341B76BEA0F4.features_labels.csv.gz |X169427-Y169427 / timestamps
    169427 / feature_names 225 / label_names 51
78A91A4E-4A51-4065-BDA7-94755F0BB3BB.features_labels.csv.gz |X181423-Y181423 / timestamps
    181423 / feature_names 225 / label_names 51
797D145F-3858-4A7F-A7C2-A4EB721E133C.features_labels.csv.gz |X185016-Y185016 / timestamps
    185016 / feature_names 225 / label_names 51
```

```
7CE37510-56D0-4120-A1CF-0E23351428D2.features_labels.csv.gz |X194777-Y194777 / timestamps
    194777 / feature_names 225 / label_names 51
7D9BB102-A612-4E2A-8E22-3159752F55D8.features_labels.csv.gz |X196377-Y196377 / timestamps
    196377 / feature_names 225 / label_names 51
8023FE1A-D3B0-4E2C-A57A-9321B7FC755F.features_labels.csv.gz |X205566-Y205566 / timestamps
    205566 / feature_names 225 / label_names 51
806289BC-AD52-4CC1-806C-0CDB14D65EB6.features_labels.csv.gz |X214808-Y214808 / timestamps
    214808 / feature_names 225 / label_names 51
81536B0A-8DBF-4D8A-AC24-9543E2E4C8E0.features_labels.csv.gz |X221215-Y221215 / timestamps
    221215 / feature_names 225 / label_names 51
83CF687B-7CEC-434B-9FE8-00C3D5799BE6.features_labels.csv.gz |X230754-Y230754 / timestamps
    230754 / feature_names 225 / label_names 51
86A4F379-B305-473D-9D83-FC7D800180EF.features_labels.csv.gz |X241492-Y241492 / timestamps
    241492 / feature_names 225 / label_names 51
96A358A0-FFF2-4239-B93E-C7425B901B47.features_labels.csv.gz |X247311-Y247311 / timestamps
    247311 / feature_names 225 / label_names 51
9759096F-1119-4E19-A0AD-6F16989C7E1C.features_labels.csv.gz |X257270-Y257270 / timestamps
    257270 / feature_names 225 / label_names 51
99B204C0-DD5C-4BB7-83E8-A37281B8D769.features_labels.csv.gz |X263308-Y263308 / timestamps
    263308 / feature_names 225 / label_names 51
9DC38D04-E82E-4F29-AB52-B476535226F2.features_labels.csv.gz |X272994-Y272994 / timestamps
    272994 / feature_names 225 / label_names 51
A5A30F76-581E-4757-97A2-957553A2C6AA.features_labels.csv.gz |X274661-Y274661 / timestamps
    274661 / feature_names 225 / label_names 51
A5CDF89D-02A2-4EC1-89F8-F534FDABDD96.features_labels.csv.gz |X280701-Y280701 / timestamps
    280701 / feature_names 225 / label_names 51
A7599A50-24AE-46A6-8EA6-2576F1011D81.features_labels.csv.gz |X284599-Y284599 / timestamps
    284599 / feature_names 225 / label_names 51
A76A5AF5-5A93-4CF2-A16E-62353BB70E8A.features_labels.csv.gz |X292119-Y292119 / timestamps
    292119 / feature_names 225 / label_names 51
B09E373F-8A54-44C8-895B-0039390B859F.features_labels.csv.gz |X300253-Y300253 / timestamps
    300253 / feature_names 225 / label_names 51
B7F9D634-263E-4A97-87F9-6FFB4DDCB36C.features_labels.csv.gz |X309636-Y309636 / timestamps
    309636 / feature_names 225 / label_names 51
B9724848-C7E2-45F4-9B3F-A1F38D864495.features_labels.csv.gz |X317262-Y317262 / timestamps
    317262 / feature_names 225 / label_names 51
BE3CA5A6-A561-4BBD-B7C9-5DF6805400FC.features_labels.csv.gz |X325571-Y325571 / timestamps
    325571 / feature_names 225 / label_names 51
BEF6C611-50DA-4971-A040-87FB979F3FC1.features_labels.csv.gz |X329022-Y329022 / timestamps
    329022 / feature_names 225 / label_names 51
C48CE857-A0DD-4DDB-BEA5-3A25449B2153.features_labels.csv.gz |X334114-Y334114 / timestamps
    334114 / feature_names 225 / label_names 51
CA820D43-E5E2-42EF-9798-BE56F776370B.features_labels.csv.gz |X341979-Y341979 / timestamps
    341979 / feature_names 225 / label_names 51
CCAF77F0-FABB-4F2F-9E24-D56AD0C5A82F.features_labels.csv.gz |X350451-Y350451 / timestamps
    350451 / feature_names 225 / label_names 51
CDA3BBF7-6631-45E8-85BA-EEB416B32A3C.features_labels.csv.gz |X353311-Y353311 / timestamps
    353311 / feature_names 225 / label_names 51
CF722AA9-2533-4E51-9FEB-9EAC84EE9AAC.features_labels.csv.gz |X356926-Y356926 / timestamps
    356926 / feature_names 225 / label_names 51
D7D20E2E-FC78-405D-B346-DBD3FD8FC92B.features_labels.csv.gz |X363136-Y363136 / timestamps
    363136 / feature_names 225 / label_names 51
E65577C1-8D5D-4F70-AF23-B3ADB9D3DBA3.features_labels.csv.gz |X366577-Y366577 / timestamps
```

