Electronics and Computer Science
Faculty of Physical Sciences and Engineering
University of Southampton

Charlie Steptoe - cs5n17
Aran McConnell - ajm2g17
Yue Kew - ysk2a15
Rohan Bungre - rsb1g17

22th Jan 2021

Recognising Activities of African Wild Dogs using Machine Learning

Project supervisor: Dr Srinandan Dasmahapatra
Second examiner: Dr Jize Yan

A Group Design Project Report submitted for the Award of
MEng Computer Science
MEng Electronic Engineering
MEng Electronic Engineering with Artificial Intelligence
MEng Electronic Engineering with Artificial Intelligence

Abstract

The African Wild Dog is currently an endangered animal, meaning conservation efforts are at an all-time high. This report focuses on ways to recognise the activities of UK domestic large dogs using machine learning techniques. The data pipeline and models created will be used as a proof of concept for future conservation projects involving African Wild Dogs.

A 3D printed mount was produced to attach an IMU device to the collars of large dogs. This device was then used to collect 6-Axis IMU data from these dogs, whilst carrying out the following activities: walking, trotting, galloping, barking, grooming, eating, sleeping, and hunting. The collected data was manually labelled using video from each corresponding session. The labelled dataset was then fed into the project's machine learning phase.

Using this dataset, AI models were used to create robust activity classifiers. Unsupervised techniques based on the raw data's density and Euclidean distance were found to be the most inaccurate. Using supervised techniques with extracted Fourier and Wavelet Transform and Statistical features was found to produce an accurate activity classifier. The final classification model had a 99% accuracy on seen session data and up to 72% accuracy on unseen sessions.

Word Count - 19915
As measured using Overleaf's Word Count feature

Contents

List of Figures

Definitions and Abbreviations

IMU - Inertial Measurement Unit

AI - Artificial Intelligence

ML - Machine Learning

DL - Deep Learning

Acknowledgements

Throughout the duration of this project we have received a great deal of support and assistance, ensuring the project's completion and success.

We would first like to thank our supervisor Dr Srinandan Dasmahapatra, for the support and guidance offered to us over the project. We appreciated being able to work on a project that involves our combined passion for Artificial Intelligence.

We would also like to thank our External Customer, Mafic Ltd, for providing us with a clear set of goals and tools for this project. We would especially like to extend our thanks to Will Woodhead and Jake Minns for sharing their expertise when needed.

We wish to show our gratitude to the lab technicians working during the pandemic, to Vlad Velici and to the members of Hampshire Dog Club who enabled us to move forward with critical phases of the project. Without them our project would have not been a success.

In addition, we would like to pay special regards to all of our friends and families for the encouragement and always providing a happy distraction to rest our minds outside of the project.

Statement of Originality

- I have read and understood the ECS Academic Integrity information and the University's Academic Integrity Guidance for Students.
- I am aware that failure to act in accordance with the Regulations Governing Academic Integrity may lead to the imposition of penalties which, for the most serious cases, may include termination of programme.
- I consent to the University copying and distributing any or all of my work in any form and using third parties (who may be based outside the EU/EEA) to verify whether my work contains plagiarised material, and for quality assurance purposes.

You must change the statements in the boxes if you do not agree with them.

We expect you to acknowledge all sources of information (e.g. ideas, algorithms, data) using citations. You must also put quotation marks around any sections of text that you have copied without paraphrasing. If any figures or tables have been taken or modified from another source, you must explain this in the caption and cite the original source.

I have acknowledged all sources, and identified any content taken from elsewhere.

If you have used any code (e.g. open-source code), reference designs, or similar resources that have been produced by anyone else, you must list them in the box below. In the report, you must explain what was used and how it relates to the work you have done.

The project utilised Python and the following Libraries:

- **sys, os, glob**
- **argparse**
- **time**
- **pandas**
- **pickle**
- **numpy**
- **scikit-image, scikit-learn**
- **pywt**
- **imbalanced-learn**
- **tensorflow, keras**
- **seaborn, matplotlib**

You can consult with module teaching staff/demonstrators, but you should not show anyone else your work (this includes uploading your work to publicly-accessible repositories e.g. Github, unless expressly permitted by the module leader), or help them to do theirs. For individual assignments, we expect you to work on your own. For group assignments, we expect that you work only with your allocated group. You must get permission in writing from the module teaching staff before you seek outside assistance, e.g. a proofreading service, and declare it here.

I did all the work myself, or with my allocated group, and have not helped anyone else.

We expect that you have not fabricated, modified or distorted any data, evidence, references, experimental results, or other material used or presented in the report. You must clearly describe your experiments and how the results were obtained, and include all data, source code and/or designs (either in the report, or submitted as a separate file) so that your results could be reproduced.

The material in the report is genuine, and I have included all my data/code/designs.

We expect that you have not previously submitted any part of this work for another assessment. You must get permission in writing from the module teaching staff before re-using any of your previously submitted work for this assessment.

I have acknowledged all sources, and identified any content taken from elsewhere.

If your work involved research/studies (including surveys) on human participants, their cells or data, or on animals, you must have been granted ethical approval before the work was carried out, and any experiments must have followed these requirements. You must give details of this in the report, and list the ethical approval reference number(s) in the box below.

ERGO approval documents clearly shown in Appendix A **ERGO Reference Number: 61647**

ECS Statement of Originality Template, updated August 2018, Alex Weddell aiofficer@ecs.soton.ac.uk

Chapter 1

Introduction

1.1 Project Aim

The aim of this project was to recognise the activities of African Wild Dogs using machine learning techniques. The project involved creating a robust AI model that was able to classify UK large dog activities to a specified accuracy. The data from the UK large dogs was collected, labelled and processed as part of the project's scope, using tools and devices provided by the external partner. The eventual aim was to use the project's success as a proof of concept for the employment of such a data pipeline on future conservation and wildlife monitoring projects. This would include the external customers goal to recognise African Wild Dog activities.

1.2 External Partner

External Partner - Mafic Ltd

Principal Contacts - Will Woodhead and Jake Minns

Mafic are a small business with the aim to improve health, safety, and productivity in the industrial workplace through the use of modern technologies. Mafic have found commercial success in the construction industry by tracking the movement of the builders to predict activities and flow within a work site. This is accomplished using data collected from a 9-axis IMU device attached to the head of builders within the work site. This data is passed through a complex deep learning model implemented by Mafic to perform the activity and workflow predictions.

1.3 African Wild Dogs

The African Wild Dog is a species native to sub-Saharan Africa. The wild dog is one of the world's most endangered mammals so conservation efforts are stronger than ever before. African wild dogs live in packs and are very social, with some known to share food and to assist weak or ill members. Social interactions are common, and the dogs communicate by touch, actions, and vocalisations. This is a trait that is followed by UK large dogs to some extent, so their activity classes would be similar.

1.4 Product Usage

Due to the success in the construction industry, Mafic was approached by ITV to see if their current technology could be adapted for the task of African Wild Dog activity recognition. If possible, the technologies would be used in TV documentaries and by animal conservationists to aide in their work. This product would help reduce unnecessary intrusion into many wild habitats, thus reduce the interference of wild animals whilst monitoring them. Furthermore, this product would also allow for a reduction in camera and labour costs as there will be no need to film the wild animals to work out what they are doing during a documentary's production.

1.5 Product Competition

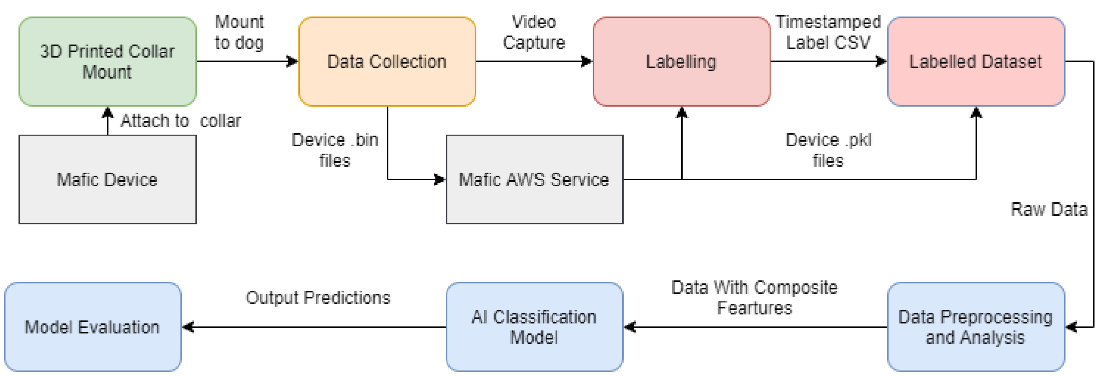
When researching potential market competition, no viable competition was found that offered the hardware and software complete solution to recognise African Wild Dog or other animal activities. In general, most competition focused purely on the hardware aspects of tracking an animal, but neglected to perform any analytics or prediction with said information. This made the work done for Mafic unique and novel.

Chapter 2

System Design & Specification

The decision was made to split the project into 4 main phases. These were: Device mount creation, data collection, data labelling and AI modelling. Each phase was broken down into a collection of requirements as stated in the specification. Each subsequent phase would rely on the previous phase being completed before moving on. Figure 2.1 demonstrates the workflow between each phase of the project. It was crucial that this structure was followed to ensure that the correct deadlines were met to make good progress over the semester.

Figure 2.1: System Design



To help define the goals and requirements of the project a detailed specification was made. This was defined by both Mafic and the group as a collective to ensure the project was both useful for Mafic and fulfilling as a 4th year GDP. For each phase of the project requirements were outlined that, if met, would mean the phase was a success. Stretch goals were also stated for each phase as, whilst not essential to the project's success, it would improve the quality of the overall project.

	Project Objective	**Specification Requirement**
1. Device Mount Creation	1.1. Mount Option Evaluation	– At least 3 possible connector options will be identified. – All options evaluated by cost and data collection plausibility. – The versatility of mount options on domestic and wild dogs will be compared.
	1.2. 3D Model Creation	– A casing of the most suitable form must be modelled to fix the **Mafic** device to a range of dog collar sizes. – Model files will be exported in the ".stl" format for ease of printing. – (Stretch) The model may be created to allow for weather-proofing of the device in deployment.
	1.3. Mount Production	– Any resultant 3D model files will likely be 3D-printed. – (Stretch) Weatherproofing will be applied to the casing.
	1.4. Testing & Adjustment	– Initial testing of casing will consist of manual manipulation of the device on test collars. – Further testing of casing is to take place during data collection. – Any significant slippage or device failings will result in design adjustments made to collar mount.
2. Data Collection	2.1. Activity Circuit Creation	– Circuits should include Walking, Trotting, Galloping, Grooming, Barking, Eating, Sleeping. – (Stretch) Circuits are to include Hunting activities.
	2.2. Conduct Activity Circuits with Device	– A camera producing clear and legible footage should be used for session duration. – All activity sessions should have time-synchronised video to accompany device data. – All activity sessions will be conducted with suitable reproducibility controls in place. – Device traces will be stored on **Mafic** hosted AWS services.
3. Data Labelling	3.1. Recording Pre-processing	– Video files may be down-sampled to facilitate efficient storage.
	3.2. Activity and Sensor Data Labelling	– All labelling should be conducted using the latest version of **Mafic**'s labelling tool. – All collected data is to be labelled using corresponding video. – Standard activities (Walking, Trotting, Galloping, Grooming, Barking, Eating, Sleeping) should be labelled. – Labelling process to be standardised by using a minimum number of labellers. – Labelled time interval data will be exported from the tool in CSV format. – (Stretch) Hunting activities should be labelled (if gathered). – (Stretch) Automate unlabelled data retrieval from AWS storage.

Table 2.1: Project Specification & Detailed Requirements

4. AI Model Generation	4.1. Data set Formatting	– Generate 50Hz labelled device data (through labelled interval data) for 6-axis IMU. – Dataset features not encompassed by 6-axis IMU are to be removed. – Standardisation and normalisation pre-processing to ensure scale invariance will be conducted. – (Stretch) Create a database to store the dataset locally or using cloud technologies. – (Stretch) Subsampled variants of labelled device data should be generated at 25Hz and 10Hz frequencies.
	4.2. AI Model Architecture	– Created artificial intelligence model should have an overall accuracy greater than 80%. – Model should be capable of predicting activities: Walking, Trotting, Galloping, Grooming, Barking, Eating, Sleeping. – Prediction model must be programmed in the Python programming language. – Training and model execution must be capable of running on an everyday laptop (Non-GPU intensive methods are required). – (Stretch) Model should be capable of predicting hunting.
	4.3. AI Model Performance Evaluation	– Compare each experimental model's classification accuracy using a variety of classification metrics and cross validation techniques. – (Stretch) The model should have its accuracy and recall compared when trained and tested on subsampled data of 25Hz and 10Hz frequencies. – (Stretch) Individual feature importance should be compared and any potential input domain reduction documented.

Project Specification & Detailed Requirements (Cont.)

Chapter 3

Device Mount Creation

In the first phase of the project, the group were tasked with creating a general purpose, adaptable dog-collar mount for a 6-axis IMU device, to be provided by the customer. As laid out in the project specification, the group required at least three designs to be identified and evaluated for plausible use with both domestic and African Wild dogs. Out of these basic designs, multiple of which had been used in previous projects from other institutions, one novel design was chosen to enter the prototyping stage. Having modelled this collar mount to surround existing device casings, a first prototype was 3D-printed and used for a large portion of the project's data collection. This mount, although satisfying the necessary criteria outlined by the group, seemed to be relatively brittle and fragile. Consequently, redesigns were needed throughout the data collection phase, resulting in a final, much sturdier and easier to use mount. The finalised 3D model files are presented to the customer in submission of this project.

3.1 Mount Designs Evaluation

In order to evaluate the three mount design options, as in the project specification, the group outlined a set of three criteria any chosen option would be required to fulfil before being taken to a prototyping stage. These criteria were:

1. Mount must not move relative to dog's neck when attached to collar.
2. Mount must not significantly impede natural movement of dog.
3. Mount design must be alterable to fit a range of collar sizes.

An initial concern within the group was that there may be a potential trade-off between resisting movement relative to the dog's neck (which is typically solved through adding extra weight to the bottom of the collar) and facilitating natural movement (which may be impeded by any excessive weight). One alternative solution to this discrepancy would be to demonstrate the mounting of the device to a canine harness, relying on the harness' inability to rotate. This solution has been used previously in studies involving a very similar IMU device to that which was given to the group [17].

Discussing this potential design with the customer, it was decided that, although such a design would be suitable for domestic dogs, when deployed for use on African Wild Dogs, a harness would quickly become unsuitable for typical conservation efforts using tracking collars (Figure 3.1) [24]. With this conclusion, the group resolved to develop a mounting system that was suitable for both domestic and African Wild dogs, so as to minimise differences in data collection methodologies and, in so doing, improve accuracy of data gathered.

Figure 3.1: Standard Collar used on African Wild Dogs in Conservation

Another design posed in existing literature involves the creation of a device casing that allows the collar to feed through either side and fold back on itself, thereby becoming a standalone section of an existing dog collar (Figure 3.2) [20]. Although this mechanism is a sturdy method of attaching the device to a collar that a researcher has prior access to and/or is flexible enough to fold onto itself, reviewing the collars used in conservation efforts once again reveals this would not be possible. African Wild dog collars are typically tightly fastened and are constructed from tough, laminated leather to ensure they do not get removed inadvertently, hence this second option would also be infeasible when deployed.

Figure 3.2: Collar Attachment of IMU Device as seen in [20]

The third and final design proposed by our team is built upon the previous two models, in addition to taking inspiration from existing dog collar attachments and, in particular, pet 'slide-on' collar ID tags, which are typically used for dogs in agility competitions and events (Figure 3.3) [1]. These lightweight tags are designed to hold securely to a collar using the existing tension in a typical canine collar, whilst also being easily deliberately attachable and detachable without removing the collar from the dog (and hence not manipulating the dog's neck by an excessive amount). This new feature is also extremely valuable when, due to the Covid-19 pandemic of 2020, the group members conducting data collection sessions were limiting exposure to participant dogs and their owners.

This new design also allows for a similarly secure attachment when the collar is stiffer but is able to be removed (as will typically be the case when working with African Wild dog collars). One experienced disadvantage to this design is the increased fragility of the mount 'arms' and so extra support must be provided to this region.

