School of Engineering, Discipline of Mechanical Engineering and Mechatronics, pp. 1–65

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

check all flagforreview [FLAGGED FOR REVIEW]
check all citationneeded [citation needed]
Tense: Past, Future etc. Search "was", "is" "ed" etc. Check contraction. it's, don't etc. check for our, we, I, they're, your, you, etc. check for st-ray - inca-se the-y block sh-t check ref and replace with fref, tref, aref etc. fix header to og check
refs{} for whether the fullstop is before or after review the use of section vs paragraph vs subsection etc. As per the FYP Handbook:
I did:

- I learnt and prototyped two machine learning solutions that takes audio and physical data (GPS, accelerometer, gyroscope) to compare whether the supplementary data could improve the classification of an acoustic event, as measured by the F1 score.
- I learnt and prototyped a pipeline to record data, process it, classify it, and output it.
- I used my mechatronics subject matter learnings to best apply machine learning to this problem. This includes Mechatronics Design (trade off and evaluations from MCHA3000), sound preprocessing, preprocessing Inertial Measuring Unit data, and how Audio waveforms can be treated as energy, and statistics from MECH2450/MCHA3900. [FLAGGED FOR REVIEW]
- I learnt and used system engineering techniques to elicit requirements from my university and work stakeholders

I helped:

- I managed and worked with another employee to implement a proof of concept app to record data as an input into this project. This involved using my software engineering skills and project management skills to 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 classifiers ability to classify an "Acoustic Event" in an Acoustic Event Detection / Classification (AED/C) problem. The motivation behind this was to improve AED/C without onerous microphone requirements, to enable more widespread commercial use of AED/C.

Section 2 of the project begins by evaluating the research problem for a car use case, and eliciting the design requirements through system engineering principles.

Section 3 reviews current research. Presently the field is expanding with broader advances in the wider machine learning community, however AED/C remains a non-trivial problem that doesn't have a viable solution yet. Comparatively to the wider machine learning community, Acoustic Event Detection is very limited in active researchers. This specific issue hasn't been explicitly answered.

Section 4 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 against the data. For this project:

- The desired output is a label (or labels) detailing the Acoustic Event in the sample (e.g. The labels found in Appendix G like 'Walking').
- The required data is the ExtraSensory dataset, or a custom dataset as recorded by the project's App.
- The score selected is the F1 score. This is due to it being an industry standard.
- The classifier chosen is the MLPClassifier which is a feedforward neural network, trained via backpropagation.
- Hyperparameters of the MLPClassifier are chosen via a Random Search Cross Validation.
- Training and Validation was run with Early Stopping using a 10% validation set, using 59 out of the 60 ExtraSensory users.
- Final testing is done with the 60$^{\text{th}}$ user.

Section 5 the desired results are a 10.5% and 11.9% relative improvement in F1 score for a Binary and Multi-label classifier respectively. This is calculated through 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. The output of the classifiers are also analysed and found to be valid classifiers for real-world use. [FLAGGED FOR REVIEW]

Section 7 evaluates and discusses limitations of the results. It will discuss how multi-sensor AED/C is effective but requires more datasets to test smaller, more specific tasks. [FLAGGED FOR REVIEW] It also discusses why the result will be valid in other projects.

Section 8 will present how these results will contribute to the field, and suggested future work into new datasets with a greater number of labels.

Acknowledgements

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

I'd also like to thank each and every mechatronics Academic staff member at the University of Newcastle . The teaching staff care about their students, and I want that to grow.

Contents

1 Introduction

* Introduce field * discuss what the current work is like in field, what current results are, maybe what can be determined from data presently etc * What is the problem/question * nitty gridd of the debrief problem? * Why does my question help solve this problem/answer this question

Introduction to the Field from a Mechatronics Perspective machine learning will be 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 is that of almost akin to that of 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)*?

This project aims to answer the question "[What is] the effect of supplementary data on acoustic event classification through machine learning". To do this, 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. Additionally, the output of the classifiers will be analysed for "validity". Does the classifier successfully detect and classify an Acoustic Event?

The desired results are a demonstrable difference (or lack thereof) in the F1 score, after comparing a classifier with and without supplementary data.

Motivation for the problem The capacity to identify a physical event in the real world using sound would open the door to a whole new realm of data. The potential implications and applications are limitless; assistance and closed captioning for the hearing impaired, improved security closed loop TV, increased transparency and documentation of a ride sharing trip, and early detection of potential accidents or medical emergencies to name a few. The primary focus of the new potential is to review AEC/D as a method of debriefing across multiple industries.

Debreifing (or After Action Report) Debriefing is effectively an education tool used to educate an individual, a group or record a more holistic history of an event, for purposes of performance feedback Kaliner (2013). The formal debrief originated from the World War 2, with SLA Marshell's "interviews after combat" Morrison (1999). 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. HEADQUARTERS (1993)"

This is a key tool in training after any group activity, and has been extended to many industry, most notably in the training of Doctors Johnson Pivec (2011). Wikipedia (2019)[FLAGGED FOR REVIEW]. Traditionally this is done through a collection of primary sources (where available) and secondary sources. In the information age, these primary sources have expanded to include large datasets, recordings and other digital forensics. This new and increasing range of primary sources provides the potential to significantly improve the steps of debriefing, and ultimately improve debriefing outcomes Psy (2012).

2 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]" Limited (2018). This perspective was informed by ISO/IEC/IEEE 15288 for Standardization (2015) and Burge Hughes Walsh's system engineering Toolbox Limited (2015). The tools adapted and applied throughout this section are primarily derived from that "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.

2.1 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. The current "18 Words" is the following: "[The project is] a portable proprietary format compliant track file data recording system coupled with offline processing to log specific physical events in proximity that occurred during recording."

2.2 Tree Diagram

With the new understanding of what the project is, a Tree Diagram was drafted to explore missed requirements, hidden modules and other aspects of the project not yet considered. Figure 1 shows a Tree Diagram breakdown for this FYP.

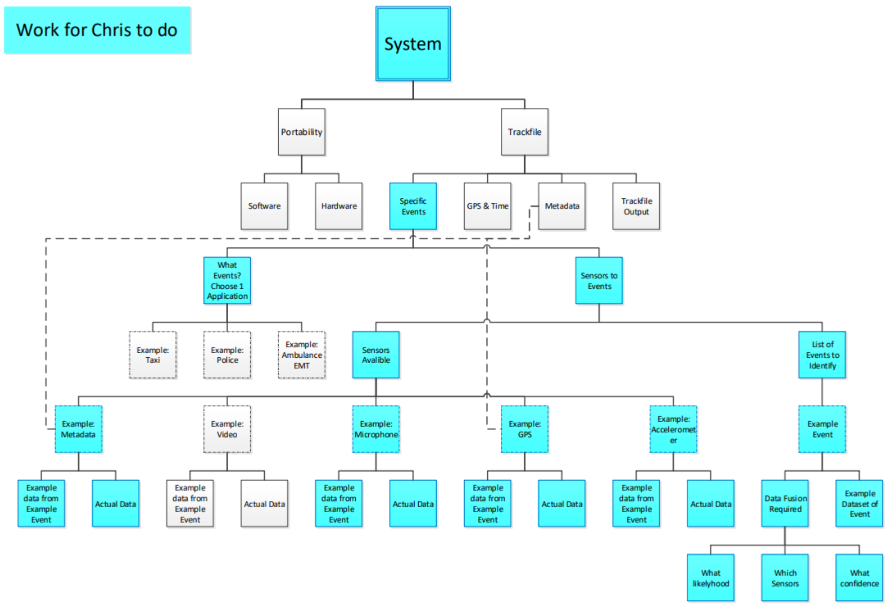

Figure 1: A tree diagram for the project, being used as a method of allocating work

2.3 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 2.

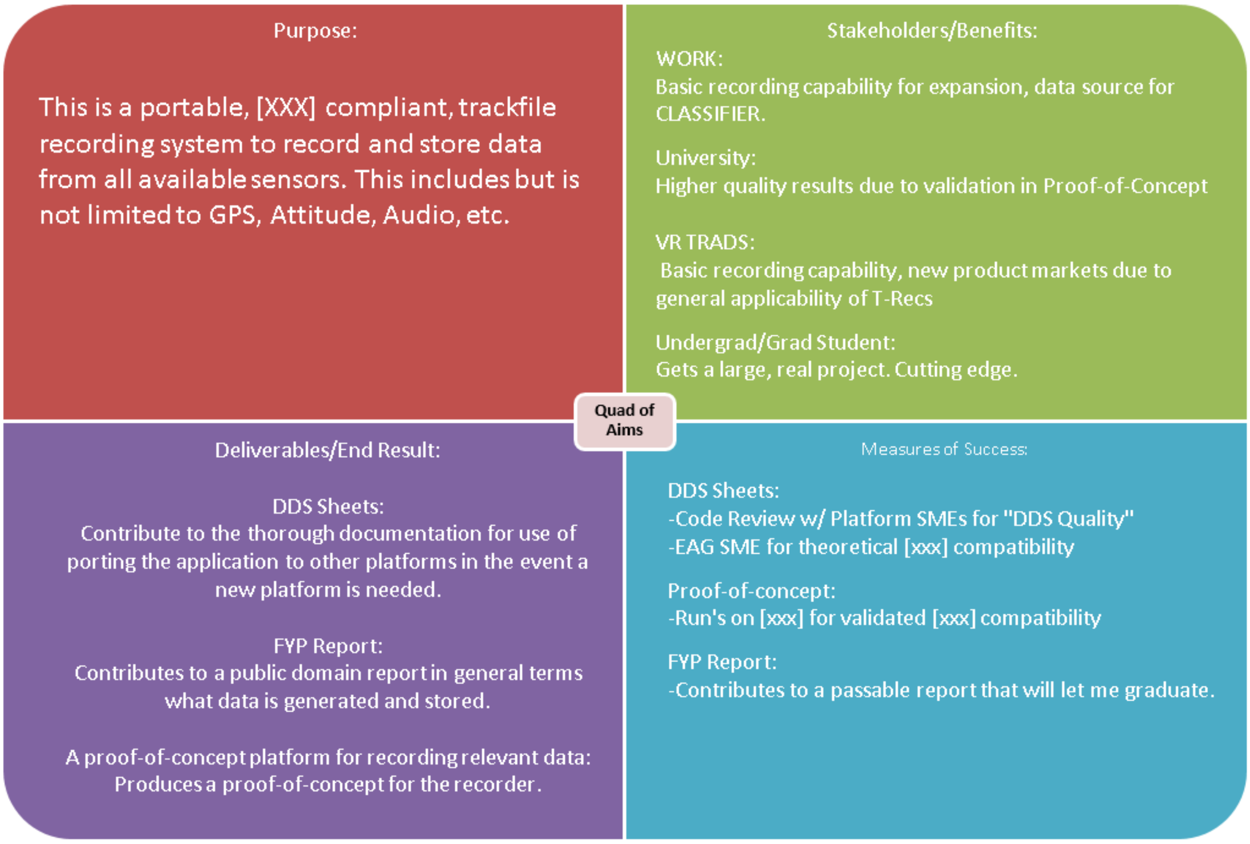

Figure 2: A Quad of Aims for the project might be A3 in size and would have the relevant information embedded. It may also be completed on a whiteboard.

This is typically done to instigate early works on a project.

2.4 Input Output Analysis

The Input Output Analysis of the system informs the bounds and requirements to operate the system. In this situation, it helped consider the full scope of the project; this includes the technical and non-technical aspects of undertaking the MECH4841 Project as shown in Figure 3.