```
    366577 / feature_names 225 / label_names 51
ECECC2AB-D32F-4F90-B74C-E12A1C69BBE2.features_labels.csv.gz |X370107-Y370107 / timestamps
    370107 / feature_names 225 / label_names 51
F50235E0-DD67-4F2A-B00B-1F31ADA998B9.features_labels.csv.gz |X372373-Y372373 / timestamps
    372373 / feature_names 225 / label_names 51
end
layer sizes
Fitting 3 folds for each of 30 candidates, totalling 90 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 4 concurrent workers.
F-Score: 0.60
{'solver': 'adam', 'learning_rate': 'invscaling', 'hidden_layer_sizes': (151, 52, 26), '
    alpha': 0.07, 'activation': 'tanh'}
----------
----------
--- Training Metrics ---
Samples: 7891
Label: 2:20
TP: 10706
TN: 132399
Sensors: ['Aud', 'Acc', 'Gyro', 'Loc']
Features: 7891 features
--- Performance Metrics ---
Accuracy: 0.95
Recall(sensitivity): 0.73
Precision: 0.79
Specificity: 0.98
F-Score: 0.75
--- Confusion Matrix CLASS 2 ---
[[2874  633]
 [ 709 3675]]
--- Confusion Matrix CLASS 3 ---
[[7876    0]
 [  15    0]]
--- Confusion Matrix CLASS 4 ---
[[7812    2]
 [  45   32]]
--- Confusion Matrix CLASS 5 ---
[[6639  214]
 [ 178  860]]
--- Confusion Matrix CLASS 6 ---
[[7822    5]
 [  60    4]]
--- Confusion Matrix CLASS 7 ---
[[7801    8]
 [  71   11]]
--- Confusion Matrix CLASS 8 ---
[[7811    0]
 [  80    0]]
--- Confusion Matrix CLASS 9 ---
[[7102  132]
 [ 292  365]]
--- Confusion Matrix CLASS 10 ---
[[4375  585]
```

```
 [ 603 2328]]
--- Confusion Matrix CLASS 11 ---
[[6234  362]
 [ 532  763]]
--- Confusion Matrix CLASS 12 ---
[[7795   14]
 [  65   17]]
--- Confusion Matrix CLASS 13 ---
[[7867    0]
 [  24    0]]
--- Confusion Matrix CLASS 14 ---
[[7757   36]
 [  72   26]]
--- Confusion Matrix CLASS 15 ---
[[7853    1]
 [  36    1]]
--- Confusion Matrix CLASS 16 ---
[[5155  511]
 [ 411 1814]]
--- Confusion Matrix CLASS 17 ---
[[7861    0]
 [  30    0]]
--- Confusion Matrix CLASS 18 ---
[[6297  266]
 [ 631  697]]
--- Confusion Matrix CLASS 19 ---
[[7757    5]
 [  85   44]]
--- Confusion Matrix CLASS 20 ---
[[7711   31]
 [  80   69]]
----------
----------
--- Training Metrics ---
Samples: 2158
Label: 2:20
TP: 537
TN: 36996
Sensors: ['Aud', 'Acc', 'Gyro', 'Loc']
Features: 2158 features
--- Performance Metrics ---
Accuracy: 0.92
Recall(sensitivity): 0.43
Precision: 0.16
Specificity: 0.93
F-Score: 0.42
--- Confusion Matrix CLASS 2 ---
[[1374  561]
 [   7  216]]
--- Confusion Matrix CLASS 3 ---
[[2158]]
--- Confusion Matrix CLASS 4 ---
[[2156    2]
```