Figure 3.3: Typical Design for Agility Pet ID Tag

3.2 3D Modelling and Production

Once the group had decided on a final concept for attaching the IMU device to a range of collars, both suitable for African Wild and domestic dogs, the group used the open-source 3D modelling tool, Blender, to construct a basic mesh [9]. This mesh was created alongside an existing imported model of the IMU device casing, as provided by the customer. Using this existing model, the group was able to obtain a reasonable level of accuracy to ensure that the device casing would fit inside any initial prototype mounts that were created. In addition, as the IMU device is charged using wireless charging, it was desirable for the mount design to have a slightly thinner back portion so as to allow for this.

The 3D model was then exported so as to be in the industry standard, STL file format, ready for the customer should the initial design not need to be changed, and to provide the model in an understandable format for those members of the University performing the printing. The completed initial design is visible in Figure 3.4.

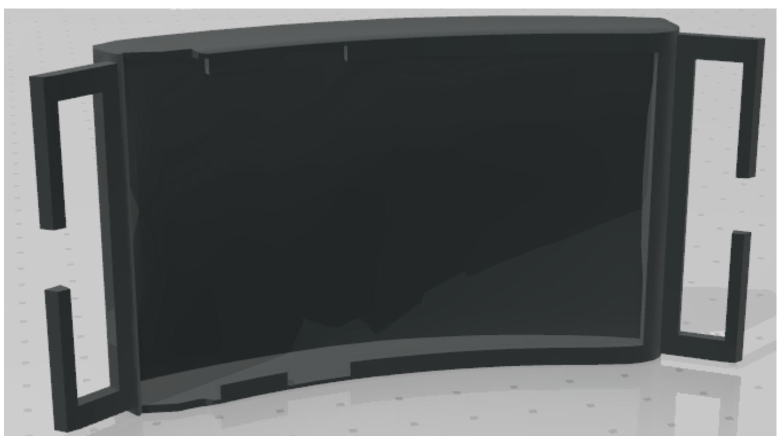


Figure 3.4: Initial Prototype Design for Device Mount

After modelling was completed, the model was passed on to the University lab staff (due to Coronavirus restrictions) and printed on University machines using PLA material, so chosen for quick prototyping and relatively low cost. It is suggested that a more thorough analysis of chosen materials be conducted in the future and before deployment to more rugged environments.

3.3 Testing and Redesign

Due to restricted time and number of participants, all device mount testing was conducted concurrently with data collection. This afforded the group a large amount of data on the mount performance in a short space of time, allowing for rapid iterations, should an element require changing. The limiting factor to extra prototyping was the time required to order and have new designs printed.

Nevertheless, three notable changes were identified over the course of data collection. The first and most significant relates to the fragility of the mechanism ‘arms’. During the first session attached to a dog one of the retrospectively thin arms created to surround the collar snapped and the member conducting the session was required to find an alternative method to secure the device. Consequently, the next prototype possessed much thicker and supported collar-surrounding portions.

Two other changes made over the sessions are the widening of the device-containing well in the mount (To ensure no undue pressure was placed on the device casing) and the extruded sections surrounding this well were reduced, allowing for less difficulty in replacing the device in the mount if required (On initial prototypes, these tabs were typically broken when affixing the device).

There was no clear need to adjust the design in addition to this, but the device has not yet been tested in more challenging settings and thus, unexpected alterations may be necessary. The provided STL files will support this. The final adjustments can be seen in Figure 3.5.

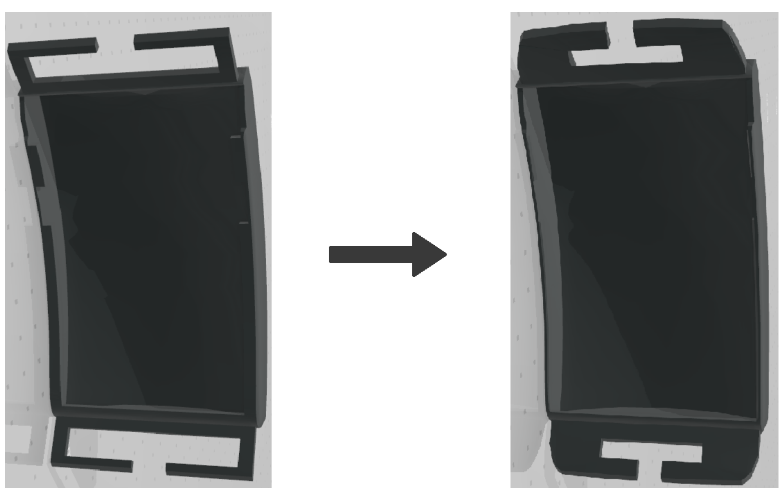


Figure 3.5: Redesign of Device Mount (Left - Before) vs (Right - After)

3.4 Comparison with Specification

Specification 1.1

Section 1.1 of the project specification entails the evaluation of at least three possible mount design options, by way of plausibility for us on domestic and African Wild Dogs. To this end, the group have reviewed three different mount designs, alongside using existing literature and previous studies to understand the benefits or disadvantages of each method. Our final design provided the versatility necessary to fit a range of collars and satisfied our three criteria for evaluation of suitability.

Specification 1.2 & 1.3

Sections 1.2 & 1.3 of the project specification outlines the production of 3D models for production of the chosen mounting system and potential printing stages of this phase. The deliverable from this project phase is the completed 3D model of the refined device mount in STL format, which has been laid out in these sections of the specification. Furthermore, as anticipated, the designs were 3D printed using University 3D printing facilities, as per the specification.

However, the stretch objectives relating to weather-proofing device and mount were unsuccessful. This has unfortunately not been fulfilled due to the ensuing difficulty of rapidly printing prototypes and iterating over design caused by Coronavirus lockdown restrictions. It was decided that if weatherproofing was necessary, this could be performed on the device casing, as opposed to the mount, and hence the mount was designed to affix to the device casing itself.

Specification 1.4

Section 1.4 of the specification relates to testing and iteration phase of the device mount design. As evidenced in Subsection 3.3, there were three major alterations made to the mount design that significantly increased the usability and robustness of the the final mount. Testing of the design was conducted concurrently with initial data collection to minimise delay between initial and subsequent designs, which enabled the team to make the aforementioned improvements. As a result, the final design of the collar mount in the deliverable is significantly better suited to rugged environments.

Chapter 4

Data Collection

In phase 2 of project, the group were tasked to collect data on domestic dogs in the UK. The dogs are required to carry out circuits of activities for each session. All sessions are logged using an IMU device provided by Mafic and recorded using a camera. To date, there is no dataset on dog activities widely available to the public. The purpose of this phase is to carry out sessions of various dog activities for activities to be labelled in the next chapter, Data Labelling.

4.1 Experimental Setup

Equipment

All sessions are filmed using a 1080p KONIG Action Camera and logged using a sensor device comprised of tri-axial accelerometer and gyroscope with a sampling rate of 50Hz. The sensor device is housed in a 3D printed casing, attached on an adjustable dog collar. A TP-Link Archer C50 Dual-band Wireless WiFi router is used for uploading sensor readings to a cloud database.

Subjects

8 healthy adult dogs of several breeds are selected for data collection. Breeds include German Shepherd, Belgian Shepherd (Malinois), Carpathian Shepherd, Romanian Mioritic Shepherd and Labrador Retriever, which are all considered as large dog breeds. 3 out of 8 are purebred Belgian Shepherds while the others are crossbreeds, based on Table 4.1. Dogs of unknown dog breeds are adopted dogs and are likely crossbreeds of Carpathian Shepherd and Mioritic Shepherd. All subjects selected are of 2 to 4 years of age and are considered to be in adulthood.

Height is measured from the highest point of shoulder to the base of the paws. In our study, the subjects have heights ranging from 55 to 70cm. Their weight ranges from 25 to 35kg. Subjects can be split into two categories, pet dogs and professionally trained security dogs. All Belgian and German Shepherds in this study are professionally trained for security purposes, typically for guarding and detection, by a certified dog trainer approved by NASDU (National Association of Security Dog Users). According to World Wildlife Fund (WWF), African Wild Dogs typically weigh about 18-32kg and are measured at 76cm in height [3]. In comparison to selected subjects for this study, African Wild Dogs are similar in weight but marginally taller.

Dog Breeds	Age	Gender (M/F)	Weight (kg)	Height (cm)
Belgian Shepherd (Malinois)	4	M	35	64
Belgian Shepherd (Malinois)	2	M	32	59
Belgian Shepherd (Malinois)	2	M	33	66
German Shepherd x Belgian Shepherd	4	F	28	56
Labrador Retriever x German Shepherd	3	F	34	58
Unknown crossbreeds	2	M	30	68
Unknown crossbreeds	2	F	31	55
Unknown crossbreeds	2	F	26	55

Table 4.1: Overview of subjects participated in the study.

4.2 Activity Circuit Creation

Data are collected with the aim of recognizing African Wild Dogs activities. However, the wild dogs are only found in Coastal East Africa [3], hence domestic dogs of large breeds were selected to carry out circuits of activities. 8 non-overlapping activities were defined. These activities include walking, trotting, galloping, grooming, barking, eating, sleeping and hunting. Data collected are all based around these defined activities. There are two different groups of dogs in our study, pet dogs and professionally trained security dogs. Due to different characteristics of both groups, different schema was used for session filming.

Terminology

Below compiles some terminologies used in this section.

- **Activity Circuit/ Designed Circuit**: Refer to Table 4.2.
- **Schema**: A schema refers to selected designed circuits for a session. For example, Schema 1 would consist of designed circuits 1,2,3,4 and Schema 2 would comprise designed circuits 2,3,5,6.

– **Session**: A session is defined as a subject undertaking all designed circuits in a schema.

– **Session Filming**: The action of filming a subject undertaking a series of designed circuits in a session.

4.2.1 Session Filming Schema

In order to obtain defined activities, a circuit of activities shown in Table 4.2 is executed by the subjects, with the exception of some, according to the group of dogs the particular subject belongs to, pet dog or trained dog. Table 4.3 compiles a guide for choosing designed circuits in a session for subjects of different categories and characteristics. For a professionally trained dog, a session would consist of designed circuits 1,2,5,7 and 9. Executing these activities would acquire walking, trotting, galloping, barking, and eating data as a result. These activities can be done under commands of a dog trainer. Figure 4.1 shows some of the designed circuits.

If a dog is trained for detection and bite work, design circuits 11 and 12 will be carried out. Figure 4.2 shows the process of a designed circuit for object searching. The dog trainer would start by bringing an object to the dog to sniff the scent of the object, allowing it to recognize. The object used here was a toy gun and a mat in the container holding the toy gun was brought to the dog to recognize. The object would then be hidden away somewhere unnoticeable while the dog is being distracted. The dog would start searching for the object once it is allowed to. These set of activities concludes the detection of prey in hunting behaviour. Furthermore, capturing prey is represented by designed circuit 12, object biting. The dog trainer would either wear a bite sleeve and command the dog to bite or he would use a bite pillow. The dog would only let go of the bite sleeve or bite pillow under command. Figure 4.3 and 4.4 shows designed circuit 12 executed by a security dog.

For a pet dog, a typical session would consist of designed circuits 1,6, 7 and 8. Circuits 2,3,4 are filler circuits for obtaining additional data for walking, galloping and trotting. Circuits are also selected based on their personality, generally calmer or more active. For example, a calm dog would require additional “Play chase” and “Dog running” circuits whereas an active dog would require additional “Dog walking” circuit in their respective sessions. Data on “Sleeping” is only collected if the dog handler agrees to indoor filming.

No.	Designed Circuit	Activity	Description
1	Play fetch	Walking, Trotting, Galloping	Handler throws an object to a distance and the dog would fetch the object back.
2	Dog walking	Walking	Handler walks the dog on a leash.
3	Play chase	Trotting, Galloping	Handler starts running at a faster pace, allowing the dog to chase from behind.
4	Dog running	Trotting, Galloping	Handler runs with the dog.
5	Dog grooming	Grooming	Handler uses a brush to groom the dog.
6	Playing with other dogs	Barking, Walking, Trotting	Handler allows the dog to social with other dogs.
7	Barking	Barking	Handler commands the dog to speak (bark).
8	Giving treats	Eating	Handler gives treats to the dog.
9	Eating a meal	Eating	Dog eats dog biscuits from a bowl.
10	Sleeping	Sleeping	Dog sleeps in their own kennel/ with their owner at home.
11	Object searching	Hunting	Handler first allows the dog to recognize the scent of the object, then hides the object. The dog will search for the object that it has just sniffed.
12	Object biting	Hunting	Handler wears a bite sleeve/ uses a bite pillow, signalling the dog to bite on the item for an extended period of time.

Table 4.2: A detailed activity circuit creation.

No.	Designed Circuit	Dog Category		Dog Characteristic		Indoor Filming		Notes
		Pet	Trained	Calm	Active	Yes	No	
1	Play fetch	/	/					
2	Dog walking		/		/			
3	Play chase			/				
4	Dog running			/				
5	Grooms dog		/			/		
6	Playing with other dogs	/						A session on trained dogs are typically done 1-to-1.
7	Barking	/	/					Normally dogs bark when triggered by some form of stimulus but trained dogs are able to bark under command.
8	Giving treats	/						Treats are given because sessions with pet dogs are typically done in an open field (treats are easier to carry around).
9	Eating a meal		/			/		
10	Sleeping					/		
11	Object searching		/					
12	Object biting		/					

Table 4.3: Schema for selecting designed circuits for a session.

Figure 4.1: Examples of Designed Circuits. (Left - Walking), (Middle - Barking (hand command circled in yellow)), (Right - Grooming)

Figure 4.2: Process of Object Searching. (1 - Recognizing Scent of Object), (2 - Searching for Object), (3 - Object Found), (4 - Revelation of Object)

Figure 4.3: Process of Object Biting Designed Circuit. (Left - Recognize Object), (Right - Signals to Bite Pillow)

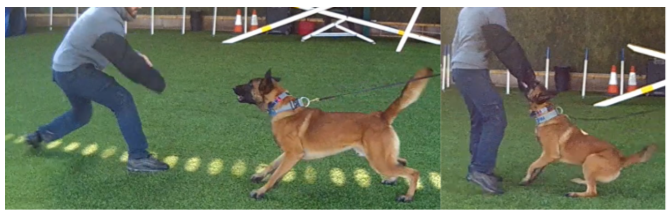


Figure 4.4: Process of Object Biting Designed Circuit. (Left - Recognize Object), (Right - Signals to Bite Sleeve)

4.3 Data Collection Protocol

Data were collected for approximately one month, from November to December 2020 at two open field locations, a shaded dog training canopy with two acres wide grass field and a park of 32 hectares. Both locations are mainly grassy and flat with no major hills. The park has a concrete-paved walkway all around. Each session is 30-60 minutes long. Data collection can be divided into 3 phases, pre-session, session filming and post-session, as shown in Figure 4.5.

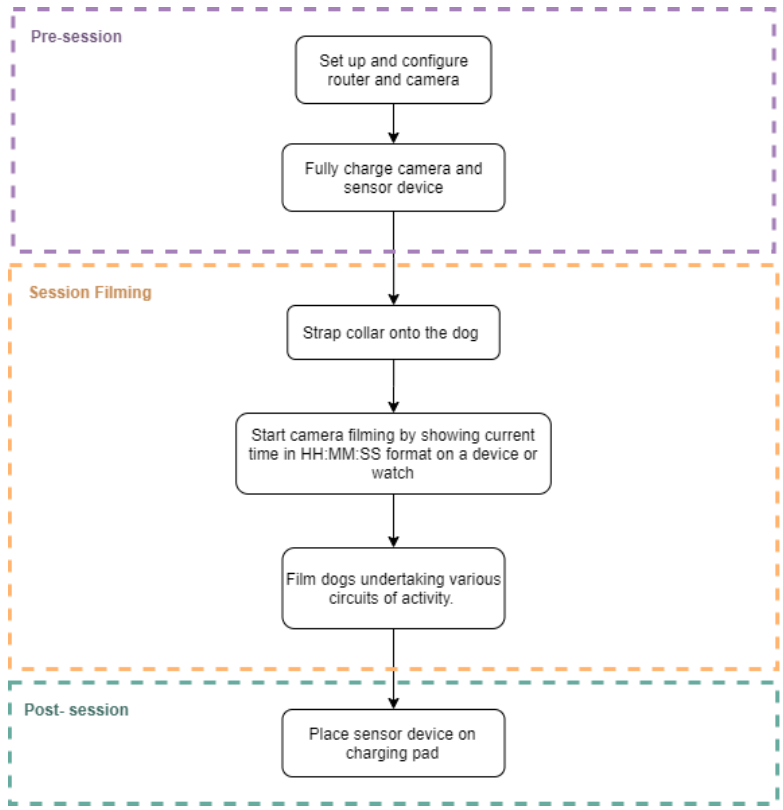


Figure 4.5: Flow Chart of Data Collection Process

Pre-session

1. Configure WiFi SSID and password of router specified by Mafic.
2. Camera settings is set to 720p on 30FPS.
3. Ensure that the device is switched on and fully charged.
4. Ensure that camera is fully charged.