2.5 Affinity Diagram

An Affinity Diagram helps to solidify the high level, vague ideas, requirements and tests into a more detailed view. There is 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, and subsequently splitting the system into smaller modules. Our primary system had 2 key sub-systems: The recording, and the processing systems. An Affinity Diagram of the top-level system was created, and the 2 affinity diagrams of the sub-systems were informed and made from this. After this was complete, previous work was updated to reflect these changes. Figure 4. [citation needed]

2.6 Systems Map

A Systems Map takes 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 great effect to measure and estimate

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

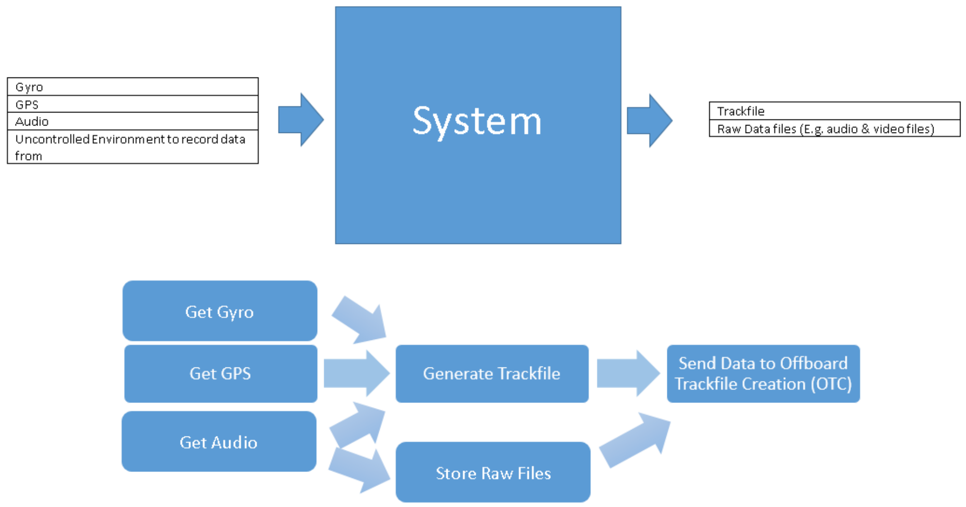

Figure 3: The input output analysis for the system

the workload necessary to implement data fusion alongside deep learning of acoustic classification and detection as shown in Figure 5.

2.7 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 6. 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.

2.8 N^2 Analysis

An N^2 Analysis methodically expands on the what data moves around the system. This is to compliment the discussed complexities in the Sequence Diagram by documenting what data is expected. A example of this is shown in Figure 7

2.9 Spray Diagram

The Spray Diagram shown in Figure 8 shows how details of the system can have multi-factored effects in design requirements, outcomes and operational use. Of interest in the diagram is the relationship between high-quality output will require high-quality input, and this may increase production and design costs.

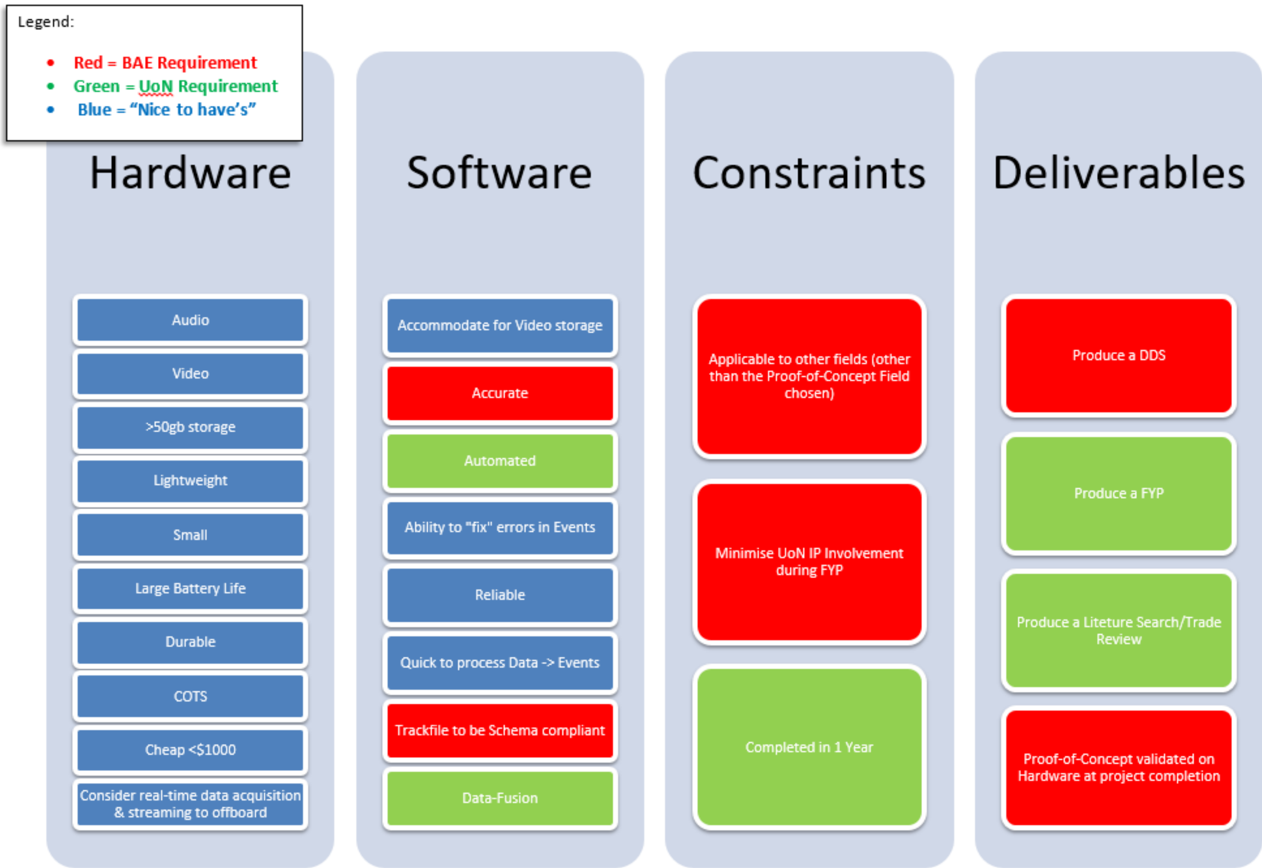

Figure 4: The Affinity Diagrams for the full system architecture

2.10 Matrix Diagram

A matrix diagram was made to review the now more refined, reduced scope as shown in Figure 9. This works by using a "strong", "weak" or "none" indicator for each aspect of the project. It highlighted the difficulty in balancing the needs of both major stakeholders.

2.11 Final Problem Analysis and Discussion

The purpose of the project is to hypothesis and produce a proof-of-concept of how digital technologies could improve debriefing processes and outcomes. This seems achievable within 1 year, and should provide an excellent learning opportunity for the author.

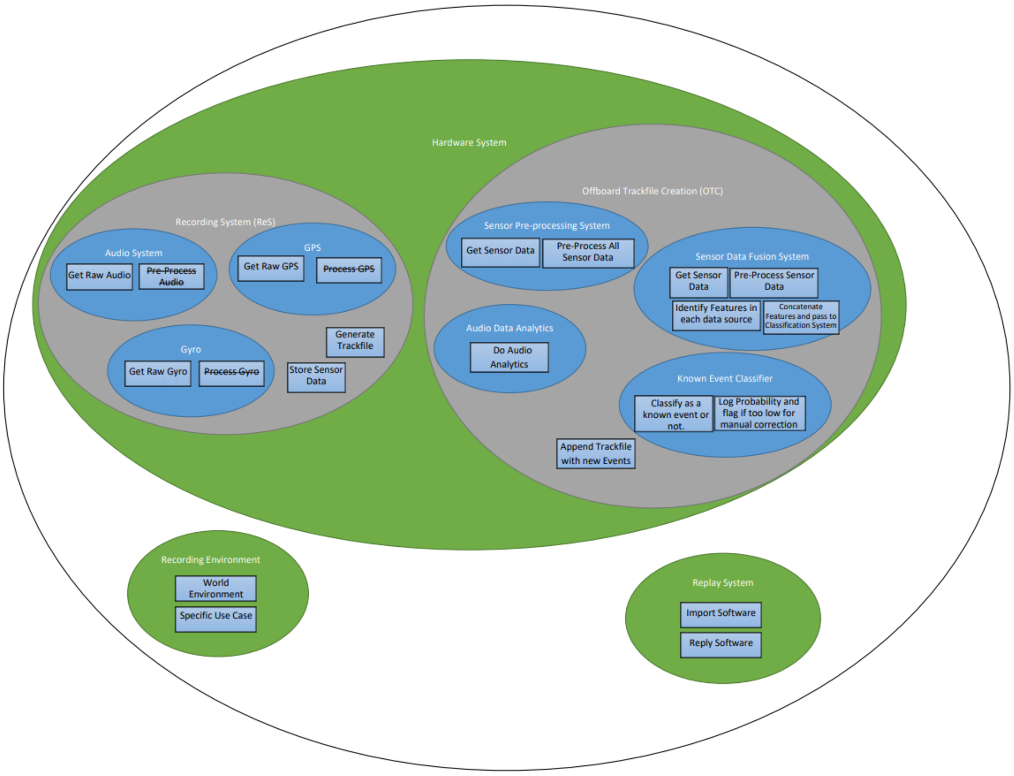

Figure 5: A systems map for the project

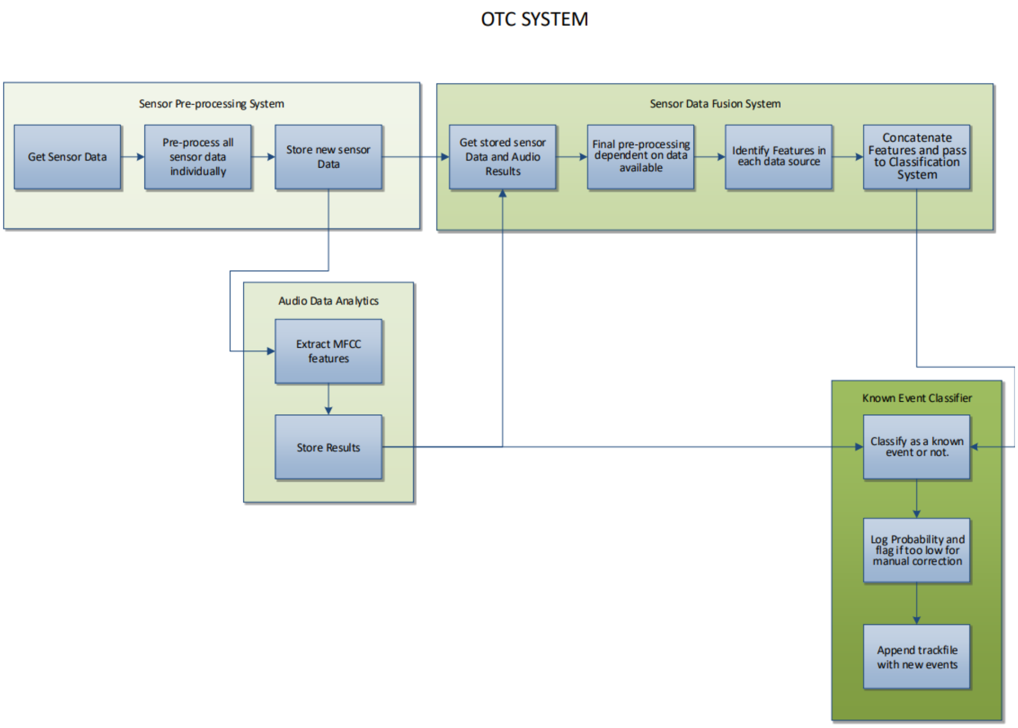

Figure 6: A basic Sequence Diagram for the project

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 7: A N^2 Analysis for the project