```
 [   0    0]]
--- Confusion Matrix CLASS 5 ---
[[1998  160]
 [   0    0]]
--- Confusion Matrix CLASS 6 ---
[[2041  117]
 [   0    0]]
--- Confusion Matrix CLASS 7 ---
[[2155    3]
 [   0    0]]
--- Confusion Matrix CLASS 8 ---
[[2158]]
--- Confusion Matrix CLASS 9 ---
[[1685  473]
 [   0    0]]
--- Confusion Matrix CLASS 10 ---
[[1502  656]
 [   0    0]]
--- Confusion Matrix CLASS 11 ---
[[2024  134]
 [   0    0]]
--- Confusion Matrix CLASS 12 ---
[[2125   33]
 [   0    0]]
--- Confusion Matrix CLASS 13 ---
[[2158]]
--- Confusion Matrix CLASS 14 ---
[[2011    0]
 [  89   58]]
--- Confusion Matrix CLASS 15 ---
[[2154    4]
 [   0    0]]
--- Confusion Matrix CLASS 16 ---
[[1731  427]
 [   0    0]]
--- Confusion Matrix CLASS 17 ---
[[2158]]
--- Confusion Matrix CLASS 18 ---
[[1096  184]
 [ 615  263]]
--- Confusion Matrix CLASS 19 ---
[[2154    4]
 [   0    0]]
--- Confusion Matrix CLASS 20 ---
[[2158]]
layer sizes

 ------ REMOVING  --- SUPPLEMENTARY  --- DATA --- ---

F-Score: 0.42
----------
----------
--- Training Metrics ---
```

```
Samples: 7903
Label: 2:20
TP: 9233
TN: 132687
Sensors: ['Aud']
Features: 7903 features
--- Performance Metrics ---
Accuracy: 0.95
Recall(sensitivity): 0.63
Precision: 0.76
Specificity: 0.98
F-Score: 0.67
--- Confusion Matrix CLASS 2 ---
[[2364 1126]
 [ 691 3722]]
--- Confusion Matrix CLASS 3 ---
[[7891    0]
 [  12    0]]
--- Confusion Matrix CLASS 4 ---
[[7828    2]
 [  68    5]]
--- Confusion Matrix CLASS 5 ---
[[6744  160]
 [ 306  693]]
--- Confusion Matrix CLASS 6 ---
[[7848    1]
 [  47    7]]
--- Confusion Matrix CLASS 7 ---
[[7822    0]
 [  81    0]]
--- Confusion Matrix CLASS 8 ---
[[7810    0]
 [  93    0]]
--- Confusion Matrix CLASS 9 ---
[[7133  100]
 [ 453  217]]
--- Confusion Matrix CLASS 10 ---
[[4610  489]
 [ 833 1971]]
--- Confusion Matrix CLASS 11 ---
[[6297  255]
 [ 807  544]]
--- Confusion Matrix CLASS 12 ---
[[7813    2]
 [  88    0]]
--- Confusion Matrix CLASS 13 ---
[[7881    0]
 [  22    0]]
--- Confusion Matrix CLASS 14 ---
[[7798    7]
 [  92    6]]
--- Confusion Matrix CLASS 15 ---
[[7865    0]
```

```
 [  38    0]]
--- Confusion Matrix CLASS 16 ---
[[5344  405]
 [ 658 1496]]
--- Confusion Matrix CLASS 17 ---
[[7860    0]
 [  43    0]]
--- Confusion Matrix CLASS 18 ---
[[6274  314]
 [ 813  502]]
--- Confusion Matrix CLASS 19 ---
[[7784    8]
 [ 102    9]]
--- Confusion Matrix CLASS 20 ---
[[7721   12]
 [ 109   61]]
----------
----------
--- Training Metrics ---
Samples: 2158
Label: 2:20
TP: 380
TN: 37523
Sensors: ['Aud']
Features: 2158 features
--- Performance Metrics ---
Accuracy: 0.92
Recall(sensitivity): 0.30
Precision: 0.15
Specificity: 0.94
F-Score: 0.29
--- Confusion Matrix CLASS 2 ---
[[1307  628]
 [  12  211]]
--- Confusion Matrix CLASS 3 ---
[[2158]]
--- Confusion Matrix CLASS 4 ---
[[2158]]
--- Confusion Matrix CLASS 5 ---
[[1845  313]
 [   0    0]]
--- Confusion Matrix CLASS 6 ---
[[2127   31]
 [   0    0]]
--- Confusion Matrix CLASS 7 ---
[[2158]]
--- Confusion Matrix CLASS 8 ---
[[2158]]
--- Confusion Matrix CLASS 9 ---
[[2139   19]
 [   0    0]]
--- Confusion Matrix CLASS 10 ---
[[1389  769]
```