Each sensor device provided by Mafic is programmed to a certain set of SSIDs and passwords. The camera is configured such that recordings are low resolution, but clear enough for humans to recognize dog activities, allowing longer video footage.

Session Filming

1. Put the collar with the sensor device attached around the neck of the dog. Ensure that the sensor device is always located at the base of its neck.

2. Start filming by showing the current time in HH:MM:SS format from a smartphone or a smartwatch as shown in Figure 4.6. Angle the camera such that the dog is in the frame and close enough that activities carried out are recognizable.

3. The dog undertakes the selected circuits of activities in Table 4.2.

Post-session

1. Charge the sensor device.

2. A member of the team requests the data for a session from Mafic by giving the MAC address of sensor device and date of session carried out.

When the sensor device is on charge and is connected to the WiFi SSID, sensor readings are sent over to a cloud server owned by Mafic automatically.

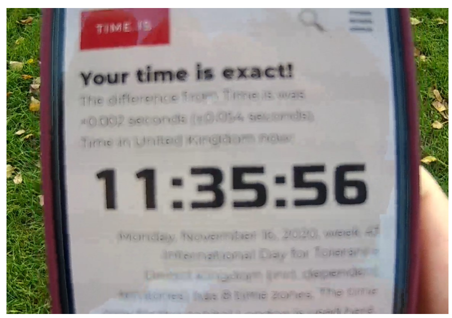


Figure 4.6: Time-synchronisation of video feed on a smartphone in HH:MM:SS format.

4.3.1 Data Collection Challenges

Some activities defined were challenging to obtain. Grooming, sleeping, and hunting activities are of which. Grooming is a spontaneous action which cannot be commanded. "Dog self-grooms for several reasons such as cleaning its wound, removing ectoparasites, removing loose hair when shedding and relief of stress" [4]. Selected subjects are maintained regularly, therefore self-grooming would not be seen as often. Due to the spontaneous nature of this activity, such data may be obtained if monitored for an extended period. However, there are limitations on equipment used, for long hours monitoring to not be possible. The action camera used has only 30 minutes of battery life when recording continuously. Sleeping data also requires long hours of monitoring since dogs do not normally stay asleep throughout the night. They are highly sensitive to noise and light. Pet dogs are often within the premise of their homes if not going out for a walk with their owners. This results in an invasion of privacy when trying to obtain data for sleeping activity with pet dogs, therefore in our study, only sleeping data with trained security dogs were obtained. Figure 4.7 shows the filming set up for collecting sleeping data. A camera is set up in a dog kennel and is filmed until the battery exhausts.

African Wild Dogs are considered predators as they hunt for food, while domestic dogs, especially pet dogs do not. Although modern dogs have innate predatory aggression, that does not mean that all dogs can hunt by instinct. For example, a house pet dog does not need to hunt for food and, for that reason, it may not have the skill set to hunt. Canine hunting behaviour refers to dogs detecting and capturing prey [29]. To resemble canine hunting behaviour, circuits 11 and 12 are designed. These circuits are only executed by trained security dogs. In comparison, the hunting strategy of African Wild Dogs differs slightly. "They exhibit pack behaviour whereby they hunt by approaching their prey in silence then chase for an average of 2km in distance, up to a speed of 66km/h" [23]. African Wild Dogs have extraordinary stamina which benefits them in a sprint chase. A notable difference between both species in hunting behaviour is that with domestic dogs, sniffing then biting action can be seen in sequence meanwhile with African Wild Dogs, trotting, galloping then biting action in series can be observed.

Figure 4.7: Filming set up for obtaining sleeping data.

4.4 Comparison with Specification

Specification 2.1

Section 2.1 of the project specification entails the creation of activity circuits to obtain data on activities: walking, trotting, galloping, grooming, barking, eating and sleeping. The team have obtained not only these specified activities, but other activities such as sniffing, playing, jumping, sitting, laying and standing too were also gathered. Circuits were designed to achieve this goal as shown in Table 4.2. Additional activities aid in analyzing behaviours exhibited by the subjects prior to the execution of an activity. As a stretch objective, the team were tasked to acquire data on hunting, which were met by designing two circuits of activity that demonstrate resemblance to hunting behaviours of African Wild Dogs.

Specification 2.2

Section 2.2 of project specification calls for the execution of activity with a sensor device. Camera used had a video resolution of 1080p though was set to 720p for the purpose of efficient storage. All sessions were first time-synchronised by showing the current time in HH:MM:SS format on a smartphone or a smartwatch before any designed circuits were carried out. All activity sessions are straightforward and feasible. Hunting activity, in particular, has certain requirements for data of such to be reproducible as detailed in Subsection 4.3.1. Device traces are sent to an AWS server hosted by Mafic as long as the device is on charge and connected to the router, which is done by a team member after each session. In conclusion, all goals and stretch goals are met by the team for Phase 2 of the project.

Chapter 5

Data Labelling

In this third phase of the project, the aim was to produce a labelled dataset. This involved combining the data collected using Mafic's IMU device with the video recorded in our data collection sessions. By recording video of each session, and syncing up the timestamp of the video with a timestamp in the data from the device, each different "activity" carried out by the dog could be given a label. These labels would be essential in generating a supervised Machine Learning model, as well as evaluating unsupervised techniques.

5.1 Process Overview

The labelling process made use of the "GlassGUI" proprietary software tool provided by our partner company, Mafic Ltd. An overview of the labelling process is illustrated in Figure 5.1.

For an individual session, a .pkl file for that day of device data was loaded into GlassGUI, along with all the video files recorded for that session.

The video files was then synced up with the timestamps in the trace. This was carried out for each video file.

The labeller must then go through each video, labelling periods of time where an activity is happening, for every activity seen in video.

Finally a CSV file of labels for the session is exported from the tool to be used in the next phase of the project.

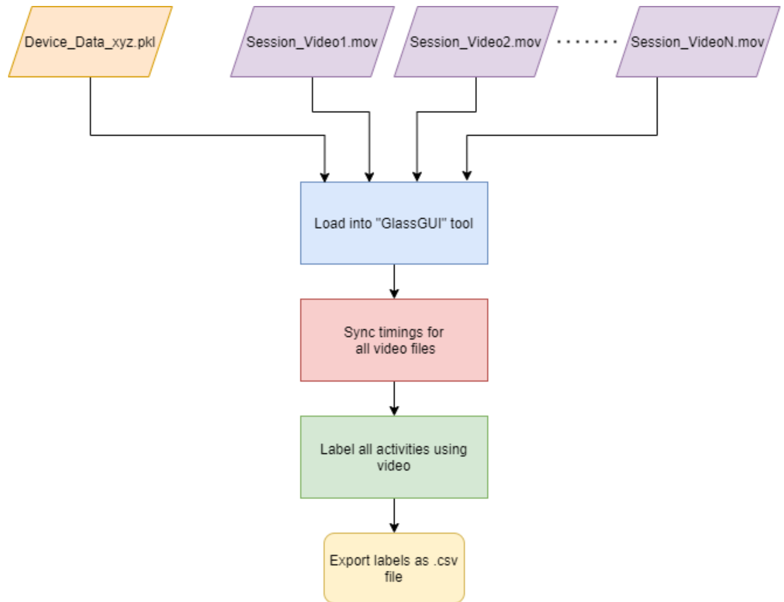


Figure 5.1: Flow Chart of Labelling Process

5.2 Detailed Process

5.2.1 Data Retrieval

Before the data itself can be labelled, it must first be retrieved. In regards to the data collected using Mafic's IMU Device, once a session is finished the data will be uploaded to Mafic's AWS server. From there, a member of the team would email a member of staff at Mafic, with the device MAC address and the date of the session, and they would send over two files.

Both files are .pkl files (from the Python Pickle module, a serialisation method [2]), one being the full data from the session, and one only including the timestamp, and the 3-axis acceleration data. This second file is used for labelling. As this second file contains less data points these can be easily plotted as a trace in the tool and be used to aid labelling. Once the files are retrieved they are stored in the Team's Gitlab repository so they are accessible to everyone.

In terms of video data, these were retrieved from the SD card of the action camera used, labelled with the date and start time in the filename, and uploaded to the team OneDrive (due to the large file size).

5.2.2 Syncing Video

Once the relevant files had been collected for a session, these were all loaded into the GlassGUI tool. From there, each video file had to be synchronised individually with a point in device data, or 'trace', which typically covers a larger time period than the session.

To sync a video file, a timestamp shown in the video was used, usually consisting of recording a phone with an exact time at the start of each video. By taking this frame of the video, and selecting the point in the trace with the matching timestamp, they could be synced. Once the video is synced to this point, it is moved in the timeline to the specified point. As a second check, once the video is synced, the trace can be used to ensure that the video is synced correctly. By scrolling to an obvious activity such as a jump on video, the trace can be analysed for a spike in movement to verify the sync. This syncing process can be seen in Figure 5.2

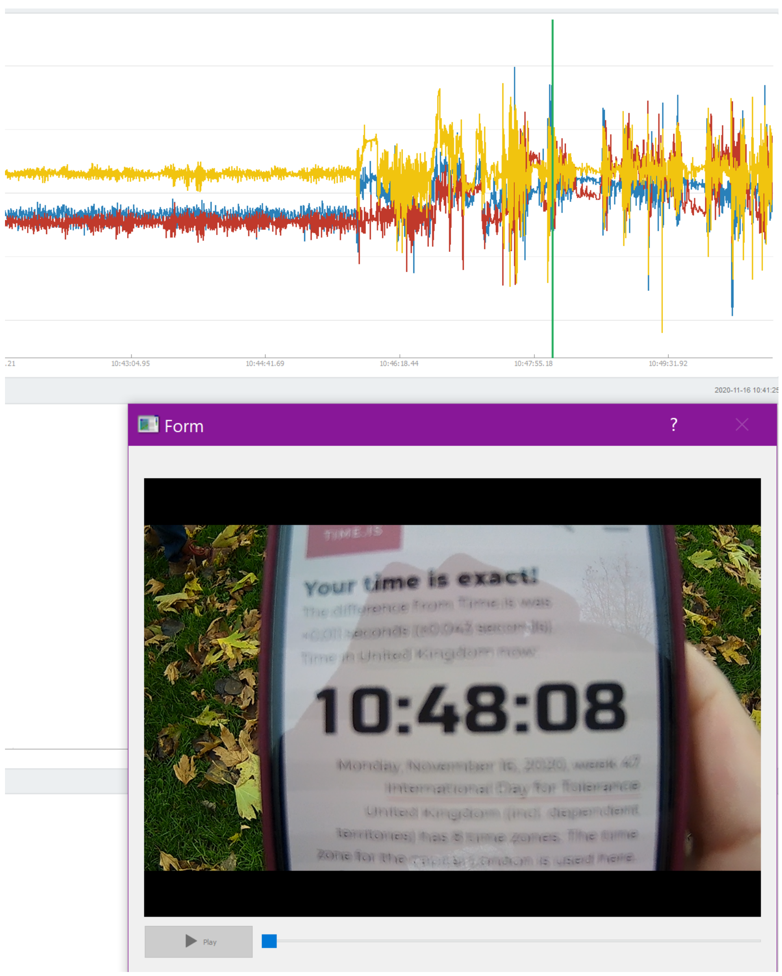


Figure 5.2: Syncing Video With Trace

5.2.3 Labelling

Once these setup stages were complete, the session needs to be labelled. In order to reduce subjective variation in labelling, a minimum number of labellers were used, two. This was as each person may define a certain activity, or where one starts and ends, differently, and thus keeping the number of labellers low gives comparable labels. The decided activities to label are laid out in the schema in Table 4.2 below.

ACTIVITY	**DEFINITION AND CHARACTERISTICS**
Walking	All four legs make contact with the ground independently of one another, resulting in a relatively slow forward movement overall.
Trotting	Often observed as a slightly faster walk, albeit with less vertical movement of the abdomen. Diagonal legs make contact with the ground at the same time, resulting in forward motion overall.
Galloping	A much faster motion than walking or trotting, the front leg pair make contact with ground nearly simultaneously, followed by the rear leg pair, resulting in forward motion overall. May result in all feet above ground at one time, and so excludes jumping. (Often associated with chasing a thrown/launched item.)
Eating	With some food object or liquid in the dog's mouth the process of chewing and/or swallowing, with characteristic movement of the jaw and neck.
Standing	All four paws remain on the ground, with all hocks raised off ground for >1 second, with no overall motion.
Sniffing	Dog's nose is angled downwards and close to the ground whilst standing for >1 second. This action can be seen with dog moving or stationary.
Laying	A dog's hind and front legs (hock and pastern included) are in constant contact with the ground for >1 second. The activity may also include resting on the dog's side and back. Excludes Sleeping.
Sitting	Hind feet and upper thighs are in constant contact with the ground for >1 second, with front legs straight or close to straight, resulting in a diagonal back position.
Jumping	All four paws are off the ground at the same time as a result of the dog's propulsion upwards, resulting in a notable vertical motion, potentially with a small degree of forward motion.
Barking	A barking noise is produced, and the mouth opens to show this. The audio attached to the video is essential in identifying this.

Table 5.1: Activites Schema

The labelling process then consisted of playing through all video clips for the session in the GlassGUI tool, marking the start and end point of each activity identified, and entering a label for it. To do this precisely, a mix of scrolling through the video clip slowly, and looking at the accelerometer trace was used. Looking at the accelerometer trace could speed this process up greatly, as certain activities were easily recognisable here and could be identified quickly and more accurately.

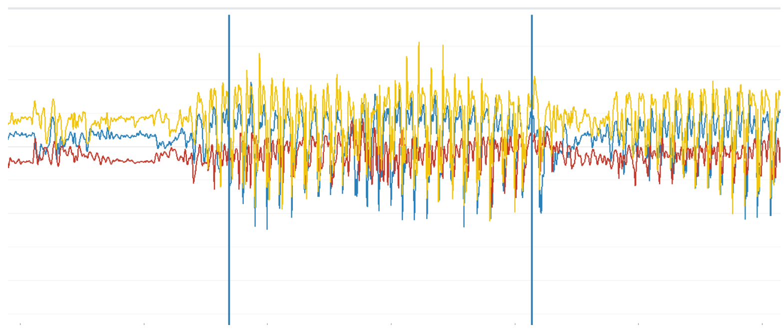


Figure 5.3: Trace of ‘Galloping’ with start and end bars

As shown in Figure 5.3, the ‘galloping’ activity could be easily recognised from the trace by the large, high frequency peaks seen here. The blue bars denote the start and end points of the activity. These start and end points were often placed slightly after the activity started, and slightly before it ended to prevent overlap between labels. Overlap between labels would mean that some data point would likely be mislabelled, and having overlapping labels would cause issues when adding these labels to the data at a later stage. These intermediate stages between activities were also less characteristic of the activity, for example when the dog is starting to run, and slowing down.

Once every activity seen in the video of the session has been labelled, the labels can be exported from the GlassGUI tool. The format of the exported labels is a .csv file, with each label name, and the start and end timestamp. Each activity has a new row. An example of the CSV format can be seen in Figure 5.4. This CSV containing the labels for this session is then stored in our Gitlab repository ready for processing in the next stage.

	A	B	C
1	start_time	end_time	activity
2	2020-11-05 14:10:31.999742	2020-11-05 14:10:32.880472	jumping
3	2020-11-05 14:10:33.001560	2020-11-05 14:10:34.082089	jumping
4	2020-11-05 14:11:06.622403	2020-11-05 14:11:12.468232	galloping
5	2020-11-05 14:11:23.355164	2020-11-05 14:11:25.097815	sitting
6	2020-11-05 14:11:25.771735	2020-11-05 14:11:26.317840	jumping
7	2020-11-05 14:11:28.267151	2020-11-05 14:11:29.516435	sitting
8	2020-11-05 14:11:29.522819	2020-11-05 14:11:30.334919	barking
9	2020-11-05 14:11:41.424262	2020-11-05 14:11:42.233073	jumping
10	2020-11-05 14:11:44.945894	2020-11-05 14:12:03.315025	galloping
11	2020-11-05 14:12:15.652499	2020-11-05 14:12:17.731086	sitting
12	2020-11-05 14:12:34.152698	2020-11-05 14:12:36.496045	galloping
13	2020-11-05 14:12:36.496045	2020-11-05 14:12:36.982319	jumping

Figure 5.4: Labels CSV Format

5.3 Comparison with Specification

The work detailed in this chapter can be directly compared to the specification written at the onset of the project. This can be used as a measure of how successful the phase has been.