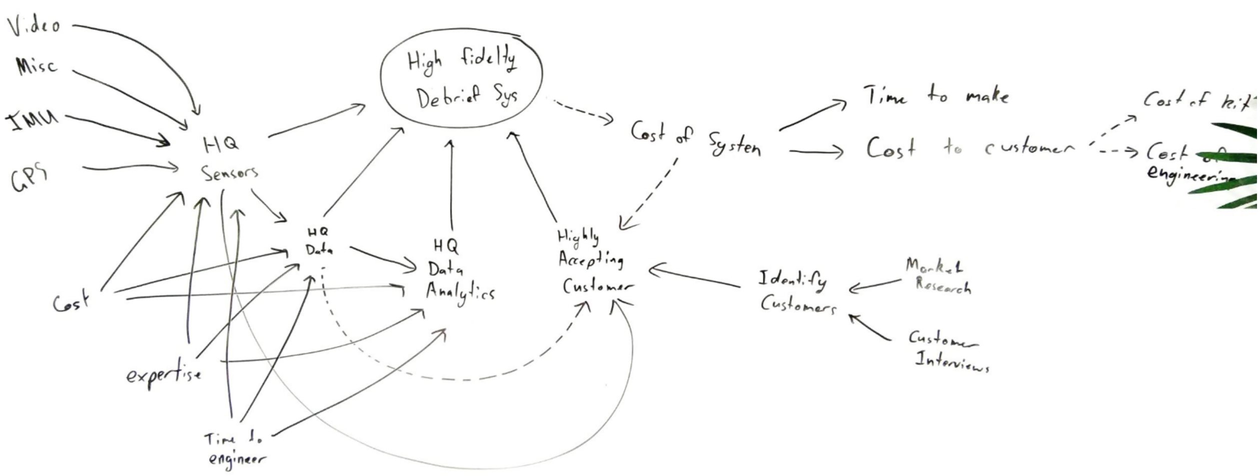

Figure 8: A Spray Diagram for the project

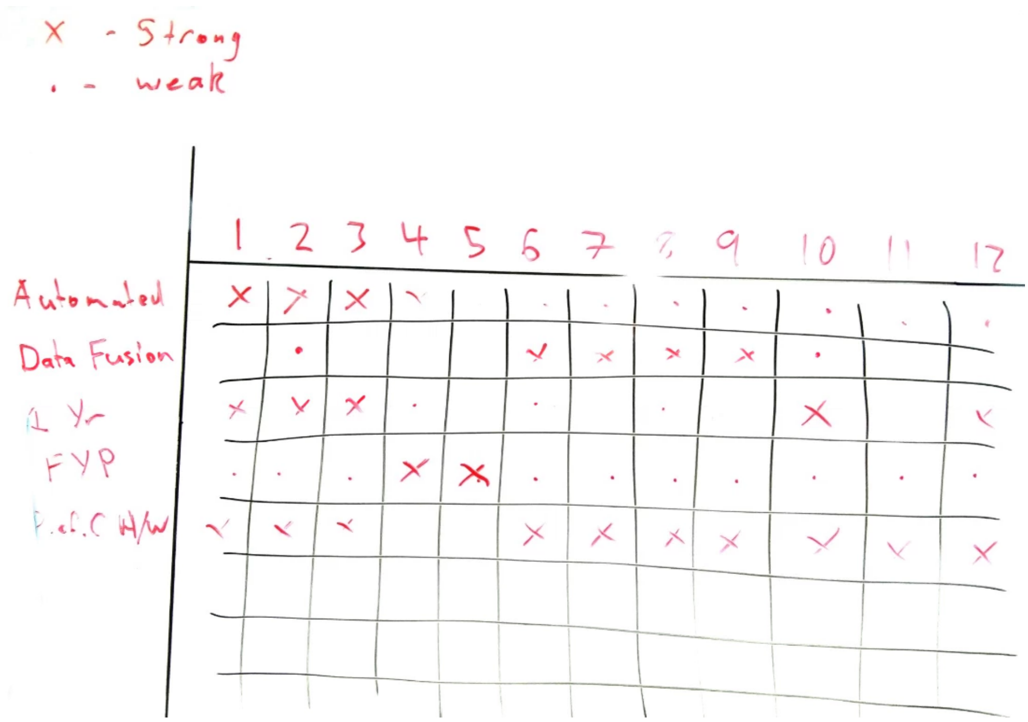

Figure 9: A Matrix Diagram for the project

3 Review Of Literature

A literature review of the current research was conducted with guidance from University of Queensland's guide to Literature reviews of Queensland (2018) . There are 3 nested areas of research that are of interest to this project. The wider classification and regression research topic, the machine learning topic that covers many various fields of implemented and theoretical machine learning applications, and finally the specific field of Acoustic Event Detection and Classification (AED/AEC); this field has traditionally relied on older techniques, and is undergoing a revolution with machine learning.

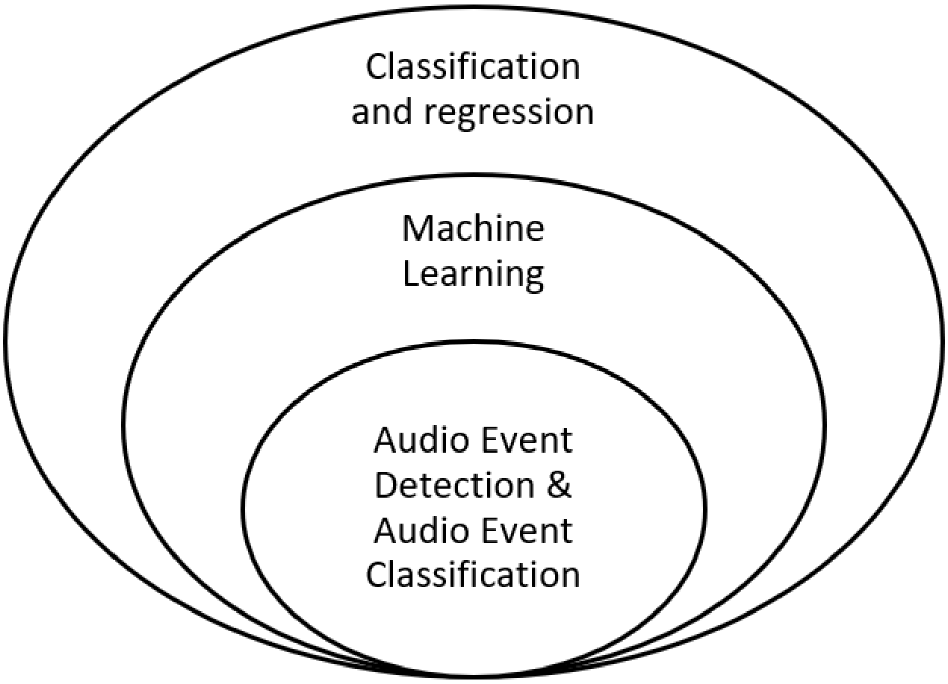

Figure 10: The literature for this project is a niche topic in the machine learning area of research

3.1 machine learning

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. Mitchell (1997)

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

The computer program described in this project has been produced in Python, using the sklearn and TensorFlow libraries. [FLAGGED FOR REVIEW]

3.2 Acoustic Event Detection and acoustic event classification

Historically the technologies for AED/AEC have been Support vector machine (SVMs), Hidden Markov Models (HHMs), and more generally, basic DSP (Digital Signal Processing) classifiers. But in the last 8 years, AED and AEC research has emerged as a hot topic of research. Research focuses on the two key tasks: Detection (when did an acoustic event occur in the audio, and when did it stop?) and classification (what sound occurred?).

New research papers are often the result of DCASE competitions, which is an official IEEE Audio and Acoustic Signal Processing (AASP) competition. Figure 11.

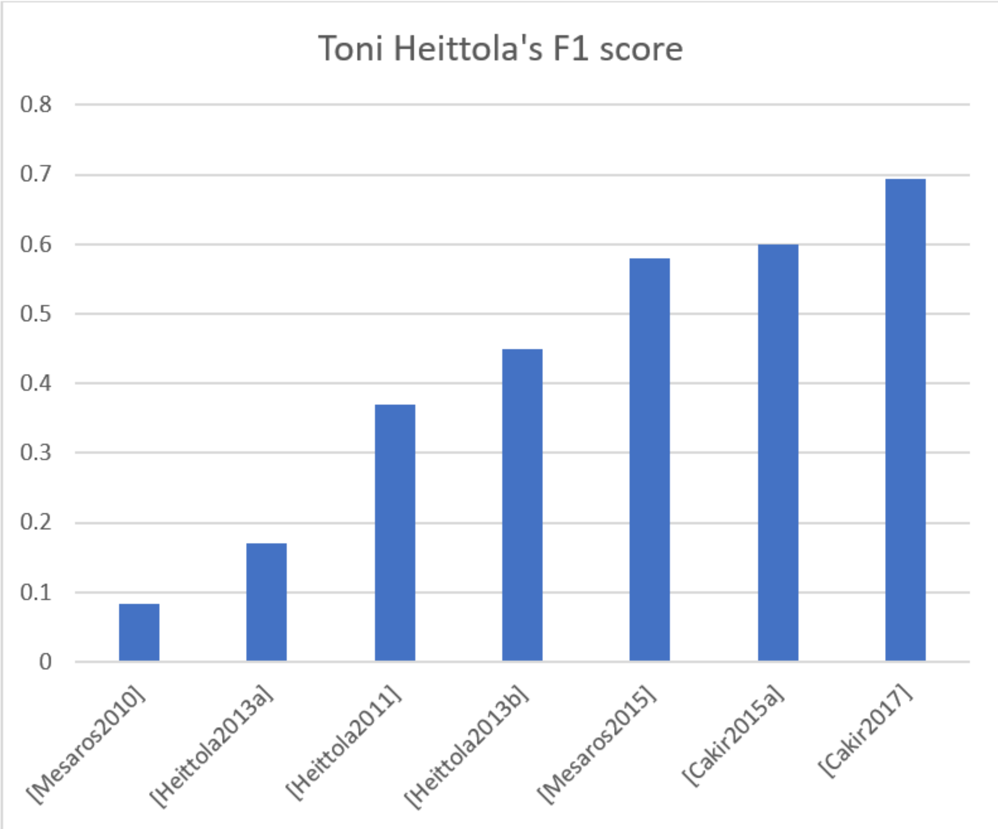

Figure 11: DCASE Researcher Toni Heittola's F1 Results reflect the AED/AEC field of research with drastic improvements over the last 8 years.Heittola (2018)

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

The DCASE competitions format started 2013, had its second competition in 2016, and since then has been a yearly event. Since 2017, there has been a new format where the competition starts march, formal results are released by September, and a workshop for participants on the best team's work in NovemberDCASE. (????). 2016 was the first to feature machine learning (winning teams incorporated machine learning into ensemble models/classifiers). By 2017 all entries utilised machine learning. Recently the results from DCASE2018 have been published; progress in the AED/AEC field has allowed more sophisticated, real-world applications to be evaluated (Challenge Task #4)201 (2018). The complexity of the dataset, the number of acoustic classes and fidelity of output are unprecedented in the AEC field. The winning results at this level of complexity are not as high quality compared to last year's more controlled environment/dataset (2017 had a 41.7% F1 score201 (2017) vs 2018's 32.4%201 (2018)), but pre-sent the best precedent for the problem addressed in this paper.

3.4 Leading Research, DCASE2018 Task 4 Winner

The winning model used in the DCASE2018 Task 4 challenge used a "Mean-teacher" model for classification (useful abstract results averaging applicable to a range of classifier models), a CNN for context gating (a pre-classifier step to improve flaws in training methodologies for some machine learning modelsDauphin et al. (2016)) and a bidirectional recurrent neural network (RNN) to improve the utilisation of unlabelled, unbalanced training datasetsJiaKai (2018); this last component is important for this paper, as the leading dataset outside of DCASE challenges is the weakly labelled Google AudioSet, of which only small percentage of is balanced.

Table 2: AudioSet data on cars

Dataset	Number of videos	Duration hours
Evaluation	280	0.8
Balanced train	296	0.8
Unbalanced train	40,978	113.3
Overall	41,554	114.9

$$CarAudioSet : \frac{0.8}{114.9} = 0.69\%$$

4 Method

4.1 Introduction to the Method

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

Stage 1: Data

Stage 2: Preprocessing

Stage 3: Model

Stage 4: Fitting/Training

Stage 5: Evaluation

[citation needed].

Stage 1 Stage 1 is choosing what raw data is available and what information the model must product. This is the inputs and the outputs that you want from the system, and the expected result.

Stage 2 Stage 2 is preprocessing. This is about taking the data available transforming it. This could be optimising, filtering, feature building (producing refined data of interest to the model e.g. MFFCs from raw Audio) and in 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 picking the model. The method of picking a model may be based on doing research into what is best practice for your particular problem, looking at recent papers and investigating whether the particular topic demands any special requirements. Once a model is picked, the parameters of the model must be chosen (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 to cost Stochastic Gradient Descent (SGD) and back propagationLeCun et al. (2012). In summary, it is the process of changing the weights proportionally to their contribution to the error. Equation 4.1 adapted from LeCun et al. (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.

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

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 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 to name a few. F1 score will be used as the primary measure because the F1 has been is the research industry standard for evaluating algorithms.

It is beneficial to work up slowly to this goal. 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 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 in
- Build the machine learning classifier as described above, and test its effectiveness
- Validate the whole system by using the classifier in the pipeline (by take the classified output, and appending it to the trackfile)

4.2 Required Data

The machine learning uses GPS, accelerometer, gyroscope, and audio. A Google Pixel 2 was selected as the model data acquisition, which table /ref below shows.