```
 [   0    0]]
--- Confusion Matrix CLASS 11 ---
[[2156    2]
 [   0    0]]
--- Confusion Matrix CLASS 12 ---
[[2149    9]
 [   0    0]]
--- Confusion Matrix CLASS 13 ---
[[2158]]
--- Confusion Matrix CLASS 14 ---
[[2009    2]
 [ 133   14]]
--- Confusion Matrix CLASS 15 ---
[[2158]]
--- Confusion Matrix CLASS 16 ---
[[1780  378]
 [   0    0]]
--- Confusion Matrix CLASS 17 ---
[[2158]]
--- Confusion Matrix CLASS 18 ---
[[1201   79]
 [ 723  155]]
--- Confusion Matrix CLASS 19 ---
[[2157    1]
 [   0    0]]
--- Confusion Matrix CLASS 20 ---
[[2158]]
```

D Dataset Feature Importance

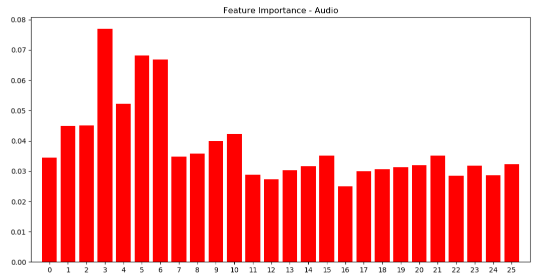

Figure 20: A Random Tree Classifier was used to evaluate feature importance, during the design phase of the main classifier. Features were Audio (26 bins)

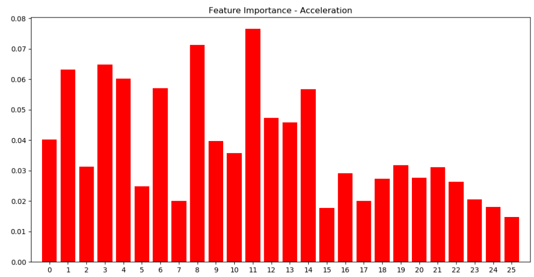

Figure 21: A Random Tree Classifier was used to evaluate feature importance, during the design phase of the main classifier. Features were Accelerometer(26)

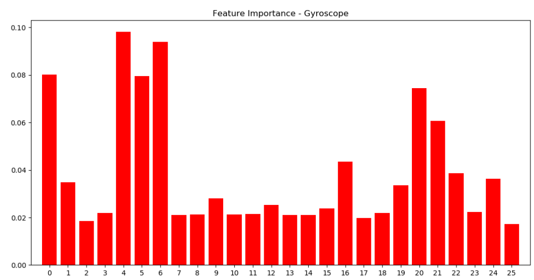

Figure 22: A Random Tree Classifier was used to evaluate feature importance, during the design phase of the main classifier. Features were Gyroscope(26)

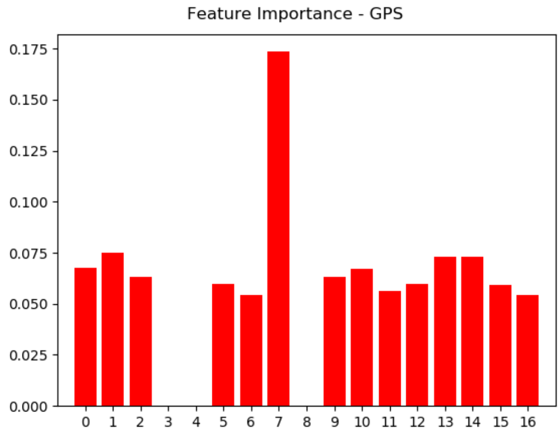

Figure 23: A Random Tree Classifier was used to evaluate feature importance, during the design phase of the main classifier. Features were GPS (16)

E Trade-Review of "Online vs Offline" processing

Online vs Offline Data Fusion & Data Analytics This document will compare the costs, benefits and risks in designing an "online" (run on the device, in either real time or batched process) verses "offline" (processed on external, significantly better hardware. Again either real time or batched).

Online/Onboard: The platform would need to run 2 highly computationally expensive tasks:

1. Take recorded data and fuse it (subject to further research about best practice)
2. Run a compressed inference model of whatever data analytics models are chosen

The benefits of this architecture is to streamline the overall project & the project designs, where the project becomes a black box that takes in onboard sensor data and produces a final, processed file.

Example Costs:

- $1000-3000 Hardware costs
- Large design costs as online machine learning inference has a high performance requirement, and working to produce a solution that can do it where performance is an issue carries significant risk

Example Performance:

- Weak or slow. Real time processing not likely in this option
- Fidelity in data & analytics would suffer, but it may be fit for use depending on use case

Offline/Offboard:

This platform breaks the larger system into

1. A data recording system
2. A separate processing system

The benefits of this are to optimise each system for its own purpose. E.g. a data collection embedded solution could be optimised for battery and data fidelity.

The separate processing system could also batch process information on either cloud or local dedicated hardware. This would likely result in the highest and quickest performance.

The cost and risk associated with this option, is that 2 separate solutions would need to be developed to support the original aim of the project.

Example Costs:

- $1000-10000 Hardware costs, in proof-of-concept hardware alone.
  - This cost covers a much cheaper data acquisition unit, and a higher cost offline processing unit
- Proven platform with limited "unknown" risks. Whilst we're not sure if the proof-of-concept will be fit for purpose, there are known examples of this architecture being used.

Example Performance:

- The best option. Real time processing could be an option, but not required.
- Fidelity in data & analytics would suffer, but it may be fit for use depending on use case

Pugh Matrix

Requirements	Weight	Online/Onboard	Offline/Offboard
Cost	3	+	-
Confidence in Data	7	-	+
Performance (Speed)	6	-	+
Footprint/Weight	10	-	+
Risks	9	-	+

It is clear that Offline processing offers less risk and more capacity to achieve what we want, so it is my (Chris) recommendation to go with 2 discrete systems (data acquisition, data processing)

F GridSearch Attempt

This shows how a Grid Search was used for hyperparameter optimisation. In particular, this is an example of an iterative process where the grid was manually narrowed and precision increase.