Specification 3.1

Part 3.1 of the specification states that video files should be down-sampled when collected so that they may be stored more efficiently. When it came to the labelling stage of the project it was decided this step was not necessary, as they had been specifically recorded at the lowest resolution supported by the camera to keep file sizes small. The file sizes of the video files were then very manageable, and did not cause a bottleneck when loading into the tool. Therefore this specification point was not met, but on revision it would be removed, or changed to "record at 720p resolution to reduce file size" or similar.

Specification 3.2

Part 3.2 of the specification contains a number of points, many of which were met. Throughout the labelling process Mafic's GlassGUI tool was used, along with the corresponding collected data to label the traces. Not all of the video collected was used however, as a few issues were encountered with the device not collecting the data as expected. Because of this a small number of sessions have video, but no trace data to label.

In terms of activities labelled, the majority of those specified were labelled. However our goal to label "sleeping" or "grooming" data was not met. This came down to problems in the data collection phase, as these two activities were very hard to encourage from the dogs. In addition to these specified activities, we labelled a few extra activities (standing, sitting, laying, eating), as these were prevalent in the video footage, and labelling them did not require much more effort. One of our stretch goals was to label the "hunting" activity. As this had not been simulated in our retrieved sessions, no data and video was available to be labelled.

Another goal was to use a minimum number of labellers, for the reasons laid out in 2.3.2. In the end, only 2 labellers were used, as it was decided to be too time consuming for one person, and so achieved this goal.

The last two specification points apply to the process. As the GlassGUI tool was used for all labelling, a csv file output was obtained for each session, and so completed this goal. As a stretch goal the team aimed to automate the retrieval of data from Mafic's AWS servers. This was not met, due to not collecting enough data for this to be worthwhile, and with the speed of obtaining it manually, time was better spent in other areas of the project.

Chapter 6

AI Model Generation

In the fourth and final stage of the project the group was tasked with generating an AI model with the ability to predict the stated UK large dog activity from the data collected and labelled in the previous sections. As stated in the project specification, a major objective for this project was to achieve an overall classification accuracy in excess of 80%. This would allow the customer to use our model as a proof of concept for the eventual task of predicting African Wild Dog activities.

Several supervised learning models were tested over numerous stages to assess both intra-session and inter-session accuracies. Intra-session accuracies were found to be extremely promising, reaching above 99% accuracy, but testing in an inter-session configuration, which is more representative of performance in deployment, classification accuracy decreased to approximately 65%. This accuracy would predictably increase with a larger number and variety of sessions on which to train these models, but an upper bound of accuracy was demonstrated with impressive results.

Initial statistical and density based unsupervised clustering was performed on the raw IMU dataset. The results from this gave an initial understanding into the structure of the dataset. Techniques such as K-means and other models that need to know the number of clusters were used, however produced similarly low accuracy. Due to this, composite features involving directional and statistical information were created to help extract the most meaning from the dataset.

A semi-supervised technique was proposed to implement a form of pseudo-labelling. This was due to the long and strenuous task of manually labelling data. If the model could accurately produce labels, it would be a great asset for Mafic. The initial semi-supervised learning produced an accuracy of around 95% if 50% of the data was used to train the model and the rest used to be labelled and then retrain a more accurate model.

Deep Learning approaches were explored, given the successes of recognition tasks in similar fields. If similar performance can be seen with dog activities recognition tasks, Mafic could apply these methods if a large set of labelled data is accumulated, creating a system with less efforts in feature engineering. However, upon further testing, only low performances were obtainable.

6.1 Dataset Analytics

The product of this project's data collection and data labelling phases, with all corrupted and irretrievable collection sessions removed, constitutes roughly 515,000 labelled data points (which translates to 2 hours 52 minutes of session data). This data contained 6 features which are produced by default by the IMU device. These are: acc_x, acc_y, acc_z, gyro_x, gyro_y & gyro_z. These variables correspond to linear acceleration along three axes (including gravity), in addition to rotational acceleration about the same three axes.

A brief description of the distributions of these values over one session can be seen in Figure 6.1.

	acc_x	acc_y	acc_z	gyro_x	gyro_y	gyro_z
count	51603.000000	51603.000000	51603.000000	51603.000000	51603.000000	5.160300e+04
mean	-593.534077	-282.372672	-607.041606	-1613.547274	-4044.129508	-5.850420e+03
std	924.074716	738.856394	893.257968	133615.227727	106539.700871	1.288504e+05
min	-7994.000000	-7860.000000	-7989.000000	-982590.000000	-714070.000000	-9.460500e+05
25%	-883.000000	-656.000000	-901.000000	-62930.000000	-56280.000000	-5.950000e+04
50%	-499.000000	-339.000000	-525.000000	-70.000000	-3640.000000	-3.360000e+03
75%	-117.000000	67.000000	-132.000000	58170.000000	44380.000000	4.704000e+04
max	4713.000000	7825.000000	4820.000000	912170.000000	814030.000000	1.127840e+06

Figure 6.1: Distribution of Raw 6-axis IMU Values

From this analysis of the basic values, it is clear the scales of linear and rotational acceleration differ by an order of magnitude at minimum. In fact, taking the minimum or maximum values for these creates over two orders of magnitude difference. And so, in the case of any scale dependent classifier models being implemented over this data, the team was careful to scale all variables so as to ensure a mean of zero and unit standard deviation. However, it is important to observe the inter-session value discrepancy (as can be seen using Appendix G/E/'Basic Value Data Distribution') in order to appreciate how dramatically one session may differ from another. For classifiers that were scale independent, the team maintained the original values where possible, so as to keep these identified differences. This is such that sessions with unequal distributions of certain activities are treated completely independent of one another, maintaining comparability.

In order to augment this original data set, multiple secondary features were generated on which to train and evaluate the various machine learning models.

6.2 Feature Generation

6.2.1 Statistical Features

From the six raw features produced from the 6-axis IMU (XYZ accelerometer/-gyrometer features), there was a large amount of fluctuation and high frequency perturbations that may disrupt any attempt in using them for classification. As a result, it is necessary to process these 6 features so as to smooth the values over a small time window, as well as gather data regarding how sizable this variation is within this short duration.

In order to obtain the smoothed feature values, pre-processing code applies a windowing function over the raw features, using a window size of approximately 400ms, and calculating the mean value over that window. This small window size allows for a suitable degree of smoothing, whilst also maintaining granularity of the underlying data. Furthermore, if higher granularity is needed for the machine learning model, the existing features are kept to be trained on.

To measure an amount of variation over a short duration, the variance measurement was used. Again, this was calculated over the same 400ms window size for the entire data set. This measurement was necessary as it is the only feature that will take into consideration the magnitude of high frequency perturbations. These perturbations may be indicative of high-energy activities such as trotting, galloping and hunting. It is clear from the results shown in Section 7 that windowed variance exceeds many other more complex measurements in identifying such activities, as expected.

6.2.2 Bi-dependent Features

Another set of features calculated for each data-set before labelling or training is that of bi-dependent features. That is, features that are calculated from two similar features and which may take the moniker of 'directional' features in this report as a result of their correlation to directionality of a collection of features. These bidirectional features are calculated in two sets of three, being accelerometer and gyrometer data. For each of these XYZ feature triplets, the values of X/Y, X/Z & Y/Z are calculated. This calculation is performed over the raw, unaveraged values, so as to maintain granularity of features.

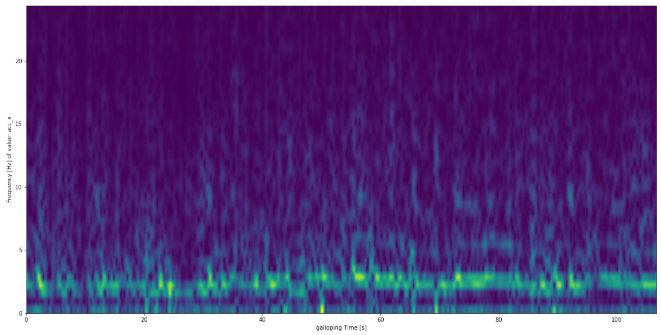
The bi-dependent features calculated this way have a strong tendency to scale disproportionately, varying rapidly between many orders of magnitude. This issue is inexorably caused by some raw values being very close to zero and inflating values to the point of being unusable for classification. Due to this issue, the decision was taken to perform a subsequent logarithm operation on the results to guarantee a more usable scale for machine learning.

These features were calculated and used in the machine learning training through adherence to existing literature. [17] identifies one of these features as being the single most important in their dataset in a set of 126 features. For the sake of testing, these features required analysis over various machine learning techniques and their importance validated.

6.2.3 Fourier Transform Features

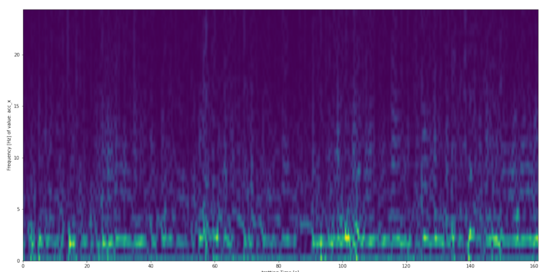
Beyond the basic features of windowed mean and variance, with the composite, directional variables, significantly more information can be extracted from the session data of the 6-axis IMU with bi-dependent features. One demonstration of this is any potential time-frequency data that may be governing the raw values observable in the data. One form of analysis to obtain such metrics is the Fast Fourier Transform (or FFT) analysis which, for any set of given data and a chosen feature, will create a set of values pertaining to the strength of a sinusoidal signal of a range of frequencies.

The Fast Fourier Transform algorithm was applied to the data-set in windows with duration of 1.6 seconds for each indexed data point. This allowed the algorithm to calculate the aforementioned frequencies' strengths for every time present in the data, whilst ensuring granularity and precision within that region. Without this windowing functionality, the FFT algorithm would calculate the strength of a given signal over the entire data set, which encapsulates many different activities and would likely produce an averaged signal strength, which would be unusable to a machine learning algorithm. The value of 1.6 seconds was chosen through visual inspection of the produced data to maximise the difference between values produced over various frequencies. A complete set of Fourier value plots is present in the attached Appendix G/E/Heatmaps.

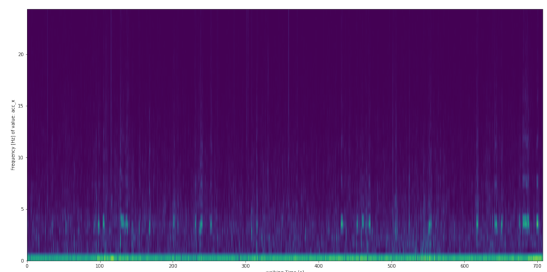


(a) Galloping

Figure 6.2: X-axis Accelerometer Fourier Analysis



(b) Trotting



(c) Sitting

Figure 6.2: X-axis Accelerometer Fourier Analysis

Figure 6.2 shows the heatmaps of the Fourier transform results produced by X-axis acceleration data over three different activities: Galloping, Trotting and Sitting. In these plots, green and yellow points are maxima, dark blue points are minima. Through inspection of these plots, it can be observed that there are prominent differences in the frequency of the maximum values between activities of varying activity levels. For example, from Figures 6.2b & 6.2c there is a significant variation of value for the signal corresponding to that of the third and fourth lowest frequencies, shown by the clear yellow band in the trotting data. Similarly, comparing Figures 6.2c with 6.2a, the lowest frequency strengths are much less pronounced in galloping data, as expected as galloping is an activity of a higher activity level compared to remaining seated.

The aforementioned frequencies produce a notable difference in strength between one set of activities and another. However, when activities are much more similar in activity level, such as galloping and trotting (which are arguably indistinguishable by most humans at a glance), the Fourier data is much less refined. There are observable differences, such as in the third lowest frequency shown in these plots but, given the variations in multiple instances of trotting or galloping, these may be somewhat unreliable as metrics taken out-of-context.

Due to the number of frequencies being analysed through Fast Fourier Transform in this case, it is infeasible to append the strength of every frequency to our set of features as this would produce roughly 360 extra features (40 frequencies analysed over 9 features). As such, it is required to select the desired frequencies from the values plotted in Figure 6.2. In the case of this project, where testing time was extremely limited, the quickest and most rudimentary method was used; manually performing analysis and 'cherry-picking' frequencies from heatmaps as in those shown above. Although this was a fast method that obtained important data to use in machine learning models, in retrospect, another method, such as instead taking the frequency of highest strength (or set of highest strength frequencies) would have produced a more flexible data collection. As it stands in the current model, the group is satisfied with the frequencies selected for the purposes of domestic dogs, but it is acknowledged these may need adjustment for African Wild Dogs.

The complete list of frequencies used from the analysed features' Fourier Transforms is present in Table 6.1. The frequency index refers to the number of the frequency increasing up the heatmaps shown previously.

Feature	**Frequency Index**
acc_x	0, 2, 3, 4, 5, 7
acc_y	0, 4, 5
acc_z	0, 2, 3, 4, 5
gyro_x	5
gyro_y	3, 4
acc_x/acc_y	0, 2, 3, 4
acc_x/acc_z	5
acc_y/acc_z	0, 2, 3

Table 6.1: Table of chosen frequencies from Fourier Transform analysis

6.2.4 Wavelet Transform Features

An additional source of data using time-frequency domain data is a method known as the Wavelet Transform. Wavelet Transforms differ from Fourier Transforms in its improved resolution when applied to very high and very low frequency signals in the analysed data [28]. Furthermore, although the Wavelet Transform translates the raw time-series results to the time-frequency domain, it uses a very different signal shape when compared against Fourier's sinusoidal signal. Although there are many variations of wavelets supported in Wavelet Transforms, the chosen wavelet function is known as the 'Morlet' wavelet [21]. The shape of such a wavelet is derived from the equation:

$$\psi(t) = \exp\left(-\beta^2 t^2/2\right)\cos\left(\pi t\right)$$

Figure 6.3 shows the shape of a typical Morlet wavelet being used for the wavelet transform analysis of the collected data.

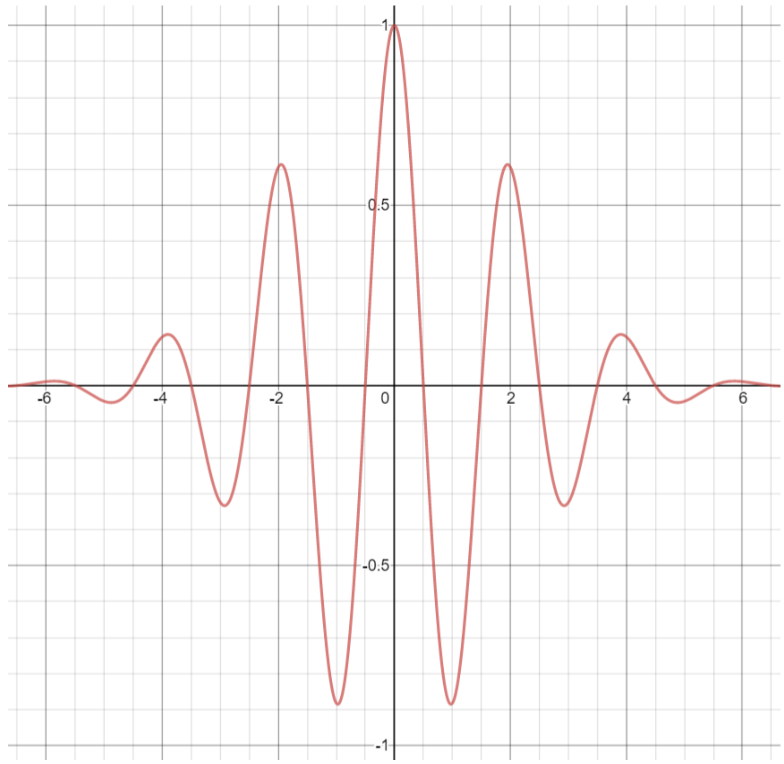
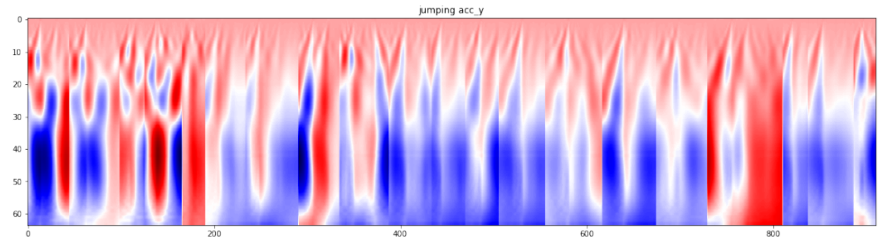
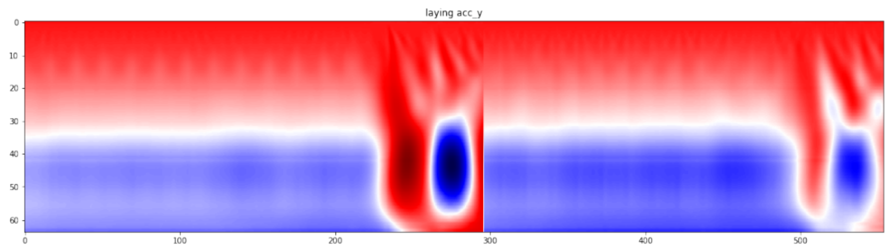


Figure 6.3: Morlet Wavelet with $\beta = 0.5$

As with the Fourier features, in order to maintain granularity and preserve performance, a window of duration 1.6 seconds was chosen and for each data point Wavelet Transform is performed over this window. the results from the wavelet transforms indicate the degree of 'spiking' observed in a variable on a specific data set. As such, one would expect to see much more varied and possibly extreme values at higher frequencies produced in activities such as galloping, jumping and trotting, whilst less active activities, such as laying and sitting, show more consistency.