Table 3: Data Available Havard (2016)

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 choice of what data to collected and use was found through the first problem analysis step. As the goal is to maximise the accuracy of the classifier (whilst still being an achievable goal), 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 smartphone as shown in section 2 and appendix E.

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 4, of which GPS and acceleration will be used. It was also chosen 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 development took 5 weeks to get to a useful state. It can record Audio to a .wav file and record raw GPS lat/long/alt, raw accelerometer, and raw gyroscope to a csv. There are technical issues still present in the app (notably that the screen must be unlocked to record).

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 helped with 30% of the overall programming, and code reviewed the product along with other colleagues. Credit goes to [Sebastian Wallman].

4.3.1 Data collected during project vs ExtraSensory dataset

The table ?? shows a snippet of the Comma-Separated Values (csv) data produced from the app. The id is a unique count, The table ?? shows a snippet of the csv data from the ExtraSensory dataset.

Both datasets share the same timestamp format, so a "Time" column has been added to the table for illustration purposes only.

The ExtraSensory dataset is compiled from 60 users with a diverse range of phones ("34 iPhone users, 26 Android users."Vaizman et al. (2017). This is in contrast to this project's phone range, which was limited to 2

Listing 1: Code snippet for getting Accelerometer data from the Android System as above Google (2019) above

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

    ...
}
```

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

Android devices, a Samsung S6 and a Lenovo Zuk Z2. The Samsung uses a InvenSense MPU-6500, the Lenovo Zuk Z2 unknown.

> “ Devices: The users in ExtraSensory had a variety of phone devices. iPhone generations: 4, 4S, 5, 5S, 5C, 6 and 6S. iPhone operating system versions ranging from iOS-7 to iOS-9. Android devices: Samsung, Nexus, HTC, moto G, LG, Motorola, One Plus One, Sony. ”

4.4 Data Pre-Processing

Python was used to process the data before the machine learning modules of code, and Matlab was used with GPS, IMU to prototype the algorithms.

~~The data from the ExtraSensory dataset is provided in its raw format, which is best option is for the data to be raw, as this allows platform independent filtering, smoothing and sensor fusion[citation needed]~~.

Audio Data preprocessing of the Audio was done by Mel Frequency Cepstral Coefficients (MFCCs). Start with a recorded .wav sample, and split the whole sample into smaller sections via the sliding window method[citation needed]. Take each window and calculate the Fourier transform. This in turn is both “Feature Engineering”, as well as preprocessing to be training compatible.

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

Audio to a .wav file and record raw GPS lat/long/alt, raw accelerometer, and raw gyroscope to a csv.

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

Listing 2: Matlab trial of MFCC extraction

```
speech=readwav(file_path,'s',-1);
%rng('default');
%speech=speech+randn(size(speech))*eps;                          %dithering
%-------------------------- PRE-EMPHASIS ----------------------------------
speech = filter( [1 -0.97], 1, speech);
%---------------------------FRAMING & WINDOWING----------------------------
frame_length_inSample=(Fs/1000)*Window_Length;
framedspeech=buffer(speech,frame_length_inSample,frame_length_inSample/2,'nodelay')';
w=hamming(frame_length_inSample);
y_framed=framedspeech.*repmat(w',size(framedspeech,1),1);
%--------------------------------------------------------------------------
f=(Fs/2)*linspace(0,1,NFFT/2+1);
fmel=2595*log10(1+f./700); % CONVERTING TO MEL SCALE
fmelmax=max(fmel);
fmelmin=min(fmel);
filbandwidthsmel=linspace(fmelmin,fmelmax,No_Filter+2);
filbandwidthsf=700*(10.^(filbandwidthsmel/2595)-1);
fr_all=(abs(fft(y_framed',NFFT))).^2;
fa_all=fr_all(1:(NFFT/2)+1,:)';
filterbank=zeros((NFFT/2)+1,No_Filter);
for i=1:No_Filter
    filterbank(:,i)=trimf(f,[filbandwidthsf(i),filbandwidthsf(i+1),...
        filbandwidthsf(i+2)]);
end
filbanksum=fa_all*filterbank(1:end,:);
%-------------------------Calculate Static Cepstral------------------------
t=dct(log10(filbanksum'+eps));
t=(t(1:No_Filter,:));
stat=t';
delta=deltas(stat',3)';
double_delta=deltas(delta',3)';
%--------------------------------------------------------------------------
```

“First, resample the audio clips at 22,050 Hz, because the high frequency part of sound signal is not useful for event detection in daily life

Extract the log mel-spectrogram from the audio clips by 128-bin, 2048-window and 365-hop (1683- overlap)” Aka classic sliding window, but this specifies some good over-lap, data density etc.

GPS ~~Talk about what a GPS is, and how we get the signal~~ The GPS signal is requested by the ”Fused Location Provider”, in the Google Play services location API and comes heavily prefiltered[citation needed]. 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 applied a Low Pass Filter[citation needed]. A Kalman Filter was considered but wasn't used to avoid modelling the specific android sensor plant dynamics.

Figure 12

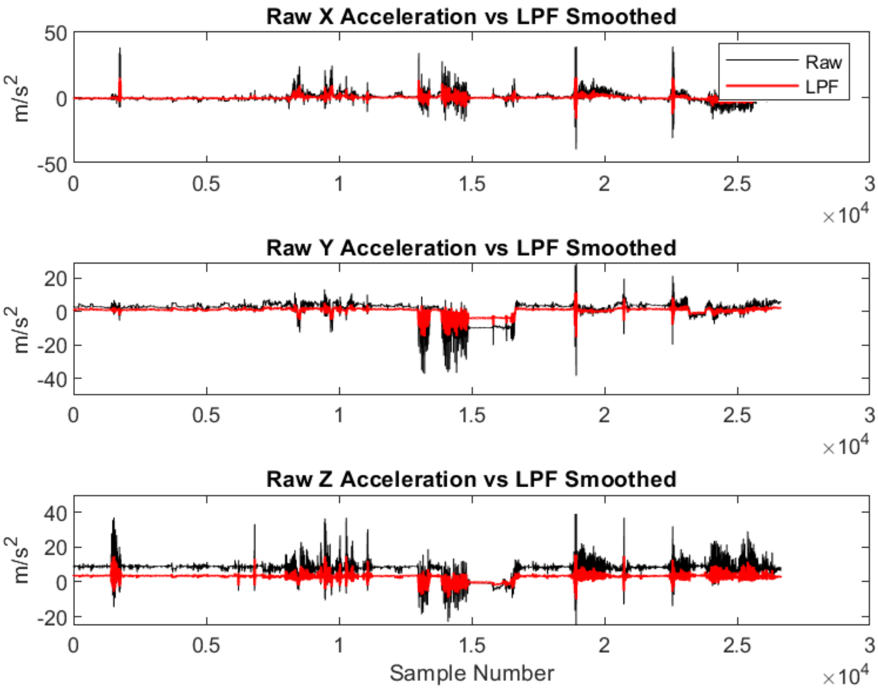

Figure 12: Results of Low Pass Filter vs Raw accelerometer data

4.4.1 Audio

Data Training, Testing and Evaluation Set The ExtraSensory dataset contains over 300,000 samples across 60 Users (in a CSV format with 278 feature columns), as shown in ??. Each row contains an a 'snapshot' of the sensors at a frequency of once a minute. There are large non-continuities across each user. The dataset is unbalanced, and has had its labels cleaned by the researchers involved. Further details about the users themselves can be found in 5, or at it paper Vaizman et al. (2017).

The dataset 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 is to isolate recording hardware, and other plant noise. However this is undesirable, as this can 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, to minimise its impact in evaluation. Thus is the benefit of training on a per sample basis. By introducing other user noise into a training set, we minimise overfitting "for free". Finally as a "kfold" (or other Cross-Validate methods) is applied to the whole training dataset for training + validation purposes, the likelihood of the random selection incorporating a larger variance of noise increases. As a result, the data is used on as a per sample basis.

"kfold" Cross validation [FLAGGED FOR REVIEW]kfold Cross Validation can be described as:

> " ... approach 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. "

James et al. (2017) 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.

The question of whether supplementary data will improve the classifier's F1 score is best answered by deciding on

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')
```

an appropriate classifier, and testing it with and without the supplementary data. As discussed, the appropriate classifier could be chosen via hyperparameter testing against a validation set, however the feature analysis carried out above will also influence classifier design. The unbalanced nature of the dataset is also a factor.

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 9.

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. This could be result in "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 a ensemble of binary classifiers (combing together the output of the same dataset that has been parsed into a "Ford" classifier, a "Holden" classifier etc.), or more recently through a "One Hot"Huffman (1954) encoded output array as seen in table

Table 4: Data Available[citation needed]

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

6. A key element of multi-class classifiers are that the classes are mutually exclusive. The car brands in 6 are mutually exclusive.

However, there is another type of classifier called the multi-label classifier. An example of a multi label classifier is this project. The project may want to label whether signal in the provided data represents jogging, walking, talking, eating etc. For these labels, they are not mutually exclusive, and could all be occurring at the same time.

For this project, a binary classifier and a multi-label classifier will be used and evaluated. The reason for this is because the intent of the project is to answer the question does supplementary data assist in classifying an Acoustic Event Detection and Classification (AEC/D) problem, and a binary classifier is a valid and succinct way to answer this question. However, for most realistic use cases, what is important is whether the result is valid in a multi-label classifier. For this reason, the methodology section will be applied to creating a multi-label variant of the network. This will use the same hyperparameter optimisation as the binary classifier and will be discussed in the results.

Table 5: ExtraSensory Data Breakdown Vaizman et al. (2017)

	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

Table 6: Example Hot One Encoding

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

4.5.1 Model Selection

Model selection is an important aspect of machine learning, because of what is known as a the "No Free Lunch" theorem.

> "No Free Lunch theorems have shown that learning algorithms cannot be universally good. Magdon-Ismail (2000)"

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 datapoints of data in use, the type and format of the desired information, as well as some unique factors: desire to run onboard, desire to run generalise well (even into different use cases). The standard approach is to either a) try many options or b) have an educated guess as to the

Convolution Neural Networks CNN convolution neural networks are primarily designed and received the best results for image-based classification problems this is because the architecture has been designed with computer vision in mind for our project

Recursive Neural Networks The recursive neural network (RNN) is not well suited to tabular data this is because and this will not be used because it has been designed for producing multi-class probabilities we are we could use a recursive neural network to generate a probability output that is to say use it as a regression out progressive algorithm to output all a probabilities however we are after a binary classification and fast it is not relevant more important what the classifier deems as an appropriate probability before it classifies for future work this may in fact be a very useful extension as the tailoring of the sensitivity and specificity probability thresholds of the recursive neural network. This could also be used in conjunction with a soft Max layer[FLAGGED FOR REVIEW] to produce a normalised a labelling classifier this However it will not be used due to the extra complexity which is not needed by the use case. Other options include a random Forest classifier which have been used to great success by Phan, Phan et al. (2015). However a random Forest classifier is not well suited to unbalanced data sets and would prevent a much bigger problem with for training the future the random forest classifier has been evaluated for comparison in the results section however as a architectural decision it will not be used simile so scikit-learn also has the algorithm extra trees classifier citation needed which can be

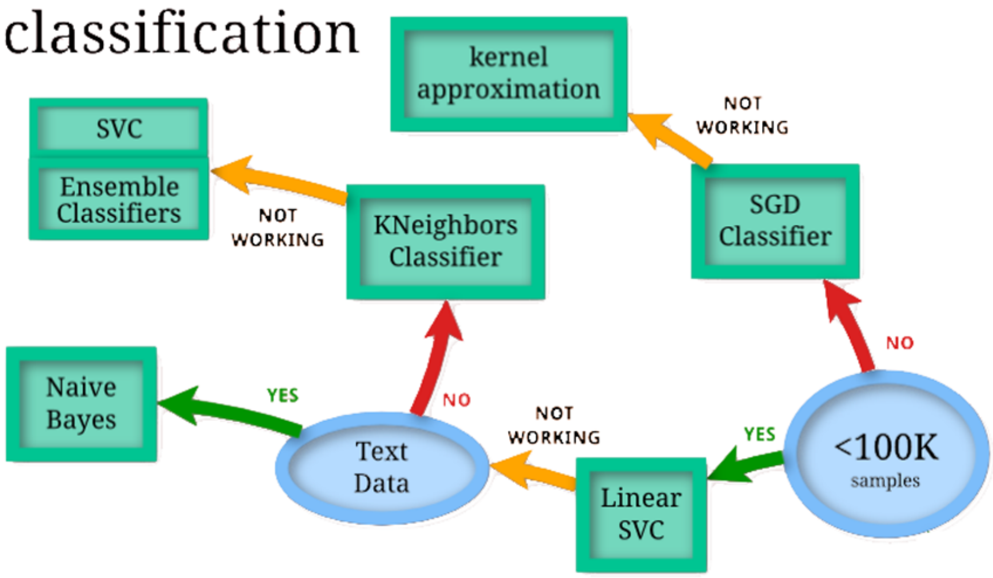

Figure 13: Scikit-learn's suggested model selection process

used in a similar way to the random Forest Hill that you eat better handles label caissons which is important for our use case ll use case is evaluating a binary class against the potential list [FLAGGED FOR REVIEW]

Mulilayer Perceptron Feed Forward Neural Network The multilayer perceptron classifier has been selected this

For our project we are using tabular time-invariant data using a snapshot data in a tabular format means that we will not see the benefits of a convolution neural network in a deep learning environment vs the more simple mlpclassifier which trains faster will receive the better result and is expected to compress better win-Network(parametee will also

Final Model [FLAGGED FOR REVIEW]fix the details The MLPClassifier is a feedforward neural network, trained via backpropagation. It was set up using the "lbfgs" solver, an alpha equal to $1e-5$, and was hand selected to have 80 by 10 hidden layers (80 long, 10 deep). This was the classifier to use for evaluation.