```
parameters={'hidden_layer_sizes': [(145,), (147,), (149,), (151,), (153)],
            'alpha': [0.09, 0.11, 0.13, 0.15, 0.17, 0.19]}
mlpc = MLPClassifier(verbose=False, early_stopping=True, learning_rate='adaptive',max_iter=1000)
clf = GridSearchCV(estimator=mlpc, scoring='f1',param_grid=parameters,n_jobs=-1,verbose=1,cv=3);
clf.fit(X_train,y);
print(clf.score(X_train, y))
print(clf.best_params_)

Fitting 3 folds for each of 30 candidates, totalling 90 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 4 concurrent workers.
[Parallel(n_jobs=-1)]: Done  42 tasks      | elapsed:  7.8min
[Parallel(n_jobs=-1)]: Done  90 out of  90 | elapsed: 14.4min finished
0.8249400479616308
{'alpha': 0.11, 'hidden_layer_sizes': (151,)}
```

G Table of ExtraSensory Labels

Table 11: ExtraSensory Labels [22]

#	Label Description
1	LYING_DOWN
2	SITTING
3	FIX_walking
4	FIX_running
5	BICYCLING
6	SLEEPING
7	LAB_WORK
8	IN_CLASS
9	IN_A_MEETING
10	LOC_main_workplace
11	OR_indoors
12	OR_outside
13	IN_A_CAR
14	ON_A_BUS
15	DRIVE_-_I_M_THE_DRIVER
16	DRIVE_-_I_M_A_PASSENGER
17	LOC_home
18	FIX_restaurant
19	PHONE_IN_POCKET
20	OR_exercise
21	COOKING
22	SHOPPING
23	STROLLING
24	DRINKING__ALCOHOL_
25	BATHING_-_SHOWER
26	CLEANING
27	DOING_LAUNDRY
28	WASHING_DISHES
29	WATCHING_TV
30	SURFING_THE_INTERNET
31	AT_A_PARTY
32	AT_A_BAR
33	LOC_beach
34	SINGING
35	TALKING
36	COMPUTER_WORK
37	EATING
38	TOILET
39	GROOMING
40	DRESSING
41	AT_THE_GYM
42	STAIRS_-_GOING_UP
43	STAIRS_-_GOING_DOWN
44	ELEVATOR
45	OR_standing
46	AT_SCHOOL
47	PHONE_IN_HAND
48	PHONE_IN_BAG
49	PHONE_ON_TABLE
50	WITH_CO-WORKERS
51	WITH_FRIENDS