(a) Jumping



(b) Laying

Figure 6.4: Y-axis Acceleration Wavelet Transform Analysis

Figure 6.4 shows the variation of values produced by the Wavelet Transform algorithm run over the Y-axis acceleration data in one of the conducted sessions, with red areas being maxima and blue showing minima. A complete set of wavelet transform value heatmaps is available in Appendix G/E/Heatmaps. Clearly there is a significant difference in the values obtained through laying (Figure 6.4b) as compared to jumping activities (Figure 6.4a). This is true of those lower frequencies that appear at the lower end of the heatmaps presented here. At low frequencies, the values produced by jumping data are much more varied, indicating that there is much more movement occurring than in the much more stationary activity of laying down. Similarly, at higher frequencies in the heatmaps, there is a much clearer detail resolution in jumping than in laying activities, demonstrating the shift of movement from low frequency to high frequency, as one would expect of these activities.

As with the Fourier Transform features, the number of features produced if all results were used would be extremely high, at roughly 384 (64 frequencies over the 6 IMU features). Adding all features would not only introduce a large quantity of irrelevant data to our processed data, leading to much longer training and labelling times, but may also lead to an effective decrease in accuracy. Despite efforts put in place through employing L1 regularisation techniques, a large number of features may lead to overfitting on misleading data. As in the Fourier Transform features, the team has manually chosen frequencies that have distinguishable differences when comparing one set of activities to another. As before, in future work, we suggest a more flexible method of representing this data is investigated. This step of optimisation is regrettably outside the scope and time-setting of this project. The complete set of frequencies chosen appear in Table 6.2. The frequency index refers to the value that appears on the y-axis with that data in the relevant heatmap.

Feature	**Frequency Index**
acc_x	14, 43, 62
acc_y	48, 15
acc_z	45
gyro_x	12, 45, 62
gyro_y	18
gyro_z	48

Table 6.2: Table of chosen frequencies from Wavelet Transform analysis

6.3 Supervised Learning

As the data collection and data labelling phase of this project had produced a number of fully labelled sessions with domestic dogs, primarily supervised learning techniques were evaluated for use in the final deliverable. It was speculated that, given the disparity between the data sets of numerous sessions, supervised learning would result in stronger performance than purely unsupervised machine learning models, particularly as the labels were known before training had begun (a scenario in which unsupervised models would prove very useful). As such, an environment for training, testing and optimising various techniques was created, with the techniques set out in Table 6.3 evaluated and compared. Observations on each classifier can be found in Chapter 7.

6.3.1 Evaluation Environment

The data obtained through the previous two project phases was observed to have activities somewhat unevenly divided between the sessions conducted. For example, one session would have as many as eight different activities clearly identified, whereas, due to session constraints and dog behaviour, another may only exhibit four activities. As such, it was productive to ensure models were evaluated using each session as both training and test data, to obtain a representative accuracy. Furthermore, due to expected randomness, sessions typically had differing distributions of activities when compared to others. To minimise bias when training models, therefore, it was decided to oversample on the data points, creating a training set with equal proportions of each activity. Finally, as a result of the Covid-19 pandemic or restrictive time limitations, the data collection phase was unable to obtain a sufficient amount of data for the following four activities: Barking, Jumping, Playing and Sleeping. For this reason, data points labelled as any of these activities were removed from any training and testing sets, so as to mitigate potential overfitting brought on by oversampling.

Raw Accuracy

Initially, to determine a first-pass, maximum bound of classification accuracy of the techniques to be evaluated, each classifier was trained on a subset of the total data (all labelled sessions) with all features enabled (As set out in Section 6.2). All data points not present in the training subset were then used for the test set. Having assessed each model available using this configuration, models that were performing very poorly were removed from the set of those being evaluated.

Generalisation

Having removed poorly performing classifiers, the set of remaining classifiers were then tested for generalisation performance. This assessment was using a complete session as a test data set, with the remaining sessions' data used for training purposes, with all features enabled as before. It is acknowledged that the performance on one subset of data is not necessarily representative of the performance on all sessions. However, for the purposes of identifying those models that performed especially well when generalising, this single session technique was effective and time-efficient. As in the previous stage, classifiers that did not generalise well were removed from the set of assessed classifiers.

Optimisation

With a small subset of the evaluated classifiers remaining, the Python library, scikit-learn was employed to optimise remaining classifiers for unseen session accuracy [26]. This included building a custom scoring class specific to the canine IMU data use case in order to separate the test data sessions from the training data. In this way, it was possible to optimise the classifiers for session-wise cross validation score, with all sessions being used as test data once. To optimise using the scikit-learn API, a grid search was performed over model hyperparameters including L1 and L2 regularisation parameters, number of training iterations and number of estimators (for ensemble classifiers). Once optimisation was complete, each classifier was run against all sessions' data, using one session as a test set on each iteration. A mean percentage classification accuracy was calculated with standard error over the sessions, producing comparable results between models.

Scalability

To evaluate the performance of a model on a metric aside from classification accuracy, this project assesses the potential for scalability of each model that achieved sufficient accuracy after optimisation. For the purposes of this project, this metric is assessed by limiting the features passed to the model for training to only those most important to the model. The models are then tested once again for session-wise accuracy over the complete dataset and their mean accuracy calculated with standard error. This evaluation phase allows the possibility of reducing the time necessary for pre-processing of data when the complete system is deployed and models must be accurate having been trained for a short time on low-power laptops (as per specification section 4.2).

6.3.2 Classifiers

A complete list of classifiers is shown in Table 6.3. Also present is the family of machine learning techniques each belongs to, with the library or Python API through which implementation was provided.

Classifier Name	Model Family	Library Used
XGBClassifier	Ensemble (Gradient Boosting)	XGBoost [8]
LGBMClassifier	Ensemble (Gradient Boosting)	LightGBM [19]
HistGradientBoostingClassifier	Ensemble (Gradient Boosting)	scikit-learn [26]
ExtraTreesClassifier	Ensemble (Forest)	scikit-learn [26]
RandomForestClassifier	Ensemble (Forest)	scikit-learn [26]
MLPClassifier	Multi-layer Perceptron	scikit-learn [26]
KNeighboursClassifier	Nearest Neighbour	scikit-learn [26]

Table 6.3: Classifiers tested during supervised learning evaluation

The number of classifiers tested was governed by time limitations and a complete analysis of fewer techniques was preferred over an incomplete analysis of additional techniques. Furthermore, unsupervised techniques are also covered in this report and their performances evaluated.

With the framework detailed above and a comprehensive testing harness for obtaining performance and feature importance metrics, all the above classifiers were tested. The classifiers to remove from further testing stages were determined by comparing performances of each model with its peers.

6.4 Unsupervised Clustering

While supervised learning was likely to be very effective in this use case, it was decided to also investigate unsupervised clustering, and whether it could be used to pick different patterns straight out of the data. This involved looking at outlier data points; hierarchical structures and potential unclassified activities. If this could be done to a high level of accuracy, it would be possible to use this to classify dog behaviour, without the requirement to label data. This was an experimental section of the project, where the results from this did not affect the final deliverable product, but experimented with techniques that the customer could further investigate for a more robust system.

6.4.1 Density Based Clustering

Motivation

The first approach for unsupervised clustering was to consider density based methods. Density based methods are ways of clustering that identify distinctive clusters in the data, based on the idea that a cluster in a data space is a contiguous region of high point density, separated from other such clusters by contiguous regions of low point density [27]. From these regions of high and low density clusters, the theory was that this clustering could give an insight into the hierarchical structure; the outlier data and new potential dog activities within the dataset.

Investigation

The main aim of this investigation was to find out if there were any interesting patterns hidden within the raw dataset. This would allow for a deeper understanding of the data the group was working with, whilst finding possible dog activity clusters. To understand the dataset in its entirety, the clustering was performed on both the full combined data set and the data from each individual session. Comparisons between the clusters and structures found were made between them, identifying the key similarities and differences.

The first stage was to identify suitable clustering methods that relied purely on the density of data. These methods are shown in Table 6.4. Mean shift clustering aims to discover blobs in a smooth density of samples. DBSCAN views clusters as areas of high density separated by areas of low density. OPTICS algorithm shares many similarities with the DBSCAN algorithm, and can be considered a generalization of DBSCAN that relaxes certain parameters. Affinity Propagation creates clusters by sending messages between pairs of samples until convergence. The thought was that having a variety of density based clustering algorithms would allow more hidden patterns and structures to be found.

Clustering Type	Use Case	Library Used
Mean Shift	Many clusters, uneven cluster size, non-flat geometry	scikit-learn
DBSCAN	Non-flat geometry, uneven cluster sizes	scikit-learn
OPTICS	Non-flat geometry, uneven cluster sizes, variable cluster density	scikit-learn
Affinity Propagation	Many clusters, uneven cluster size, non-flat geometry	scikit-learn

Table 6.4: Unsupervised Clustering Techniques Applied

Each of the clustering methods described was applied to the raw dataset both in its entirety and individual sessions. Each method from Table 6.4 required different input parameters, that need to be derived from some prior understanding of the dataset. This ranged from the dataset's bandwidth, minimum proximity to neighbouring data points and a variety of other features. From this any cluster's, outlier points and any hierarchical structures were identified for each session and as a whole.

Dimensional reduction techniques such as PCA was used for both data visualisation, and the removal of redundant features. Comparisons were made between each sessions to see if there was any underlying commonality between the structures found in the datasets. The final stage was to see if the patterns and structures found presented any useful insight into the classification of UK large dog activities.

6.4.2 K Clusters Based Clustering

Motivation

Another approach considered for Unsupervised Clustering, was K Clusters methods. As the number of unique activities in the dataset was known, classification techniques for a known number of clusters could be used.

When labelling the data, many of the activities seen had very distinct traces and could be identified accurately by eye. Due to this there was a confidence than these activities were distinct enough to be classified using these techniques. The K Means method appeared to be applicable to datasets with such a structure, as described in the Application section of [11].

Investigation

One of the aims of this investigation was to see whether any patterns found were intrinsic to the dog used, or could be generalised between different dogs and different sessions. To enable this, each session was converted into a list of each session's Data Frame, with a final Data Frame containing every session concatenated together (a combined session). These Data Frames were then converted to arrays of just the current features, as required as an input to the K Means Classifier. Setting up the data this way meant that the K Means algorithm could be applied to each session independently, as well as to all sessions combined, and the results compared.

For the initial investigation, the K means algorithm was applied to just the raw 6-axis IMU data, with no added features. Each array in the session list was then fed as the input to a K Means classifier from the scikit-learn cluster module. Each classifier was initialised with 10 clusters, corresponding to the 10 activities present in the data from the Data Labelling phase.

The K Means algorithm was also used in this same way for the dataset with a varying number of features, to investigate how this would effect clustering. The K Means algorithm was run on the original 6 Axis IMU data, as well as with Mean/Variance, Reciprocals, and Fourier features added to the dataset, as described in Section 6.2.

6.5 Semi Supervised Pseudo Labelling

6.5.1 Motivation

While labelling the collected data, it became clear how much time it requires to label even a small amount of data. As more data is collected, the accuracy of our models can be greatly increased, but only if this manual labelling is carried out.

This led to the consideration of Semi-Supervised techniques to"Pseudo label" data. This process uses available unlabelled data, to gain more of an understanding of the data population, and from this improve the accuracy of a model. By training a model on a small amount of labelled data, as long as this initial model can classify more data with a good accuracy, it can be used for this process. This initial model is then used to predict the classes of another set of unlabelled data, and as these labels are likely quite accurate, this data is concatenated onto the labelled dataset and used to train the model again. This should have the effect or reducing errors in new predictions by this model as it has a better understanding of the structure of the data, simply as it has seen more of it.

6.5.2 Configuration

To use this technique, the data was processed to include Mean/Variance, Reciprocal, and Fourier features. This was in order to give the classifier an acceptable accuracy on its first train to be able to carry out pseudo labelling. The data was then split into 3 datasets, initial training, pseudo labelling, and a test set. For pseudo labelling, it is recommenced that between $\frac{1}{4}$ and $\frac{1}{3}$ of the full dataset is Pseudo Labelled [10]. Therefore out of the 5 sessions investigated here, 4 were used to train the initial model. The final session was then separated by a train test split, with 0.25 in the test set. This gives us 0.75 sessions to be pseudo labelled. This number was used to be conservative, as different sessions have different numbers of datapoints. It is better to include a smaller amount of pseudo labelled data, than too much. It was important to pseudo label and test on unseen sessions, to prove this technique could be generalised for unseen sessions. The classifiers were the Ada Boost classifier, which has been tested with success on this problem, with the need to get high performance initially before the pseudo labelling stage.

As the pseudo labelling dataset did in fact have labels, these were used to test the initial accuracy of the model, comparing the predicted labels to the true labels. Then when the second classifier was tested on the test set, the true labels of the test set were used to measure its accuracy.

6.6 Deep Learning

Feature generation using Fourier Transforms and Wavelet Transforms are computationally expensive in terms of power and time. In comparison to supervised learning with handpicked features using Fourier and Wavelet transforms, features can be learnt automatically with Deep Learning. For this reason, Deep Learning (DL) approaches were explored by using raw time-series data from the sensor device as inputs. Approaches involve using Convolutional Neural Networks (CNN) and CNN+LSTM architectures which are popular among Human Activity Recognition (HAR) and Time Series Classification (TSC) tasks. If architectures examined could achieve higher overall accuracy on all dog activities, one may use deep learning approaches if GPU resources are available. This section is experimental and is not included in the final deliverable product, but experimented with techniques that the customer could further for a more robust system.

Motivation

Two areas of research were looked at when choosing techniques to approach the task. The first area of research explored is Human Activity Recognition (HAR). The thought behind this action is that the task looked upon is still an activity recognition task. The only thing that sets both apart is that one is on humans while the other is on animals. [13] compiled a survey on HAR using inertia, physiological and environmental sensors. This means that this paper surveyed ongoing and up-to-date trends for solving the task. The paper has reviewed 293 published papers along with 46 survey papers on traditional machine learning (CML) and deep learning (DL). DL architectures are particularly looked at for this section. CNN is the most popular architecture and has achieved an overall accuracy of 93.7% over 11 daily life activities. Long Short Term Memory (LSTM) and Recurrent Neural Networks (RNN) are placed second and third each in terms of popularity when approaching HAR tasks. LSTMs obtained 91.5% on an average of 17 activities while RNNs achieved 95% on an average of 14 daily activities. Figure 6.5 shows the model selection process for HAR. Given the circumstances and limitations on activity recognition for domestic dogs in our study, CML was deemed a good choice.

The other area of research explored was Time Series Classification (TSC). Motivation behind this approach is to explore how time-based data can be classified based on behaviours for different activities. [15] reviews DL methodologies for TSC tasks. The paper was published on May 2019 and has trained 8,730 DL models on 97 time-series datasets. Residual Network (ResNet) [18] and Fully Convolutional Neural Network (FCN) [30] showed the best results on most datasets trained and are ranked first and second respectively for the task. The paper especially addressed a characteristic regarding small training size. With a dataset of 16 training instances (DiatomSizeReduction from UCR Archive 2018 [12]), performance of ResNet and FCN dropped to 30%. In contrast, Time Convolutional Neural Network (Time-CNN) [32] achieved the best accuracy (95%) among all models tested. This model only outperformed tests for 4 out of 128 datasets as compared to ResNet which ranked first for 41 of 128 datasets and FCN with 18 [15].