4.6 Model Validation

Validation is the process of optimisation the hyperparameters of a model on the hold-out validation set of data. This is an important aspect of 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 must be optimised before training begins. This is an extension of trying to find the optimal model for the solution.

Model Performance Metrics Before the hyperparameters can be optimised, a metrics to optimise the network on must be selected, of which there are many ways to do it which will now discussed.

4.7 Terminology

Before evaluating metrics, the Terminology of the following formula must be formalised. The first is what a "correct" classification is. The following terminology will be used:

- 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 formula

Accuracy Accuracy is the measure of how many samples the classifier correctly classified across its operation, as shown in 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 you may have specific weightings on False Positive and False Negatives rates could be the medical industry, where a false positive will have a cost associated with it (the cost of further tests, or screenings) vs a false negative (where by a patient's health outcomes could be severely impacted by the incorrect diagnosis).

In this project, accuracy would be a very poor metric for machine learning because the ExtraSensory dataset compromises 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).

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 14. 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.

In this project, the Confusion Matrix gets generated to sanity 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 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 is 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.

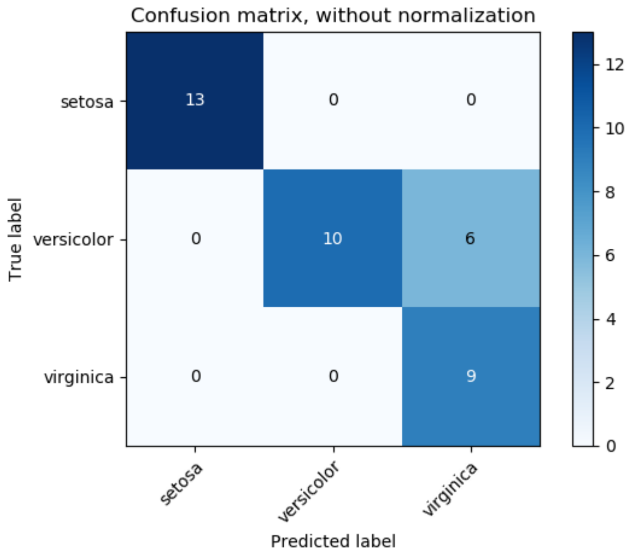

Figure 14: A confusion matrix of the Iris Database.Pedregosa et al. (2019a)

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 is 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 will ideally 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 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 will use 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 is required to apply to a multi class or multilabel problem. This is often done through a sum of F1 Scores across each class or label. For best precision, a weighted could be applied to sum 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 class-classification 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 you change the classification decision threshold. 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).

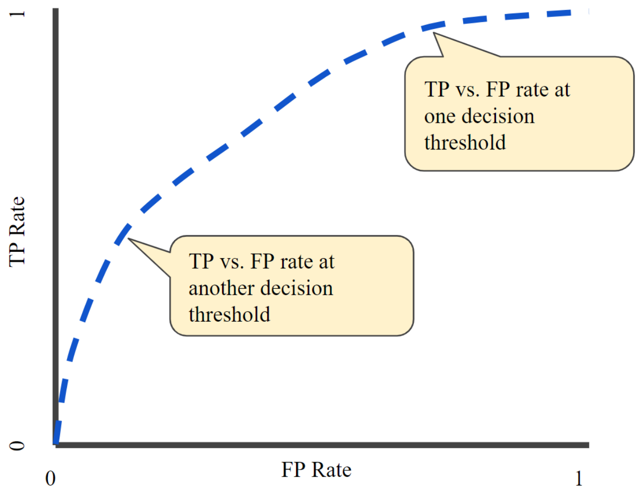

Figure 15: A ROC Curve example from Google's machine learning shortcourseGoogle Brain (2019a)

In this project, the AUROC will not be 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).

4.7.1 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 backpropagation, as it is external to the classifier. These are parameters such as hidden layer size, training rates, classifier specific strategy, etc. The way to choose these hyperparameters is therefore an optimisation problem. The problem is to maximise the score selected 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 from default hyperparameter settings on various algorithms.

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

```
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 16 shows a grid is generated based on the hyperparameters intervals selected. A power element of Grid Search is the capacity to select non-equidistant intervals.

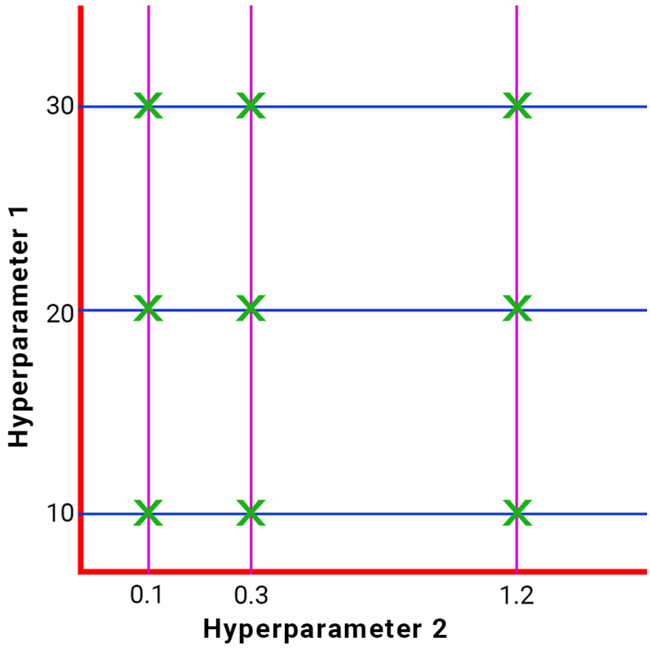

Figure 16: 3 options for Hyperparameter 1 and 2 have been selected. Each combination will be evaluated, as denoted by the green 'x'.

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[citation needed]. 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 "show[n that] empirically and theoretically, randomly chosen trials are more efficient for hyper-parameter optimization than trials on a grid"Bergstra and Bengio (2012). In part, this is due to a principle known as "Embarrassingly parallel".
"Embarrassingly parallel"Maurice Herlihy (2012) is a critical concept in machine learning that allows the Random and Grid searches to be run in parallel. 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, 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. Brochu et al. (2010)"

Bayesian Optimisation for hyperparameter selection was not tested, as it is suggested it is best suited for optimising more complex networks. Snoek et al. (2012) Brochu et al. (2010) 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 7 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.

Table 7: 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"

These parameters are then placed into a python Dict, and passed into the scikit RandomizedSearchCV function.

[FLAGGED FOR REVIEW]does it need a title/numbers etc.?

and the results of which were

Details of a Grid Search can be found in Appendix F.

4.8 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) Pedregosa et al. (2019b) Google Brain (2019c)

```
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_)
```

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

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 applyGeron (2017). "

TensorFlow and Keras were selected to initially design the classifier due its ability to run on the TensorFlow Lite. This is inline with the motivation of the project to allow widespread use of an AED/C solution. The code snippet 4 shows how a model converts.

Listing 4: Code snippet for converting the classifier model to TensorFlow Lite for mobile operation Google Brain (2019b)

```
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)
```

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. A code snippet for that design is shown in 5.

Due to the sample size of the dataset (AudioSet's large database, but smaller balanced datasets for the specific use-cases identified), and the use of GPS, Acceleration and Gyroscope data, the classifier could have been prone to overfitting. [citation needed]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 (MFCCs across N-bins), and the classifier could have a heavy bias towards it and as a result overfit. 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 17 shows the scikit Random Trees feature importance of the set (Audio, Accelerometer, Gyroscope, GPS).

A Random Tree Classifier was used to evaluate feature importance. Random Tree Classifier's perform poorly on unbalanced datasets. like on 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)

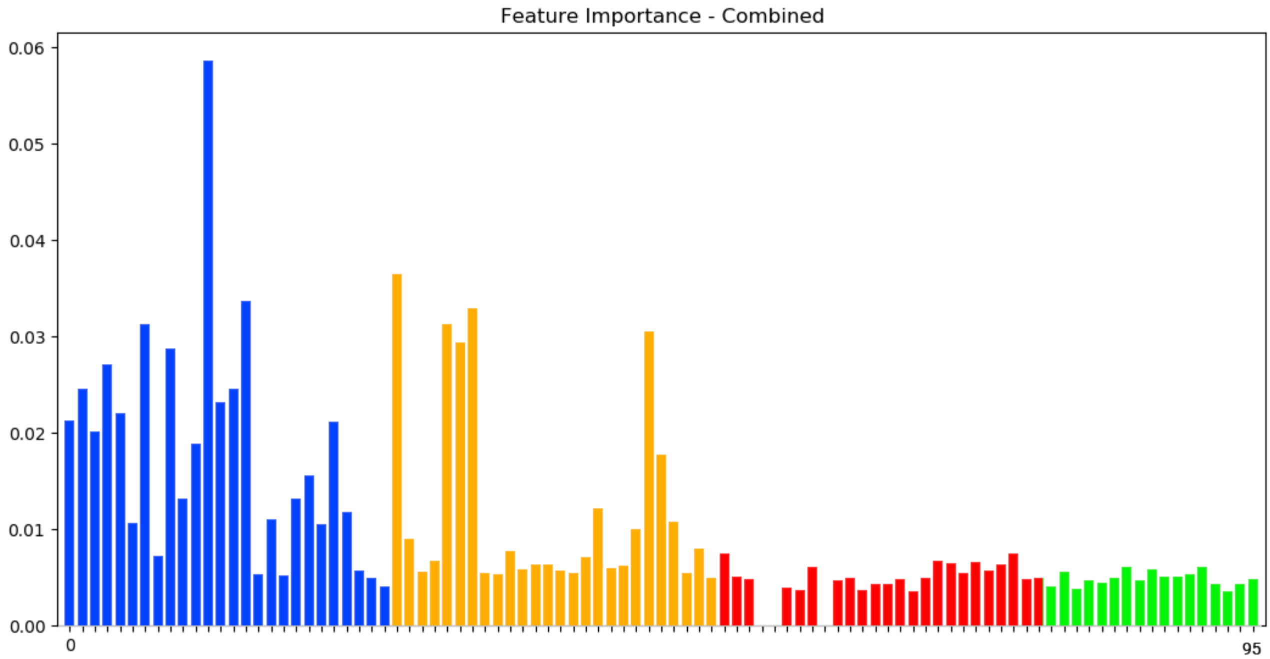

Figure 17: A Random Tree Classifier was used to evaluate feature importance on a the ExtraSensory after random undersampled dataset, during the design phase of the main classifier. Features were Audio (Blue, 26 bins), Accelerometer (Orange, 26 features), Gyroscope (Red, 26 features), GPS (Green, 17 features)

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 and in 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 have been utilised in contrast with writing a framework from scratch.

4.9 Method Summary

The method details 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 is a label (or labels) detailing the Acoustic Event in the sample (e.g. The labels found in G like 'Walking').

Listing 5: Code snippet for the early CNN model

```
model = Sequential()
model.add(Conv2D(32, (3, 3), padding='same', input_shape=x_train.shape[1:]))
model.add(Activation('relu'))
model.add(Conv2D(32, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Conv2D(64, (3, 3), padding='same'))
model.add(Activation('relu'))
model.add(Conv2D(64, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Flatten())
model.add(Dense(512))
model.add(Activation('relu'))
model.add(Dropout(0.5))
model.add(Dense(num_classes))
model.add(Activation('softmax'))
```

2. The required data is the ExtraSensory dataset, or a custom dataset as recorded by the project's App.
3. The score selected is the F1 score. This is due to it being an industry standard.
4. The classifier chosen is the MLPClassifier which is a feedforward neural network, trained via backpropagation.
5. Hyperparameters of the MLPClassifier are chosen via a Random Search Cross Validation.
6. Training and Validation was run with Early Stopping using a 10% validation set, using 59 out of the 60 ExtraSensory users.
7. Final testing is done with the 60^{th} user.

5 Results

The desired results are a demonstratable difference (or lack thereof) in the F1 score, after comparing a classifier with and without supplementary data. This will answer the question, and title of the report, "[What is] the effect of supplementary data on acoustic event classification through machine learning". To this effect, 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
```

It was trained to classify label 2. Average training time was 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.

```
--- 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 Only

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

Results from Audio, Accelerometer, Gyroscope, GPS

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

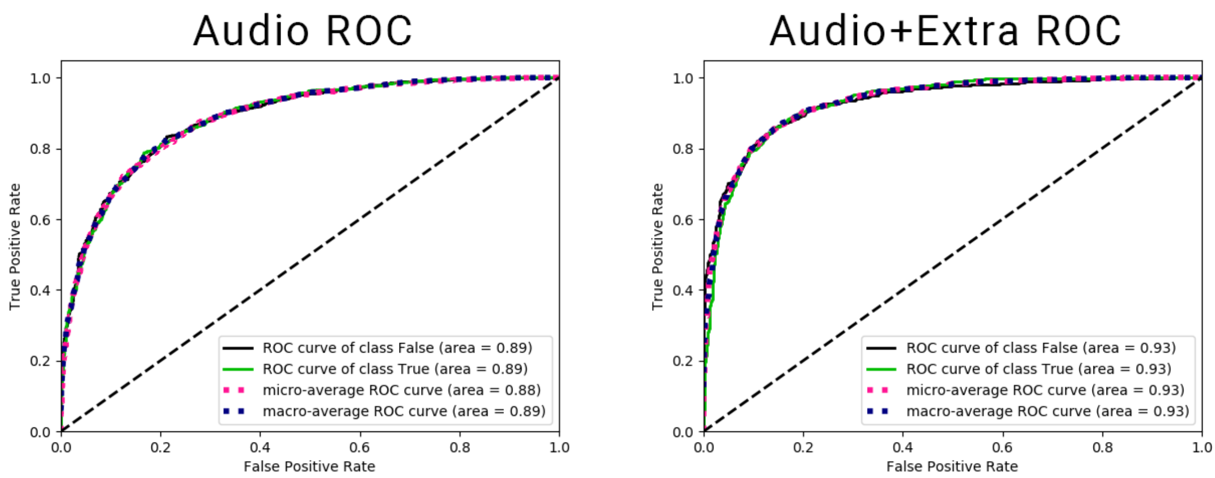

Figure 18: ROCs for Audio, Audio+Extra

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,
```

It was trained to classify labels 2 through 22 inclusive. Average training time was 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.

```
--- 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
```

```
--- 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
```

6 Summary of Results

The results achieved (shown in 8) 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 8: 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%

7 Discussion

The results above aim to demonstrate a quantitative difference in F1 score across 2 the classifier 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.

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 infers that the results of this project will affect all models, but the quantitative impact may vary post-training. Similarly so, there is no assumption on the strength of this result on application to other projects; the dataset and training may significantly affect the the strength of this result.

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 18 show the consistency of the model. This is demonstrated via the smooth continuity of the ROC curve (no major jumps in TP performance). This sanity check ensures that the results produced were not the result of a classifier optimised about artefact of the post-training model.

Changing Hyperparameters The results make the assumption 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 Tseng et al. (????) Virtanen (2010) 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 Virtanen (2010)). 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 Tseng et al. (????), which could extend to generating/classifying new labels on the ExtraSensory dataset. This would allow a more complex dataset, such as AudioSet Google (2018) to reprocess the ExtraSensory database to have a larger number of classes.

Causation The increase in F1 score could occur in 2 areas. The first is during hyperparameter optimisation, in which the hyperparameters are fitted to our 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.

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 an Acoustic Event as a binary output, rather than identifying when the Acoustic Event occurred in the sound sample. Phaun discusses the differences and complexities involved in AEC vs AED in a paper called "What Makes Audio Event Detection Harder than Classification?"Phan et al. (????). This suggests that this project's result may be valid Acoustic Event Detection, given AEC is a crucial element of the detection pipeline.

Significance The results from this project aim to present a peer-reviewable benchmark for the impact of the supplementary datasources of an accelerometer, gyroscope and GPS. The key significance is that the qualitative result can be taken and applied to similar projects.
Of note, no effort was made to discern the impact each of the 3 additional datasources played in the 10.5%

and 11.9% improvements obtain in the results. This was a conscious effort, as these datasources are prevalent, and commonly found together. It could be inferred from Appendix D that accelerometer was likely the most impactful element. It would be left to future work to investigate and discern the specific impact each sensor had, and the cost of including "less important" features vs training time.

Quality of Data ExtraSensory was used as the dataset for this project. As a result of using the a public dataset, there was no investigation into the quality of the data within the dataset, beyond reading the original study. The quality of the recording device, any preproccessing, nor whether they generalised well has been checked. This is a key limitation with analysing the quantitative result. This limitation does not invalid the result however 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 Vaizman et al. (2017). Whilst this introducing human error into the dataset, all datasets will required a decision boundry that may be controversial. The ensure the quality of the decision boundry, one would take guidance from their use case in real world use, or would take guidance from similar studies for their decision boundry.

Choice in Frameworks The choice in machine learning frameworks (and their implimentation) could effect the result. An assumption is made that this will be a minimal impact, and the results would stand valid across other frameworks.

Details of the Audio Data A key assumption was the audio data provided (mono-channel recorded from a mobile phone) is representative of all audio data for use in machine learning. Furthermore, the audio was processed into MFFCs, which are a leading way to process audio data. The 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 differ the decision boundry which supplementary data would assist.

Types of Supplementary Data The report took the assumption that "supplementary data" in the AEC/D field would be defined as accelerometer, gyroscope, and GPS data. A limitation of this result is whether the result would transfer and compare with different supplementary data sources; video, metadata, multiple datasources for one sample etc.

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

F1 Score as a Metric The project sought to measure a decernable 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 realtive increase in performance would be vastly different. A limitation of this result is that the validity of the result working in another project assumes that an improvement in F1 score throughout hyperparameter optimisation and backpropagation/training of the model is desireable in the other project.

Limiting Model Choice to Avoid Excessive Feature Some machine learning models are senstive to feature [citation needed][FLAGGED FOR REVIEW]

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. Due to the natural of the supplementary data being time varying measurements, it would be expected that their inclusion to a AED/C pipeline would benefit, however it would be left for future work to determine the exact nature of this benefit.

evaluate the generalisation of this methodology to new data sets what it does answer is the original question of does including supplementary data increase the F1 score and the answer is yes

what is important is to discuss how you would evaluate the generalisation of this model what impact the model quality has on this result if we've used a different model which we have and we will show this result what is the difference in the improvement is there any improvement how do you evaluate whether a model will benefit from this extra data points there are some models specifically which you do need to do feature reduction and feature analysis on to ensure you don't over specify the features even if you low weight the features you may still have an issue of having too much data

performance is another element of this is. For the results what was the cost so we have increase the amount of data we need to use increase the data we need to process increase the data we need to collect to train we are also looking at a larger model file size which for a use case of potentially online processing would be very detrimental

The 8% results are based on solid evidence they may not actually be a valid application or a valid progress for achieving a 5% increase for instance how do we only change the data set and retrain revalidated and then re-evaluated this process would we see similar numbers that is the model likely wouldn't produce the exact same if one school so is our 5% related or proportional to the original benefit had we not done hyperparameter searching to optimise our model potentially we would have seen a much larger game in F1 score from the addition of the additional data points that hypothetical would be because if the model is already underperforming / are an optimised then the additional data points May introduce that additional optimisation needed

8 Conclusion

This is one of the most important parts of the report. In the conclusion section, you should

- briefly summarise the results,
- reflect on the work presented,
- make recommendations,
- suggest future work or improvements.

The report demonstrated that introducing Accelerometer, Gyroscope, and GPS data contributed to an 8% increase to the F1 score in both the binary, and the multilabel MLPClassifier. This a relative increase of 10.5% and 11.9% respectively, compared to without the extra data. The results had a significant limitations in what was tested, and assumptions about typical AEC/D problems. These limitations do not affect the significance and qualative result of this report, but will affect the quantifiable improvement expected when implimenting these findings in other projects. The result was discussed and found to be valid, and a contribution to the field by specifying an estimate improvement for including of the extra data. To the best of the Author's knowledge, this has not been specifically done before.

The report focused on investigating and analysing the AEC/D problem. A heavy focus was placed on creating a reviewable machine learning implimentation, to transfer results[FLAGGED FOR REVIEW].

The work presented has demonstrated that the fear of machine learning being a "blackbox" method is unfounded. The methodical process applied in this project shows that the result can be isolated, and whilst the performance can not be attributed to hyperparameter optimisation or backpropagation analytically, it is evident that the effect is present in these.

The report looks at how the AEC/D pipeline would be created (such as the creation of the app, preprocessing, etc.), much of which is not reflected in the final results but were critical in the overal Final Year Project.[FLAGGED FOR REVIEW]The results achieved showed audio as a sole input for determining physical actions to be a viable, but the extra available data did improve the classification. The report also reflects on how much room for improvement there is[FLAGGED FOR REVIEW].

Preprocessing took some domain knowledge learnt through out Mechatronics, and processed (mel power coefficients). Results only demonstrate the sheer inclusion of the data, as no processing or optimisation on the extra data was carried out.

Classification was through machine learning (RNN, known topologies etc.). 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. The niche explored in this FYP, of amalgamating traditional audio processing with physical sensor input looks to be of limited use, to limited applications. More over, without large datasets of audio paired with physical sensor data, the machine learning tools explored in this FYP will not be able to be used. As such, the rise of audio recording equipped equipment should be the basis for further research into this space. Future work from here would be to look at the larger role of data acquisition. This is both because it was one of the larger hurdles I had to face, plus it enables further research into the effectiveness of what a diverse dataset can bring. Industry will reasonably expect to take aspects of data fusion and machine learning. To know what

Conclusion The question of "[what is] the effect of supplementary data on acoustic event classification through machine learning" has been answer: The answer is a 8% increase in F1 score, or 10.5/11.9% relative increase in their respective models.

Which is relevant for this approach for this data set because this data set does closely match our target use case this is actually a valid result and this is a worthy increase to the field what this does raise however is how does hell result generalise across other applications because this project had a specific use case it was not within the scope to evaluate whether or not supplementary data would improve audio event classification and detection in other audio sets similarly so the number of audio+Extra datasets is available is limited and the format of that data all the data selection would play a role so whilst we discussed supplementary data to specific data that we use was axela on the data a gyroscope data GPS however it did not attempt to evaluate other supplementary data.

In conclusion this seems to be a valid strategy and the next next step you would take is to look at trying a larger data set when available and 2 validate this on hardware in a actual environment further so there were other methods to approach the same problem.

9 Future Work

Discern what impact each additional datasource had on the result

Use another AEC/D to label this dataset

Novel use of Bogosort as Optimisation strategy A suggested option for Future Work could include adapting the Bogosort to machine learning, as it is an increasingly popular area of research Gruber et al. (2007) Holzer and Maurer (2018) Sherman (2013). The suggested method for using the Bogo search would be 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.

10 References and Citations

References

, 2012. Psychological Debriefing. Cambridge University Press.
URL https://www.ebook.de/de/product/3766790/psychological_debriefing.html

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

, 2018. (n. d.).

Bergstra, J., Bengio, Y., 2012. Random search for hyper-parameter optimization. Journal of Machine Learning Research 13 (Feb), 281–305.