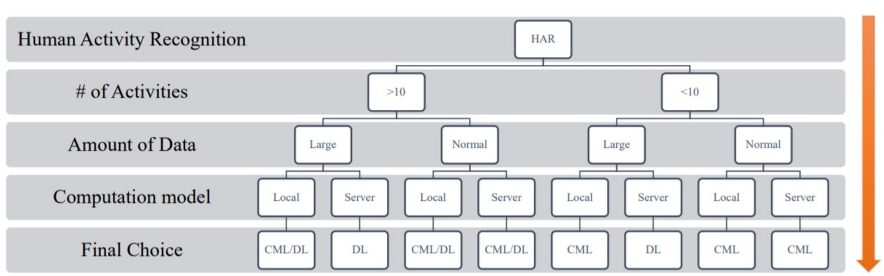


Figure 6.5: Model Selection Diagram for Human Activity Recognition [13].

Investigation

There are two main aims for this experimental approach, if raw time series data can be fed into DL architectures as inputs and if Neural Networks could identify better feature representations of data as compared to handpicked features. When labelling, some traces are easier distinguished from others. For example, 'Galloping' shows a higher frequency for all XYZ axis accelerometer readings from Figure **??** and 'Walking' shows lower frequency. If humans are able to recognize certain activities from traces after labelling data based on video feeds for several times, deep learning architectures may be able to recognize interesting patterns in IMU sensor readings.

First stage was to split data into train and test set by subject IDs. Different subjects exert different behaviours even with the same activities, therefore it is important that the model is able to recognize activities though sessions are executed by different subjects. Subjects that are pet dogs (4 subjects) are chosen as train data and professionally trained dogs (2 subjects) are chosen as test data. Validation data had a 0.2 split from train data. Motivation behind this configuration is to test on unseen data. Unseen data was not chosen for validation data due to each subject do not necessarily have all defined activities in their sessions, therefore allocating a subject for validation data could be challenging. For example, some subjects are rather calm, 'Barking' could not be obtained as a result or 'Sleeping' data is absent for pet dogs due to opposition to indoor filming by pet owners.

Regardless of data partition, there is a high class imbalance in data. To address this issue, Random Undersampler (RUS) and Random Oversampler (ROS) were used to mitigate this problem. RUS is first used to undersample the majority class to a desired value, then ROS is applied to oversample all classes to the amount of data in majority class. RUS is first used because if data are oversampled to majority class first, this creates bias in data, resulting in models training on data with bias, which is not desirable. Apart from RUS and ROS, data generator is also implemented. Reason for using data generators is such that for each epoch that the model is training on training data, it sees a balanced set of data and validates with one too. Data generator is applied to both training and validation sets. This ensures that 'Training Loss', 'Training Accuracy', 'Validation Loss' and 'Validation Accuracy' metrics could return correct values for multi-class classification task. This is important because when a model is being trained, useful callbacks like 'ReduceLRonPlateau' monitors a chosen metrics in which learning rate defined can be altered during the process while 'EarlyStopping' stops the training process if no improvements were seen on an defined metric. These metrics are calculated using overall results. An example is that if there is high class imbalance in a batch, and if the model is unable to correctly classify an activity, this lowers the overall accuracy altogether. With 'ReduceLRonPlateau' callback, if validation accuracy is chosen to monitor, learning rate of optimizer will decrease if there is no improvements in this metric. If these metrics gave wrong values during training, the model would not be able to learn correctly, leading to issues surrounding underfitting and overfitting.

Three DL architectures were explored, of which are CNN, DeepConvLSTM [25] and Time-CNN [32]. These architectures were chosen because CNNs are proven to have achieved remarkable results on HAR and TSC tasks. DeepConvLSTM achieved 95.8% overall accuracy on SKODA dataset [31] for gesture recognition on 10 car maintenance scenarios. Its architecture is shown in Figure 6.6. Time-CNN was chosen due to its outstanding performance on small training data. The architecture proposed is as shown in Figure 6.7. Table 6.5 and 6.6 shows architecture and optimization configurations for all three models. All models were experimented with different train test splits by subjects, RUS, ROS, or data generator for data pre-processing techniques, shown by Table 6.7. Implementations for model architecture used are from [22][16][14].

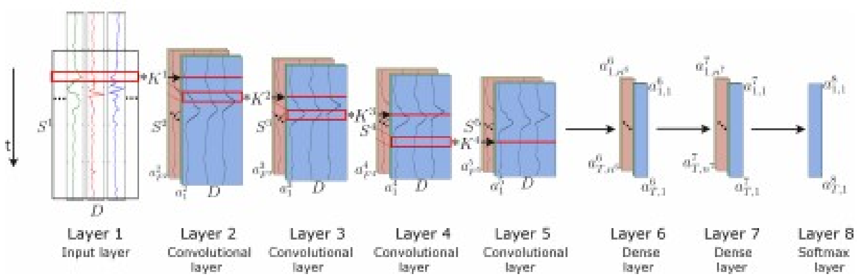


Figure 6.6: DeepConvLSTM Model Architecture [25].

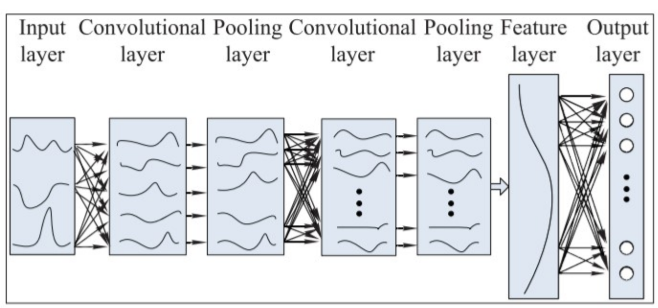


Figure 6.7: Time-CNN Model Architecture [32].

Methods	Architecture									
	#Layers	#Conv	#LSTM	#Invar	#Dropout	#Pooling	Dropout	Pooling	Activation	Regularization
CNN	8	4	0	4	2	2	0.3, 0.5	Avg, GAP	ReLU	L2 kernel
DeepConvLSTM	9	3	2	4	1	3	0.2	Max	ReLU	None
Time-CNN	3	2	0	2	0	2	None	Avg	Sigmoid	None

Table 6.5: Deep Learning architectures hyper-parameters.

Methods	Optimization					
	Algorithm	Valid	Loss	Epochs	Batch	Learning rate
CNN	Adam	20% split from train	Entropy	80	32	0.001
DeepConvLSTM	Adam	20% split from train	Entropy	80	32	0.001
Time-CNN	Adam	20% split from train	MSE	80	32	0.001

Table 6.6: Optimization hyper-parameters for DL approaches.

Test Scenarios		
Train/Test Split	**Preprocessing**	**DL Architecture**
4(P) // 2(S)	RUS+ROS	CNN
4(P) // 2(S)	RUS+ROS+Data Generator	CNN
4(3P+1S) // 2(1P+1S)	RUS+ROS	CNN
4(3P+1S) // 2(1P+1S)	RUS+ROS+Data Generator	CNN
4(P) // 2(S)	RUS+ROS	DeepConvLSTM
4(P) // 2(S)	RUS+ROS+Data Generator	DeepConvLSTM
4(3P+1S) // 2(1P+1S)	RUS+ROS	DeepConvLSTM
4(3P+1S) // 2(1P+1S)	RUS+ROS+Data Generator	DeepConvLSTM
4(P) // 2(S)	RUS+ROS	Time-CNN
4(P) // 2(S)	RUS+ROS+Data Generator	Time-CNN
4(3P+1S) // 2(1P+1S)	RUS+ROS	Time-CNN
4(3P+1S) // 2(1P+1S)	RUS+ROS+Data Generator	Time-CNN

Table 6.7: A summary of test scenarios. (P - pet dogs), (S - professionally trained security dogs)

6.7 Comparison with Specification

The work detailed in this chapter can be directly compared to the specification written at the onset of the project. This can be used as a measure of how successful the phase has been.

Specification 4.1

Section 4.1 of the project specification outlines the objectives and stretch objectives for the dataset formatting. The 50Hz data was labelled using the labelled interval csv file and all features not encompassed by the 6-axis IMU was removed. Standardisation and normalisation were applied to some techniques such as the unsupervised methods and the initial supervised techniques however it was decided not to set up a standard scaling for the final model to ensure it was as general as possible for a wide range of input data. The stretch object of creating a local or online database was not met due to delays with Covid and spending time on more important aspects of the project. The stretch objective of creating sub-sampled variants of labelled device data was in part met, as the model did sub-sample features with too much data, however separate datasets were not created as they were found to be redundant.

Specification 4.2

Section 4.2 of the project specification outlines the objectives and stretch objectives for the AI Model Architecture. An AI model was created with raw accuracy of 99% and a general accuracy of up to 72%. In the team's opinion this meant that the objective of an overall accuracy of at least 80% was met. The model was able to predict all of the activities stated in the objective and the model was programmed in Python as required. Training and execution of the AI model was capable of running on an everyday laptop due to the non GPU methods implemented. The stretch objective for predicting hunting was not met due to the lack of data collected from the UK large dogs, meaning an accurate prediction could not be produced.

Specification 4.3

Section 4.3 of the project specification outlines the objectives and stretch objectives for the AI Model Performance Evaluation. The AI model's classification accuracy was tested using K-fold cross-validation, confusion matrices and classification reports to ensure a robust model. To ensure the model was generalised, it was trained and tested on different dog sessions. The stretch objective to analyse the accuracy of lower sampled data was not met, as the team decided these models were unnecessary. Feature importance from the classification model was documented and compared with other models. The variance features were found to be the most important for classifying the dog activities.

Chapter 7

Results & Observations

7.1 Supervised Learning

7.1.1 Classification Accuracy

As described in Section 6.3, each supervised learning classifier model was tested for four accuracy measures: Raw accuracy, generalised accuracy, optimised accuracy and feature-reduced accuracy. The first three accuracies (if the classifier was not removed during the testing process) are shown in Table 7.1.

Classifier	**Raw Acc.**	**General Acc.**	**Optimised Acc.**
XGBClassifier	99.67%	69.71%	61.58% $\pm$ 3.94%
LGBMClassifier	93.65%	59.03%	
HistGradientBoostingClassifier	96.92%	69.01%	58.84% $\pm$ 3.14%
ExtraTreesClassifier	99.99%	63.00%	
RandomForestClassifier	99.95%	72.39%	64.09% $\pm$ 3.10%
MLPClassifier	31.23%		
KNeighboursClassifier	86.58%	42.70%	

Table 7.1: Classifier accuracy during supervised learning evaluation

To reiterate, the project group acknowledges the potential pitfalls in this ‘tourney’-style evaluation of models, but due to time restrictions, only a limited number of classifiers were able to be investigated completely.

Raw Accuracy

From the first tests conducted with all classifiers, it was evident that there was a clear distinction between the accuracies obtained by ensemble methods compared to other models, when trained and tested on the entire set of sessions. Although the stage of evaluation is pitting unoptimised models against each other, this represents an upper bound on accuracy using these techniques unaltered (as the training data will be from the same sessions as testing data). In addition, when optimised and regularised, this metric could represent potential accuracy when trained on a much larger and more representative data set, as would eventually be the case when the system is deployed.

The two classifiers that did not attain an accuracy above 90%, 'MLPClassifier' and 'KNeighboursClassifier', can be assumed to have lower accuracies for different reasons. The multi-layer perceptron model obtained accuracy of roughly 30% in initial testing and these models are often very difficult to optimise, given the increased flexibility of the classifiers and the exponential search space for hidden layer sizes. Given the library used, these parameters were tuned over a feasible amount of time to optimise for accuracy, but the accuracy herein is that result, which is markedly lower than that of all other models. The K-Nearest Neighbour classifier, however, has the opposite problem that it is far too rudimentary to identify any deeper correlation in the data. KNN models are also notoriously vulnerable to poor performance when trained on unbalanced data and, in our data set, the number of unique data points for some activities outweigh that of others, possibly leading to the observed lower accuracy. With its significantly lower accuracy, the MLPClassifier model was removed from further testing.

Generalised Accuracy

When the remaining models were evaluated using a complete unseen session as the test set, there was a notable decrease in classification accuracy for every classifier, ranging from a 27.56% to 43.88% decrease. This decrease was expected due to the limited number of training sessions, with each session having the potential being very different from each other, due to random error or differing behaviour between dogs when exercising. The smallest decrease, corresponding to an initial stronger generalisation ability, was observed with the RandomForestClassifier, whereas the weakest classifier for this purpose seemed to be the KNeighboursClassifier. These strengths or weaknesses may be the result of assumptions made by the classifiers; Random Forest techniques make few assumptions about the distribution of the data, but nearest neighbour makes a strong assumption that all variables' values will be distributed evenly, which is not necessarily true of the actual data collected from dogs. Again, in order to reduce the quantity of testing required, only those models reaching an accuracy above 65% were permitted for optimisation and further testing.

Optimised Accuracy

As stated in Subsection 6.3.1, optimisation was performed on the remaining classifier models: 'XGBClassifier', 'HistGradientBoostingClassifier' & 'RandomForestClassifier', using a grid search implemented by the scikit-learn API [26]. The hyperparameters searched for each model depended on their implementation, as well as the speed at which training/testing could be conducted for each iteration (slower models were evaluated over fewer possible values than faster models). The models optimised for unseen session prediction accuracy are displayed in Table 7.2. Any parameters not present in the table, but which are present in the documentation take on their default value, as per their respective documentation.

Classifier	**Optimised Parameters**
XGBClassifier	$objective = \text{'}multi : softmax'$, $n\_estimators = 100$, $max\_depth = 10$, $reg\_alpha = 0.2$, $reg\_lambda = 0.2$
HistGradientBoostingClassifier	$max\_iter = 100$, $max\_bins = 100$, $max\_depth = 7$, $l2\_regularization = 0.25$, $max\_leaf\_nodes = 50$
RandomForestClassifier	$n\_estimators = 70$, $criterion = \text{'}entropy'$

Table 7.2: Optimised supervised classifier hyperparameters

Using the parameters in the table above, the optimised classifiers were able to obtain the accuracies given in Table 7.1 over all sessions. The models were tested on the entirety of one session, having been trained on all data from the remaining sessions. It appears that the session the generalised accuracies were obtained from produced optimistic accuracies and the final observed activity recognition accuracy was several standard errors below this. From these optimised overall accuracies, the Random Forest classifier model appears to outperform both XGBoost and HistGradientBoosting classifiers, but, with standard errors factored into this, these differences are not particularly significant and could be improved by testing the models on more sessions of data. One of the project's regrets is the fairly small number of labelled sessions possible to conduct and label in the short time of the module. For the purposes of implementation into a live tool, a more indicative metric would be using the activity confusion matrices available in

Subsection 7.1.3.

7.1.2 Feature Importance

One significant difficulty with training classifier models up to this section has been the training time and the increased complexity of a large data set with 57 variables used for various tasks. Not only will a high number of variables cause an increase in training duration, there may well be redundant variables included in those 57 that could possibly confuse models and lead to deviation from correct classification. As such, it is important to analyse the effects of removing a set of variables from the data set when training and testing for the classifiers identified as accurate or usable. In addition to the effect on training times for models, data analysis and feature generation forming the data set on which to train often takes a longer time than the training itself, especially on a very long session. And so reducing the feature space would again aid the deployment of such a system on low-power laptops.

A method by which to remove redundant variables is to analyse the current feature importance in determining activity according to the classifiers themselves. The scikit-learn package provides built-in support for such a task and it is possible to plot these values as shown in Figure 7.1 [26], demonstrating the feature importances for the RandomForestClassifier. Due to its 'experimental' status, it was not possible to perform like-for-like analysis on HistGradientBoostingClassifier. For the sake of brevity, the feature importance graph for the XGBClassifier model appears in Appendix G/E/Feature Importance Plots. The 'Importance' value (y-axis) plotted here represents the fraction of total nodes within the ensemble that divide the data based on that variable.

From the plot produced from the Random Forest classifier, even without any form of L1 regularisation, the model seems to be placing a large importance on a relatively small number of features. For example, from inspection, the windowed variances of the raw 6-axis IMU data (XYZ-acceleration and XYZ-gyration values) makes up roughly 30% of total feature importance. These variables would correlate with a rough parallel of activity level (a larger variance indicates more movement in a short space of time), which would be most useful for dividing the data between more active (galloping, trotting and playing) activities and less active (laying, sitting and standing). Importantly, this speculation is supported by the project findings in Subsection 7.1.3.