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

Dauphin, Y. N., Fan, A., Auli, M., Grangier, D., 2016. Language modeling with gated convolutional networks. CoRR.

DCASE., ???? (2018). Schedule.
URL http://dcase.community/

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

Geron, A., 2017. Hands on Machine Learning with Scikit, Keras & Tensorflow. O'Reilly.

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

Google, 2019. Sensors motion.

Google Brain, 2019a. Classification: Roc curve and auc — machine learning crash course — google developers.
URL https://www.tensorflow.org/install/pip

Google Brain, 2019b. Get started with tensorflow lite.
URL https://www.tensorflow.org/install/pip

Google Brain, 2019c. Install tensorflow with pip — tensorflow.
URL https://www.tensorflow.org/lite/guide/get_started#2_convert_the_model_format

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

Havard, S., Oct. 2016. Google pixel xl teardown. Https://www.electronicproducts.com/Google_Pixel-whatsinside_text-205.aspx.

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

Heittola, T., 2018. Research: Sound event detection.

Holzer, M., Maurer, J.-T., 2018. Selection via the bogo-method - more on the analysis of perversely awful randomized algorithms.

Huffman, D. A., Mar. 1954. The synthesis of sequential switching circuits. J. Franklin Institute, vol. 257, no 257 (3), 161–190.

James, G., Witten, D., Hastie, T., Tibshirani, R., 2017. An Introduction to Statistical Learning. Springer-Verlag GmbH.
URL https://www.ebook.de/de/product/20292548/gareth_james_daniela_witten_trevor_hastie_robert_tibshirani_an_introduction_to_statistical_learning.html

JiaKai, L., 2018. Mean teacher convolution system for dcase 2018 task.

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

Kaliner, J., 2013. 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. thesis, NAVAL POSTGRADUATE SCHOOL MONTEREY, CALIFORNIA.

LeCun, Y. A., Bottou, L., Orr, G. B., Muller, K.-R., 2012. Efficient backprop. In: Neural networks: Tricks of the trade. Springer, pp. 9–48.

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

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

Magdon-Ismail, M., 2000. No free lunch for noise prediction. Neural Computation 12 (3), 547–564.
URL https://doi.org/10.1162/089976600300015709

Maurice Herlihy, N. S., 2012. The Art of Multiprocessor Programming, Revised Reprint. Elsevier Science.
URL https://www.ebook.de/de/product/19187472/maurice_herlihy_nir_shavit_the_art_of_multiprocessor_programming_revised_reprint.html

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

Morrison, John E. ; Meliza, L. L., 1999. Foundations of the after action review process. Tech. rep., INSTITUTE FOR DEFENSE ANALYSES ALEXANDRIA VA.

of Queensland, T. U., 2018. Literature review. Tech. rep., The University of Queensland.

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

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

Phan, H., Koch, P., Katzberg, F., Maass, M., Mazur, R., McLoughlin, I., Mertins, A., ???? What makes audio event detection harder than classification?

Phan, H., Maas, M., Mazur, R., Mertins, A., jan 2015. Random regression forests for acoustic event detection and classification. IEEE/ACM Transactions on Audio, Speech, and Language Processing 23 (1), 20–31.

Sherman, M., Jun. 2013. Bogo-sort is sort of slow.

Snoek, J., Larochelle, H., Adams, R. P., 2012. Practical bayesian optimization of machine learning algorithms.

Tseng, S.-Y., Li, J., Wang, Y., Szurley, J., Metze, F., Das, S., ???? Multiple instance deep learning for weakly supervised small-footprint audio event detection.

Vaizman, Y., Ellis, K., Lanckriet, G., oct 2017. Recognizing detailed human context in the wild from smartphones and smartwatches. IEEE Pervasive Computing 16 (4), 62–74.

Virtanen, T., 2010. Acoustic event detection in real life recordings.

Wikipedia, 2019. Tech. rep.

A Full Extracts of Journals

N/A, email correspondence available on request.

B Reflection

about several of the Engineers Australia 16 competencies. [FLAGGED FOR REVIEW]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 outcome I got is barely quantifiable past a common sense guess, but I can take pride in knowing the outcome is quantified and based on the '360 hours' minimum of work to get to that result. I learnt a lot starting with...

B.1 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 implementing felt. The lesson learnt is to have not done an Industry FYP without understanding how it'll affect the pathway to the final thesis.
- 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 of Industry.
- The FYP is a 10 week full time commitment to meet the 360 hours. I potentially "wasted" 6 weeks just identifying the question I was trying to answer in the report. If I hadn't elected to do a Industry FYP, this is potentially work that could be better shared between the supervisor and myself.
- Smaller Scope, get it done in Part A, expand and write in Part B.

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 / featur
098A72A5-E3E5-4F54-A152-BBDA0DF7B694.features_labels.csv.gz |X9100-Y9100 / timestamps 9100 / featur
0A986513-7828-4D53-AA1F-E02D6DF9561B.features_labels.csv.gz |X13060-Y13060 / timestamps 13060 / fea
0BFC35E2-4817-4865-BFA7-764742302A2D.features_labels.csv.gz |X16168-Y16168 / timestamps 16168 / fea
0E6184E1-90C0-48EE-B25A-F1ECB7B9714E.features_labels.csv.gz |X23689-Y23689 / timestamps 23689 / fea
1155FF54-63D3-4AB2-9863-8385D0BD0A13.features_labels.csv.gz |X26374-Y26374 / timestamps 26374 / fea
11B5EC4D-4133-4289-B475-4E737182A406.features_labels.csv.gz |X35219-Y35219 / timestamps 35219 / fea
136562B6-95B2-483D-88DC-065F28409FD2.features_labels.csv.gz |X41437-Y41437 / timestamps 41437 / fea
1538C99F-BA1E-4EFB-A949-6C7C47701B20.features_labels.csv.gz |X47986-Y47986 / timestamps 47986 / fea
1DBB0F6F-1F81-4A50-9DF4-CD62ACFA4842.features_labels.csv.gz |X55361-Y55361 / timestamps 55361 / fea
24E40C4C-A349-4F9F-93AB-01D00FB994AF.features_labels.csv.gz |X60132-Y60132 / timestamps 60132 / fea
27E04243-B138-4F40-A164-F40B60165CF3.features_labels.csv.gz |X65059-Y65059 / timestamps 65059 / fea
2C32C23E-E30C-498A-8DD2-0EFB9150A02E.features_labels.csv.gz |X73575-Y73575 / timestamps 73575 / fea
33A85C34-CFE4-4732-9E73-0A7AC861B27A.features_labels.csv.gz |X79747-Y79747 / timestamps 79747 / fea
3600D531-0C55-44A7-AE95-A7A38519464E.features_labels.csv.gz |X84950-Y84950 / timestamps 84950 / fea
40E170A7-607B-4578-AF04-F021C3B0384A.features_labels.csv.gz |X92599-Y92599 / timestamps 92599 / fea
481F4DD2-7689-43B9-A2AA-C8772227162B.features_labels.csv.gz |X99290-Y99290 / timestamps 99290 / fea
4E98F91F-4654-42EF-B908-A3389443F2E7.features_labels.csv.gz |X102540-Y102540 / timestamps 102540 /
4FC32141-E888-4BFF-8804-12559A491D8C.features_labels.csv.gz |X107519-Y107519 / timestamps 107519 /
5119D0F8-FCA8-4184-A4EB-19421A40DE0D.features_labels.csv.gz |X114136-Y114136 / timestamps 114136 /
5152A2DF-FAF3-4BA8-9CA9-E66B32671A53.features_labels.csv.gz |X120753-Y120753 / timestamps 120753 /
59818CD2-24D7-4D32-B133-24C2FE3801E5.features_labels.csv.gz |X126700-Y126700 / timestamps 126700 /
59EEFAE0-DEB0-4FFF-9250-54D2A03D0CF2.features_labels.csv.gz |X134242-Y134242 / timestamps 134242 /
5EF64122-B513-46AE-BCF1-E62AAC285D2C.features_labels.csv.gz |X138153-Y138153 / timestamps 138153 /
61359772-D8D8-480D-B623-7C636EAD0C81.features_labels.csv.gz |X144232-Y144232 / timestamps 144232 /
61976C24-1C50-4355-9C49-AAE44A7D09F6.features_labels.csv.gz |X152962-Y152962 / timestamps 152962 /
665514DE-49DC-421F-8DCB-145D0B2609AD.features_labels.csv.gz |X162129-Y162129 / timestamps 162129 /
74B86067-5D4B-43CF-82CF-341B76BEA0F4.features_labels.csv.gz |X169427-Y169427 / timestamps 169427 /
78A91A4E-4A51-4065-BDA7-94755F0BB3BB.features_labels.csv.gz |X181423-Y181423 / timestamps 181423 /
797D145F-3858-4A7F-A7C2-A4EB721E133C.features_labels.csv.gz |X185016-Y185016 / timestamps 185016 /
7CE37510-56D0-4120-A1CF-0E23351428D2.features_labels.csv.gz |X194777-Y194777 / timestamps 194777 /
7D9BB102-A612-4E2A-8E22-3159752F55D8.features_labels.csv.gz |X196377-Y196377 / timestamps 196377 /
8023FE1A-D3B0-4E2C-A57A-9321B7FC755F.features_labels.csv.gz |X205566-Y205566 / timestamps 205566 /
806289BC-AD52-4CC1-806C-0CDB14D65EB6.features_labels.csv.gz |X214808-Y214808 / timestamps 214808 /
81536B0A-8DBF-4D8A-AC24-9543E2E4C8E0.features_labels.csv.gz |X221215-Y221215 / timestamps 221215 /
83CF687B-7CEC-434B-9FE8-00C3D5799BE6.features_labels.csv.gz |X230754-Y230754 / timestamps 230754 /
86A4F379-B305-473D-9D83-FC7D800180EF.features_labels.csv.gz |X241492-Y241492 / timestamps 241492 /
96A358A0-FFF2-4239-B93E-C7425B901B47.features_labels.csv.gz |X247311-Y247311 / timestamps 247311 /
9759096F-1119-4E19-A0AD-6F16989C7E1C.features_labels.csv.gz |X257270-Y257270 / timestamps 257270 /
99B204C0-DD5C-4BB7-83E8-A37281B8D769.features_labels.csv.gz |X263308-Y263308 / timestamps 263308 /
9DC38D04-E82E-4F29-AB52-B476535226F2.features_labels.csv.gz |X272994-Y272994 / timestamps 272994 /
A5A30F76-581E-4757-97A2-957553A2C6AA.features_labels.csv.gz |X274661-Y274661 / timestamps 274661 /
A5CDF89D-02A2-4EC1-89F8-F534FDABDD96.features_labels.csv.gz |X280701-Y280701 / timestamps 280701 /
A7599A50-24AE-46A6-8EA6-2576F1011D81.features_labels.csv.gz |X284599-Y284599 / timestamps 284599 /
A76A5AF5-5A93-4CF2-A16E-62353BB70E8A.features_labels.csv.gz |X292119-Y292119 / timestamps 292119 /
B09E373F-8A54-44C8-895B-0039390B859F.features_labels.csv.gz |X300253-Y300253 / timestamps 300253 /
B7F9D634-263E-4A97-87F9-6FFB4DDCB36C.features_labels.csv.gz |X309636-Y309636 / timestamps 309636 /
B9724848-C7E2-45F4-9B3F-A1F38D864495.features_labels.csv.gz |X317262-Y317262 / timestamps 317262 /
BE3CA5A6-A561-4BBD-B7C9-5DF6805400FC.features_labels.csv.gz |X325571-Y325571 / timestamps 325571 /
```