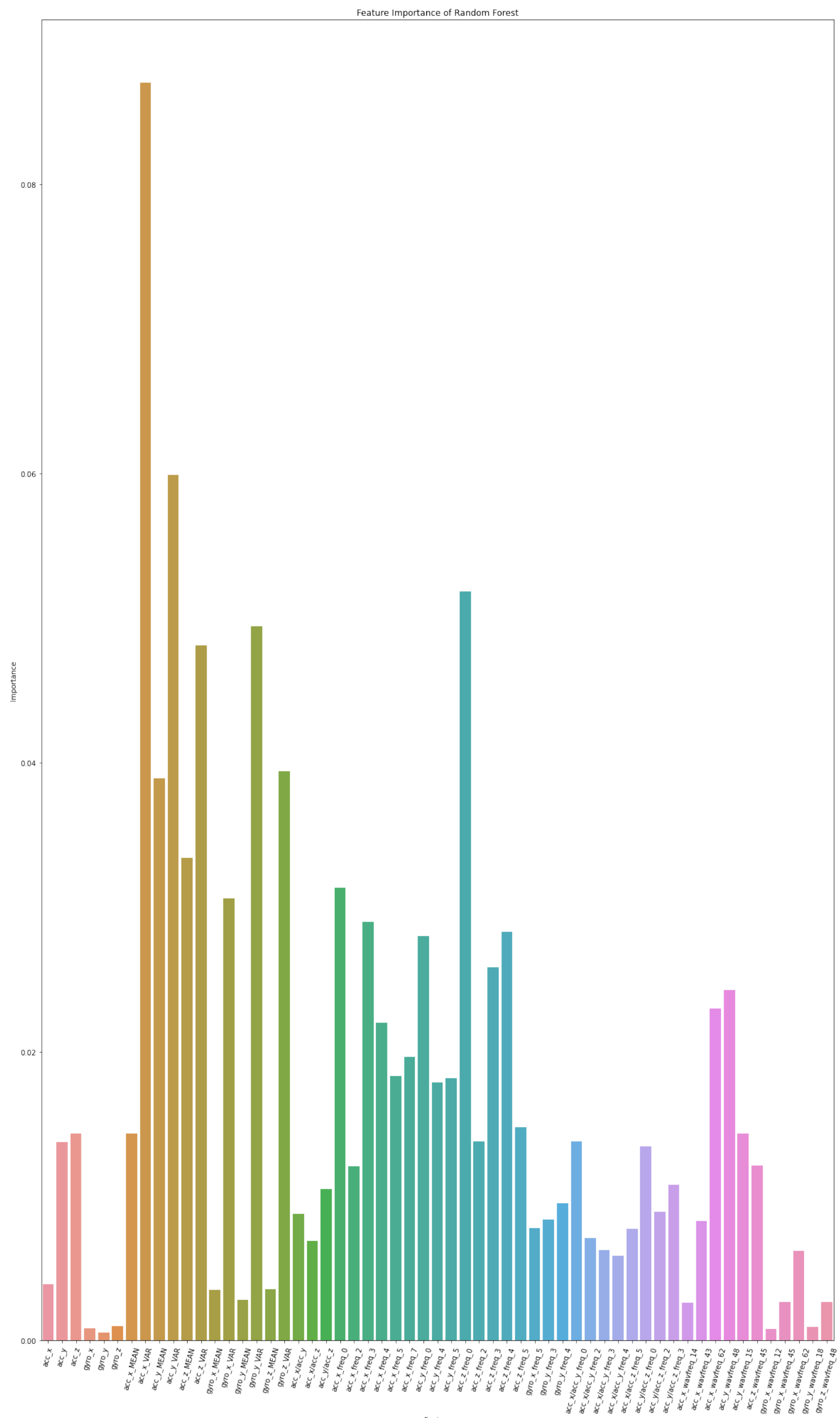


Figure 7.1: Feature Importance for RandomForestClassifier

Having analysed the relative importances of features for the two models, in order to remove redundant variables, features with importances below a given threshold, determined by the feature importance mean and distribution, are removed from the training and test data and the classifiers assessed by their session-wise accuracy once more. A full list of the variables remaining and the relevant threshold values for each model is present in Appendix G/E/'Important_Reduced_Features.txt'. The mean session-wise accuracy is given in Table 7.3.

Classifier	**Full Feature Acc.**	**Reduced Feature Acc.**
XGBClassifier	61.58% ± 3.94%	57.65% ± 4.62%
RandomForestClassifier	64.09% ± 3.10%	62.43% ± 2.98%

Table 7.3: Reduced feature accuracy comparison

As shown, from reducing the feature space to include only those features considered most important by their respective classifiers, there is a very slight decrease in classification accuracy, but only within one standard error of the accuracy with a complete set of features. This method for removing redundant features is evidently a very promising avenue that should be explored further when deployed. In the delivered product, however, all features are used in training as, without access to data from African Wild Dogs, there is no way of analysing whether the same feature importances will be observed in that genus.

Interestingly, comparing the results gathered through this method with the results obtained from [17], the relative importances of the bi-dependent features produced in Section 6.2 is significantly lower than those identified in existing literature. Notably, however, the standard deviation (another variance measure) features are identified therein as the most governing variables for the models tested, very similar to what is observed in this project's models. The decreased importance of bi-dependent features could possibly be a result of device placement with respect to the dogs' necks (The previous study attached the device to the rear of the neck, whereas this study places the device beneath).

7.1.3 Activity Confusion

Although inspecting the overall classification accuracies of the chosen classifiers certainly provides a reasonable perspective on the performance of the models on the data gathered, these accuracies do not inform on how well each activity is being classified against each other. After all, if a session contains solely the 'walking' activity, a classifier that consistently predicts 'walking' will have 100% accuracy on such a session. To this end, a number of confusion matrices have been constructed from the observed classification accuracies over a number of activities. As described previously, there is an unequal distribution of activities within the sessions recorded and thus, the confusion matrix is not necessarily indicative of overall performance. A confusion matrix with a large selection of activities present, for the Random Forest Classifier is shown in Figure 7.2. A complete set of confusion matrices for both models is available in Appendix G/E/'Session-Wise Confusion Matrices'.

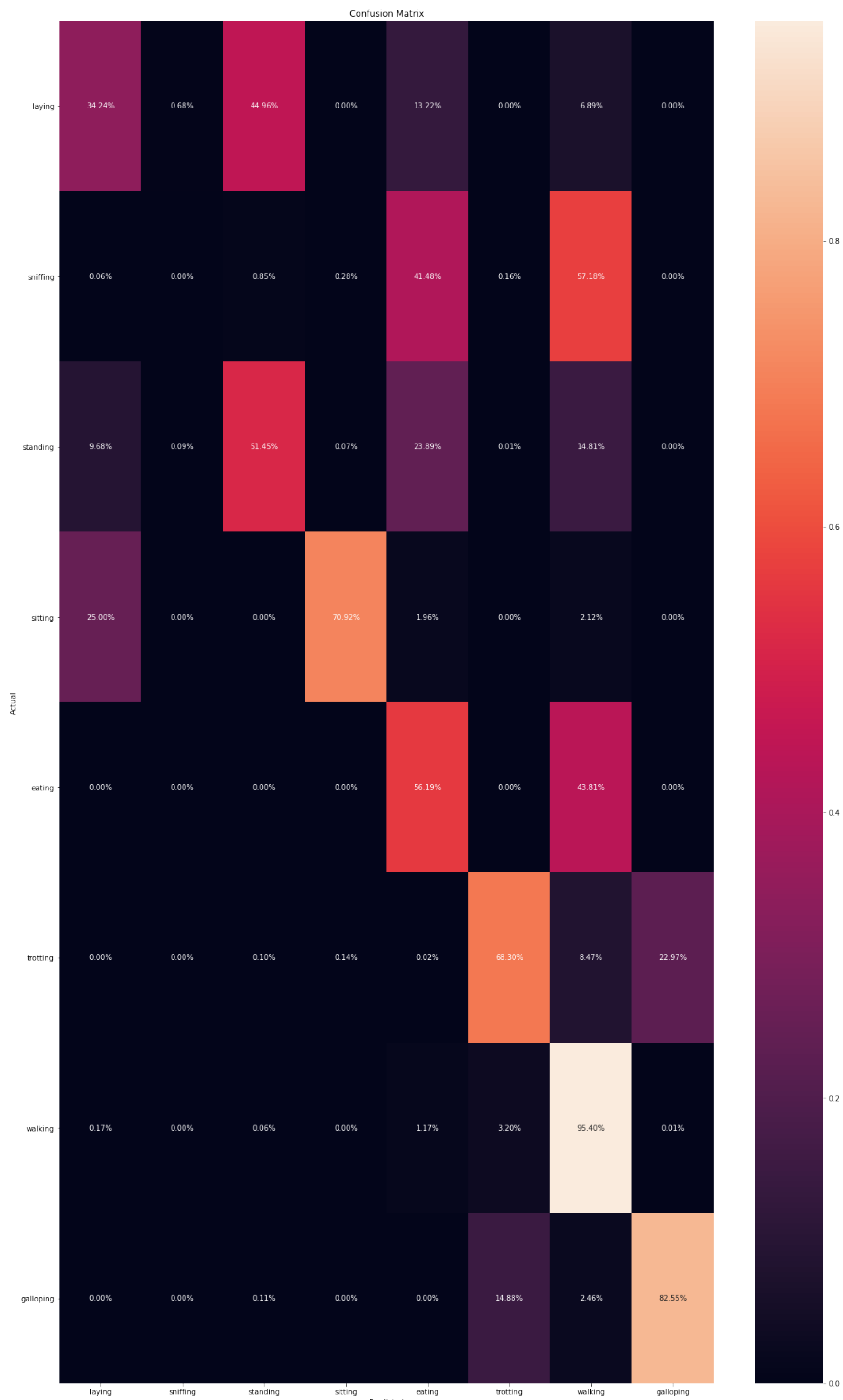
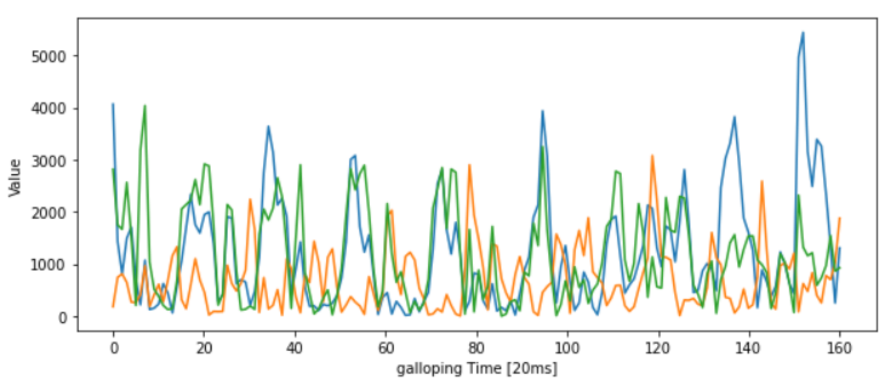


Figure 7.2: Confusion matrix for RandomForestClassifier

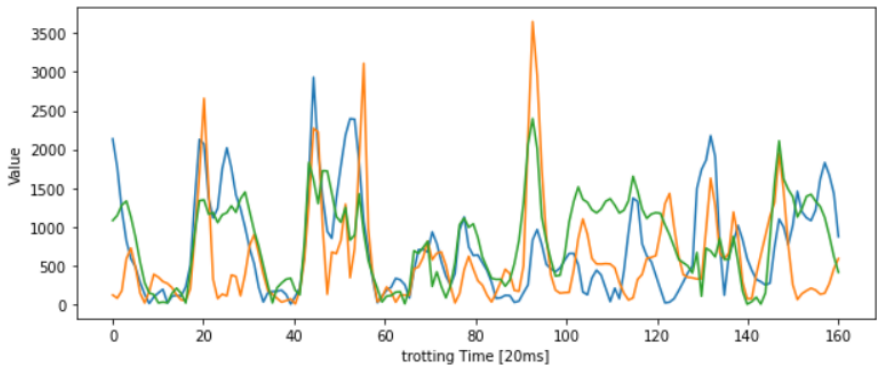
Notably, there is significant confusion within subsets of activities observed in the data collection sessions, with little confusion amongst other activities. These subsets can be summarily divided into the intensity of the activity being performed at that data point. For example, trotting and galloping activities are of very similar intensity, as is laying, sitting and standing. Finally, a moderate intensity subset consisting of sniffing, eating and walking can be seen.

An important reason behind this separation is the significance placed on variance features within the dataset which, as expressed previously, are indicative of the activity intensity. As shown in Figure 7.2, the classifiers are able to identify galloping and trotting amongst themselves with reasonable accuracy on average, however, this accuracy has been shown to vary rapidly between sessions. These high intensity activities are remarkably similar when inspecting the raw data and vary only slightly with respect to their frequency and movement magnitudes, as demonstrated in Figure 7.3. Hence, these activities may be mistaken due to differences between dogs (weight, age & other variables).

Similarly, a large source of confusion was in identifying differences between sniffing and eating activities. In this instance, for example, no sniffing activity data points are classified correctly, often being mistaken for eating or walking. The confusion with eating is an understandable mistake by the classifier; both activities require a lowering of the nose to the ground, with fine, sporadic movements of the neck (A plot demonstrating this similarity is shown in Figure 7.4). As such, when looking at the traces, these two activities are near-identical and the differences are imperceptible to the team's labellers. The mistaking of these activities for walking is less simply explained, but may be a result of proximity, or even potentially overlap (which was avoided at most), of these activities. Unavoidably, sniffing often takes place whilst the dog is walking and 'exploring' its vicinity, and hence may possess common frequency values between the activities.

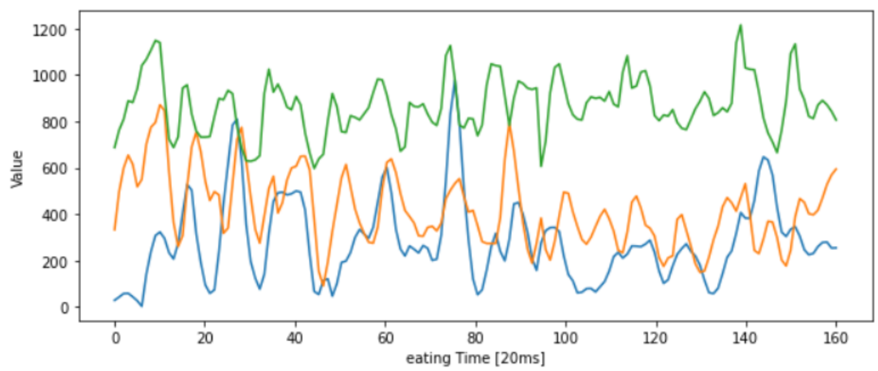


(a) Galloping

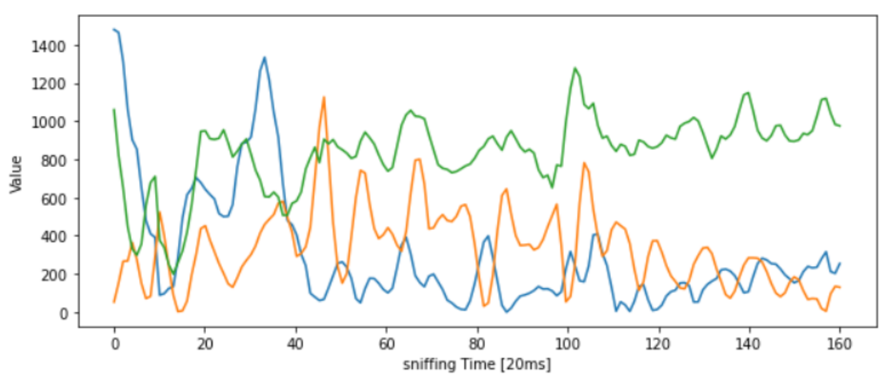


(b) Trotting

Figure 7.3: Acceleration data plots for galloping and trotting activities



(a) Eating



(b) Sniffing

Figure 7.4: Acceleration data plots for eating and sniffing activities

7.2 Unsupervised Learning

7.2.1 Density Based Methods

Evaluation

As the density based methods relied on there being distinct regions of low and high density, this method's success relied on the dataset having such properties. It also assumed that areas of high and low density correlated to separate dog activities. These methods also required some predetermined parameters approximated from the raw data. Making sure these approximations were accurate was essential for successful clustering. The difficulty in this investigation was determining if the clusters, hierarchical structures and outlier points found were useful to the project goal.

The raw data was clustered using several of the techniques, and then the activity labels within each cluster were compared to the cluster labels. Doing this would see if the clustering was able to pick up on any of the dog activities. This also evaluates if certain activities were grouped together, showing a hierarchical structure. Clusters with only a few data points would also be considered as outliers.