```
BEF6C611-50DA-4971-A040-87FB979F3FC1.features_labels.csv.gz |X329022-Y329022 / timestamps 329022 /
C48CE857-A0DD-4DDB-BEA5-3A25449B2153.features_labels.csv.gz |X334114-Y334114 / timestamps 334114 /
CA820D43-E5E2-42EF-9798-BE56F776370B.features_labels.csv.gz |X341979-Y341979 / timestamps 341979 /
CCAF77F0-FABB-4F2F-9E24-D56AD0C5A82F.features_labels.csv.gz |X350451-Y350451 / timestamps 350451 /
CDA3BBF7-6631-45E8-85BA-EEB416B32A3C.features_labels.csv.gz |X353311-Y353311 / timestamps 353311 /
CF722AA9-2533-4E51-9FEB-9EAC84EE9AAC.features_labels.csv.gz |X356926-Y356926 / timestamps 356926 /
D7D20E2E-FC78-405D-B346-DBD3FD8FC92B.features_labels.csv.gz |X363136-Y363136 / timestamps 363136 /
E65577C1-8D5D-4F70-AF23-B3ADB9D3DBA3.features_labels.csv.gz |X366577-Y366577 / timestamps 366577 /
ECECC2AB-D32F-4F90-B74C-E12A1C69BBE2.features_labels.csv.gz |X370107-Y370107 / timestamps 370107 /
F50235E0-DD67-4F2A-B00B-1F31ADA998B9.features_labels.csv.gz |X372373-Y372373 / timestamps 372373 /
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
--- 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, 'acti
--- 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_na
098A72A5-E3E5-4F54-A152-BBDA0DF7B694.features_labels.csv.gz |X9100-Y9100 / timestamps 9100 / feature_na
0A986513-7828-4D53-AA1F-E02D6DF9561B.features_labels.csv.gz |X13060-Y13060 / timestamps 13060 / feature
0BFC35E2-4817-4865-BFA7-764742302A2D.features_labels.csv.gz |X16168-Y16168 / timestamps 16168 / feature
0E6184E1-90C0-48EE-B25A-F1ECB7B9714E.features_labels.csv.gz |X23689-Y23689 / timestamps 23689 / feature
1155FF54-63D3-4AB2-9863-8385D0BD0A13.features_labels.csv.gz |X26374-Y26374 / timestamps 26374 / feature
11B5EC4D-4133-4289-B475-4E737182A406.features_labels.csv.gz |X35219-Y35219 / timestamps 35219 / feature
136562B6-95B2-483D-88DC-065F28409FD2.features_labels.csv.gz |X41437-Y41437 / timestamps 41437 / feature
1538C99F-BA1E-4EFB-A949-6C7C47701B20.features_labels.csv.gz |X47986-Y47986 / timestamps 47986 / feature
1DBB0F6F-1F81-4A50-9DF4-CD62ACFA4842.features_labels.csv.gz |X55361-Y55361 / timestamps 55361 / feature
24E40C4C-A349-4F9F-93AB-01D00FB994AF.features_labels.csv.gz |X60132-Y60132 / timestamps 60132 / feature
27E04243-B138-4F40-A164-F40B60165CF3.features_labels.csv.gz |X65059-Y65059 / timestamps 65059 / feature
2C32C23E-E30C-498A-8DD2-0EFB9150A02E.features_labels.csv.gz |X73575-Y73575 / timestamps 73575 / feature
33A85C34-CFE4-4732-9E73-0A7AC861B27A.features_labels.csv.gz |X79747-Y79747 / timestamps 79747 / feature
3600D531-0C55-44A7-AE95-A7A38519464E.features_labels.csv.gz |X84950-Y84950 / timestamps 84950 / feature
```

```
40E170A7-607B-4578-AF04-F021C3B0384A.features_labels.csv.gz |X92599-Y92599 / timestamps 92599 / feature
481F4DD2-7689-43B9-A2AA-C8772227162B.features_labels.csv.gz |X99290-Y99290 / timestamps 99290 / feature
4E98F91F-4654-42EF-B908-A3389443F2E7.features_labels.csv.gz |X102540-Y102540 / timestamps 102540 / feat
4FC32141-E888-4BFF-8804-12559A491D8C.features_labels.csv.gz |X107519-Y107519 / timestamps 107519 / feat
5119D0F8-FCA8-4184-A4EB-19421A40DE0D.features_labels.csv.gz |X114136-Y114136 / timestamps 114136 / feat
5152A2DF-FAF3-4BA8-9CA9-E66B32671A53.features_labels.csv.gz |X120753-Y120753 / timestamps 120753 / feat
59818CD2-24D7-4D32-B133-24C2FE3801E5.features_labels.csv.gz |X126700-Y126700 / timestamps 126700 / feat
59EEFAE0-DEB0-4FFF-9250-54D2A03D0CF2.features_labels.csv.gz |X134242-Y134242 / timestamps 134242 / feat
5EF64122-B513-46AE-BCF1-E62AAC285D2C.features_labels.csv.gz |X138153-Y138153 / timestamps 138153 / feat
61359772-D8D8-480D-B623-7C636EAD0C81.features_labels.csv.gz |X144232-Y144232 / timestamps 144232 / feat
61976C24-1C50-4355-9C49-AAE44A7D09F6.features_labels.csv.gz |X152962-Y152962 / timestamps 152962 / feat
665514DE-49DC-421F-8DCB-145D0B2609AD.features_labels.csv.gz |X162129-Y162129 / timestamps 162129 / feat
74B86067-5D4B-43CF-82CF-341B76BEA0F4.features_labels.csv.gz |X169427-Y169427 / timestamps 169427 / feat
78A91A4E-4A51-4065-BDA7-94755F0BB3BB.features_labels.csv.gz |X181423-Y181423 / timestamps 181423 / feat
797D145F-3858-4A7F-A7C2-A4EB721E133C.features_labels.csv.gz |X185016-Y185016 / timestamps 185016 / feat
7CE37510-56D0-4120-A1CF-0E23351428D2.features_labels.csv.gz |X194777-Y194777 / timestamps 194777 / feat
7D9BB102-A612-4E2A-8E22-3159752F55D8.features_labels.csv.gz |X196377-Y196377 / timestamps 196377 / feat
8023FE1A-D3B0-4E2C-A57A-9321B7FC755F.features_labels.csv.gz |X205566-Y205566 / timestamps 205566 / feat
806289BC-AD52-4CC1-806C-0CDB14D65EB6.features_labels.csv.gz |X214808-Y214808 / timestamps 214808 / feat
81536B0A-8DBF-4D8A-AC24-9543E2E4C8E0.features_labels.csv.gz |X221215-Y221215 / timestamps 221215 / feat
83CF687B-7CEC-434B-9FE8-00C3D5799BE6.features_labels.csv.gz |X230754-Y230754 / timestamps 230754 / feat
86A4F379-B305-473D-9D83-FC7D800180EF.features_labels.csv.gz |X241492-Y241492 / timestamps 241492 / feat
96A358A0-FFF2-4239-B93E-C7425B901B47.features_labels.csv.gz |X247311-Y247311 / timestamps 247311 / feat
9759096F-1119-4E19-A0AD-6F16989C7E1C.features_labels.csv.gz |X257270-Y257270 / timestamps 257270 / feat
99B204C0-DD5C-4BB7-83E8-A37281B8D769.features_labels.csv.gz |X263308-Y263308 / timestamps 263308 / feat
9DC38D04-E82E-4F29-AB52-B476535226F2.features_labels.csv.gz |X272994-Y272994 / timestamps 272994 / feat
A5A30F76-581E-4757-97A2-957553A2C6AA.features_labels.csv.gz |X274661-Y274661 / timestamps 274661 / feat
A5CDF89D-02A2-4EC1-89F8-F534FDABDD96.features_labels.csv.gz |X280701-Y280701 / timestamps 280701 / feat
A7599A50-24AE-46A6-8EA6-2576F1011D81.features_labels.csv.gz |X284599-Y284599 / timestamps 284599 / feat
A76A5AF5-5A93-4CF2-A16E-62353BB70E8A.features_labels.csv.gz |X292119-Y292119 / timestamps 292119 / feat
B09E373F-8A54-44C8-895B-0039390B859F.features_labels.csv.gz |X300253-Y300253 / timestamps 300253 / feat
B7F9D634-263E-4A97-87F9-6FFB4DDCB36C.features_labels.csv.gz |X309636-Y309636 / timestamps 309636 / feat
B9724848-C7E2-45F4-9B3F-A1F38D864495.features_labels.csv.gz |X317262-Y317262 / timestamps 317262 / feat
BE3CA5A6-A561-4BBD-B7C9-5DF6805400FC.features_labels.csv.gz |X325571-Y325571 / timestamps 325571 / feat
BEF6C611-50DA-4971-A040-87FB979F3FC1.features_labels.csv.gz |X329022-Y329022 / timestamps 329022 / feat
C48CE857-A0DD-4DDB-BEA5-3A25449B2153.features_labels.csv.gz |X334114-Y334114 / timestamps 334114 / feat
CA820D43-E5E2-42EF-9798-BE56F776370B.features_labels.csv.gz |X341979-Y341979 / timestamps 341979 / feat
CCAF77F0-FABB-4F2F-9E24-D56AD0C5A82F.features_labels.csv.gz |X350451-Y350451 / timestamps 350451 / feat
CDA3BBF7-6631-45E8-85BA-EEB416B32A3C.features_labels.csv.gz |X353311-Y353311 / timestamps 353311 / feat
CF722AA9-2533-4E51-9FEB-9EAC84EE9AAC.features_labels.csv.gz |X356926-Y356926 / timestamps 356926 / feat
D7D20E2E-FC78-405D-B346-DBD3FD8FC92B.features_labels.csv.gz |X363136-Y363136 / timestamps 363136 / feat
E65577C1-8D5D-4F70-AF23-B3ADB9D3DBA3.features_labels.csv.gz |X366577-Y366577 / timestamps 366577 / feat
ECECC2AB-D32F-4F90-B74C-E12A1C69BBE2.features_labels.csv.gz |X370107-Y370107 / timestamps 370107 / feat
F50235E0-DD67-4F2A-B00B-1F31ADA998B9.features_labels.csv.gz |X372373-Y372373 / timestamps 372373 / feat
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, '
----------
----------
--- 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

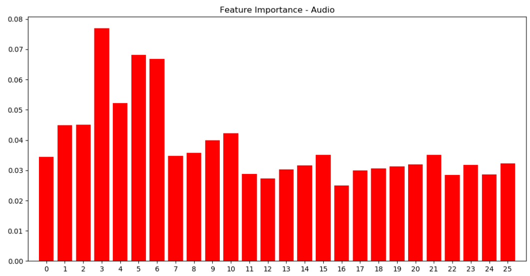

Figure 19: A Random Tree Classifier was used to evaluate feature importance, during the design phase of the main classifier. Features were Audio (26 bins)

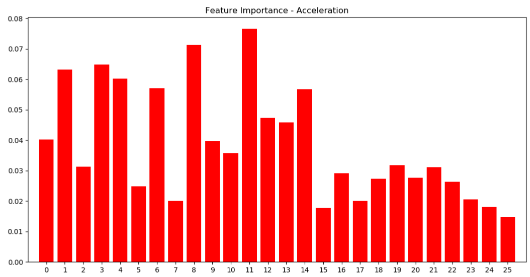

Figure 20: A Random Tree Classifier was used to evaluate feature importance, during the design phase of the main classifier. Features were Accelerometer(26)

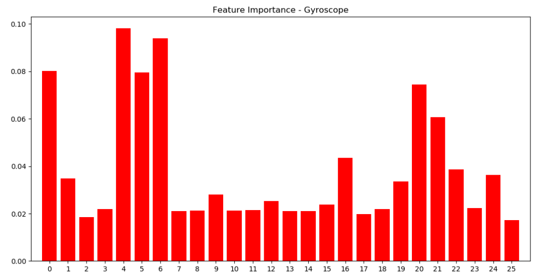

Figure 21: A Random Tree Classifier was used to evaluate feature importance, during the design phase of the main classifier. Features were Gyroscope(26)

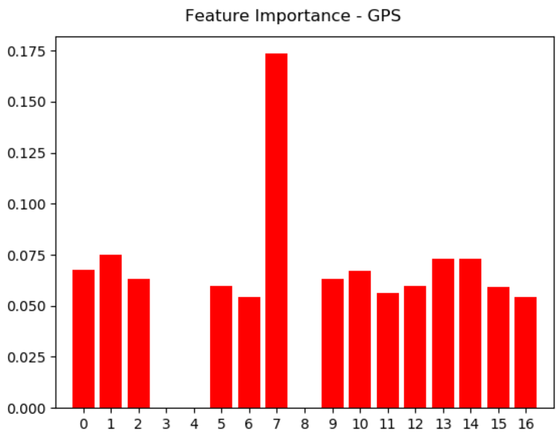

Figure 22: A Random Tree Classifier was used to evaluate feature importance, during the design phase of the main classifier. Features were GPS (16)

E Trade-Review of "Online vs Offline" processing

F GridSearch Attempts

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 9: ExtraSensory Labels Vaizman et al. (2017)

#	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