Results

When analysing the combined dataset, the density based methods only found 1 cluster (DBSCAN found 2 as it also creates an outlier cluster) as shown in table 7.4. Due to the sporadic data collected over several large dog sessions there was just a constant high area of dense data, so the 11 activities could not be identified. This demonstrated that density based clustering could not be used to classify or find helpful patterns within the raw combined dataset.

Clustering Method	**Number of Clusters**	**Number of Activities**
Mean Shift	1	11
DBSCAN	2	11
OPTICS	1	11
Affinity Propagation	1	11

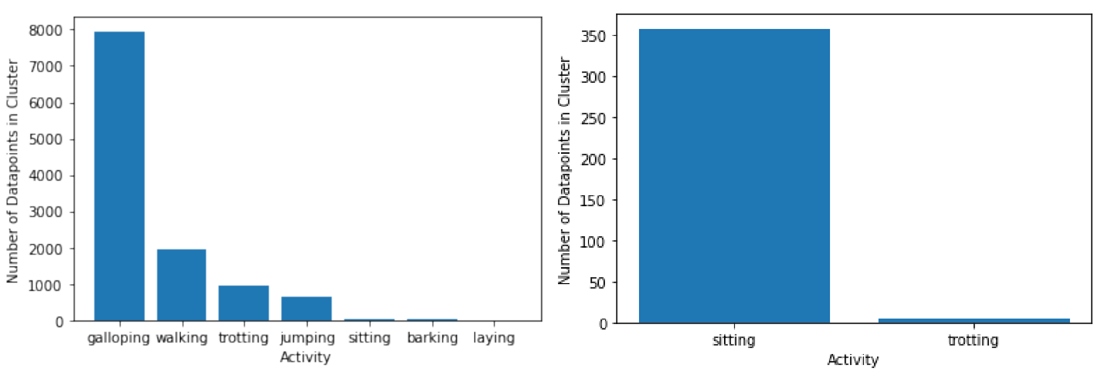
Table 7.4: Number of clusters found in the whole dataset

Analysing each individual session, the density based methods were able to cluster the data into distinct clusters. The best method was found to be the DBSCAN algorithm, with input parameters approximated from the dataset. Table 7.5 shows the number of clusters found, compared to the number of distinct activities within each session.

Session	**Number of Clusters**	**Number of Activities**
2020-11-05	7	7
2020-11-16	10	9
2020-11-23	18	4
2020-11-27	7	8

Table 7.5: Number of clusters found per session using DBSCAN

Although 3 out of the 4 sessions looked to produce promising results, analysing the distribution of activities within each cluster exposed the inaccuracy of this method. Figure 7.5 shows two example cluster distributions illustrating the problem. Although the number of clusters found was close to the number of activities present, each cluster did not contain a unique majority of activity labels. This meant that the density based methods were not accurately identifying the dog activities. Figure 7.5a is an example of the distribution of most clusters in the 2020-11-05 session. They were dominated by the galloping activity label. Figure 7.5b shows a unique cluster in the 2020-11-16 session, however can not be used to identify the sitting activity, due to lack of data points. These were problems with all of the clusters found.



(a) Cluster_1 from 2020-11-05 (b) Cluster_4 from 2020-11-16

Figure 7.5: Distribution of activity labels within individual clusters

PCA was used to help visualise the data and potentially reduce redundant features. Unfortunately this did not help improve clustering the whole dataset or each individual sessions. When reducing the data to 2 PCA axis, they could only explain 30% of the total variance within the data. DBSCAN was able to find supposed outliers, however when analysing this cluster, it just consisted of random data points from each label. There was no evidence that these were actual outlier points.

Unfortunately density based methods were not suitable for clustering the data into distinct activities, finding a hierarchical structure or finding outlier points. It was clear that euclidean distance and density was not the correct way to classify the dog data.

7.2.2 K Means Methods

Evaluation

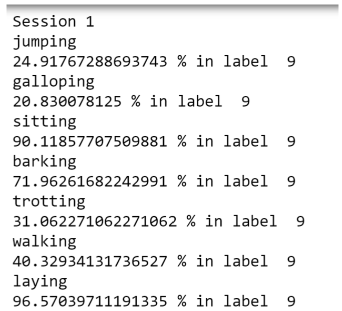
As the K Means method is only given the number of clusters or activities, and not labels for these activities, it is not straightforward to evaluate it's effectiveness. This is due to the fact that, while it will classify different data points into clusters, each cluster is not inherently tied to one activity, and so it can only be evaluated by comparing the cluster distribution of each activity to one another. For example, if a significant percentage of points labelled as "galloping" were placed in cluster 3, whilst "trotting" points were largely placed in cluster 4, this shows that these activities can be separated by the K Means method.

For each session, the activities that were not present were removed, and then each activity was plotted against the 10 clusters as a heatmap, with the square intensity representing the percentage of points that fell in that cluster. Along with this, the most favoured cluster for each activity was identified.

Results

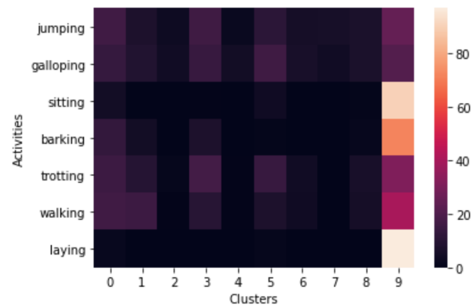
The results for each session were calculated individually, and initially these are the best indicator of success. If the K Means algorithm can be shown to separate activities well within one session, the combined sessions can be analysed to see if this separation can be generalised to multiple sessions or dogs.

As shown in Figure 7.6a the clustering of the Raw 6-Axis IMU data for session 1 did not separate activities at all, placing all activities present in one cluster. The only sensible pattern to be seen here is more "active" activities such as trotting, galloping, and walking were spread much more evenly between clusters as seen in Figure 7.6b. Whilst more inactive activities were strongly clustered to cluster 9. This is likely due to that with raw data, inactive activities will have small range in their accelerometer and gyroscope values, sitting close to zero, whilst more active activities will have more extreme values.



```
Session 1
jumping
24.91767288693743 % in label  9
galloping
20.830078125 % in label  9
sitting
90.11857707509881 % in label  9
barking
71.96261682242991 % in label  9
trotting
31.062271062271062 % in label  9
walking
40.32934131736527 % in label  9
laying
96.57039711191335 % in label  9
```

(a) Favoured Clusters



(b) Heatmap

Figure 7.6: Clustering results of Session 1 with Raw 6-Axis IMU Data

Adding more features to the dataset (Mean, Variance, Reciprocal, and Fourier) did very little to improve the clustering performance, with inactive activities still being strongly classified into one cluster, and active activities spread more evenly between clusters. Jumping and galloping consistently stood out as being placed in distinct clusters, but only weakly with between 12% and 20% of points placed in the most favoured cluster.

7.3 Semi-supervised Learning

The accuracy of the model when trained on the first 4 sessions, and used to classify the pseudo labelled data was 74.468%. This was about expected for unseen data, based on the results in Section 7.1. When the classifier was retrained with the pseudo labelled data, and tested on the test set the accuracy actually fell slightly to 74.093%. However this change varied from run to run.

This poor performance may be due to the accuracy of the model being too low on the first pass, or due to not having enough data to be able to carry out this technique (the pseudo labelling and test sets were quite small subsequently). This performance meant that Pseudo Labelling was not brought any further in the project and not incorporated into any part of our final product.

7.4 Deep Learning

Evaluation

Scenarios are created based on three variables, 'Train/Test Split', Pre-processing and DL architectures. There are three test cases for 'Train/Test Split', 2:1 ratio for pets and security dogs respectively, one security dog session in each partition and a split without partition by subjects. Pre-processing scenarios explored were with or without using data generator. DL architectures are on CNN, DeepConvLSTM and Time-CNN. All test scenarios had a batch size of 32 on 80 epochs, with an input shape of (None, 80, 6). The optimizer used for all cases was 'Adam' on default settings. These models were trained on 5 classes, 'Walking', 'Trotting', 'Galloping', 'Eating' and 'Barking'. 'ReduceLRonPlateau' was also used to monitor the validation accuracy metric. All test scenarios were executed 5 times to get an average result on training, validation and testing accuracy.

To adhere to Specification 4.2, specifically non-GPU intensive methods were preferred, all investigation on DL approaches were implemented and executed with CPU. Notably, the script was run on a hosted server from University of Southampton (YANN Server), which has two Intel Xeon E5-2623 v4 2.6GHz 8 thread processors and 126GB of RAM.

Results

Table 7.6 shows the results for all test scenarios. In general, complex architectures like CNN and DeepConvLSTM tend to overfit. This is due to possibly overly complex model with excessive parameters. The recommended settings with Time-CNN is a batch size 16 over 2000 epochs, hence results with Time-CNN shows behaviour of needing more time to learn since the model were trained on batch size 32 with 80 epochs. Data generators did not improve model performance due to the usage of RUS+ROS prior to that. Non-partitioned data test scenarios have an overall better performance since it is testing and validating on seen data, lowering the chances of models learning incorrect weights.

'Barking' activity accuracy in particular was considerably lower in samples. All data of this activity was removed for further testing. Following the recommended settings for Time-CNN, more tests have been done on using this architecture. However, batch size of 16 on 2000 epochs took approximately 3 hours to train, therefore this test case was not continued any further. In addition, pre-processing using only data generator were also tested. Similar behaviour was found. A test case of not using both RUS+ROS and data generator too were explored. Though

Test Scenarios			Results Overall		
Train/Test Split	**Preprocessing**	**DL Architecture**	**Train Acc.**	**Valid Acc.**	**Test Acc.**
4 (P) // 2 (S)	RUS+ROS	CNN	100.00	46.20	11.00
4 (3P+1S) // 2(1P+1S)	RUS+ROS	CNN	100.00	28.20	10.00
N/A	RUS+ROS	CNN	100.00	47.80	45.40
4 (P) // 2 (S)	RUS+ROS	DeepConvLSTM	100.00	55.00	12.20
4 (P) // 2 (S)	RUS+ROS+Data Generator	DeepConvLSTM	100.00	45.80	11.00
4 (3P+1S) // 2(1P+1S)	RUS+ROS	DeepConvLSTM	100.00	24.20	10.20
4 (3P+1S) // 2(1P+1S)	RUS+ROS+Data Generator	DeepConvLSTM	100.00	29.00	10.00
N/A	RUS+ROS	DeepConvLSTM	100.00	58.00	42.00
N/A	RUS+ROS+Data Generator	DeepConvLSTM	100.00	57.80	46.20
4 (P) // 2 (S)	RUS+ROS	Time-CNN	96.80	45.40	11.00
4 (P) // 2 (S)	RUS+ROS+Data Generator	Time-CNN	100.00	62.60	12.20
4 (3P+1S) // 2(1P+1S)	RUS+ROS	Time-CNN	97.20	51.00	8.20
4 (3P+1S) // 2(1P+1S)	RUS+ROS+Data Generator	Time-CNN	100.00	64.00	16.40
N/A	RUS+ROS	Time-CNN	77.4	25.60	22.40
N/A	RUS+ROS+Data Generator	Time-CNN	100.00	36.60	58.20

Table 7.6: Results of all test scenarios on 3 models for 5 runs[1]

performance for this case were substantially better in terms of accuracy, model suffered training, validating and testing on only one class as seen on Figure 7.7.

Deep learning requires large amount of data and high computational power. Only a limited number of sessions were collected. As compared to sessions organised, there were a significant number affected by data corruption, leading to only 6 sessions being collected. Among the 6 sessions, 4 were carried out by pet dogs and the remaining 2 by professionally trained security dogs. After a train test split and reshaping of data, only 2500 samples were obtained, even after Random Under and Oversampling. Lack of data made it challenging for the DL models to learn any useful weights. High class imbalance also persists even though efforts have been made through the implementation of RUS, ROS and data generator, with ROS resulting in high data bias. Data generator also did not work because behaviours between dogs differ even when carrying out the same activity. For example, a subject 1 would walk while sniffing while subject 3 would walk with its head up. Another example that could be made is with the 'eating' activity. Subject 2 would be classified as eating when eating treats, which is done by dog handlers holding dog threats near to its mouth, making its head positioned upwards. In contrast, Subject 1 would be eating from a bowl and that results in its head positioned downwards.

Due to the unsatisfactory performance of deep learning approaches, it was decided that this would not be included in the final deliverable. However, techniques and architectures mentioned may be worth experimenting if more data and a more balanced set of data was obtained. Transfer learning could also be looked at with models pre-trained on human activity recognition. It is believed that there are similarities between human activities and dog activities. An example that could be made is that when eating, both humans and dogs angle their head downwards, though with the condition of similar sensor placements.

	precision	recall	f1-score	support
0	0.00	0.00	0.00	453
1	0.00	0.00	0.00	377
2	0.72	1.00	0.84	2215
3	0.00	0.00	0.00	35
accuracy			0.72	3080
macro avg	0.18	0.25	0.21	3080
weighted avg	0.52	0.72	0.60	3080

	precision	recall	f1-score	support
0	0.00	0.00	0.00	111
1	0.00	0.00	0.00	96
2	0.72	1.00	0.84	553
3	0.00	0.00	0.00	8
accuracy			0.72	768
macro avg	0.18	0.25	0.21	768
weighted avg	0.52	0.72	0.60	768

	precision	recall	f1-score	support
0	0.00	0.00	0.00	272
1	0.00	0.00	0.00	728
2	0.48	1.00	0.65	1215
3	0.00	0.00	0.00	295
accuracy			0.48	2510
macro avg	0.12	0.25	0.16	2510
weighted avg	0.23	0.48	0.32	2510

Figure 7.7: F1 metrics on all classes. (Top - Train Data), (Middle - Validation Data), (Bottom - Test Data) and (0,1,2,3 - Galloping, Trotting, Walking, Eating)

Chapter 8

Project Deliverables

8.1 Device Mount 3D Model

So as to provide the customer with a system by which to attach their proprietary device to a generic canine collar, the team has created a 3D printable device mount that encloses the existing IMU casing. The design supports wireless charging in the same position as was supported by the enclosed device originally, but provides an alterable slide on mechanism for easily connecting to generic collars. To allow for a wider range of supported collar sizes, such as those often attached to African Wild Dogs in Zimbabwe, the .stl file format is able to be imported to a 3D modelling tool of choice and the mechanism expanded. The final design of the collar attachable device mount can be seen in Figure 8.1.

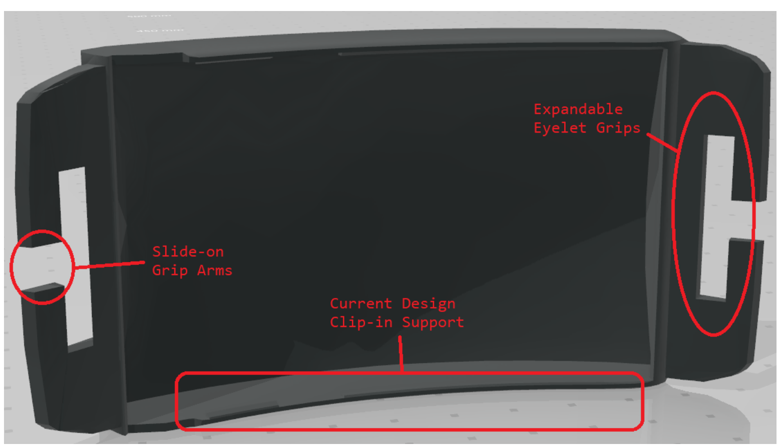


Figure 8.1: Labelled Device Mount Design

The complete 3D printable device mount is available for download in Appendix G/A.

8.2 Command Line Tool

In order to provide the customer with a consolidation of the team's work in the Artificial Intelligence phase of the project, it was decided to build a command line tool containing features of this work. The tool was also designed to be actually useful, allowing the user to classify domestic dogs, and hopefully African Wild Dogs.

8.2.1 Functionality

The tool was designed to have a number of different functions, that would make up the components required to train a classifier from training data, and use it to predict activities in unlabelled data. These functions were broken down into the following:

1. Label data
2. Add features to that data
3. Train a classifier
4. Add features to unlabelled data
5. Predict activities

1. The label function goes through the"Data" sub-directory and for each session, if there is a "raw.pkl" and a "labels.csv" file, the raw data is labelled, and saved as "labelled.pkl".
2. Depending on the level specified (1,2,3,4), features are added to the labelled data and saved to disk as "dataFeatured.pkl". The levels are, Mean and Variance, Reciprocal, Fourier, and Wavelet respectively.
3. A classifier of the type specified is trained on the featured data and saved to disk. The classifier types are Extra Trees, Random Forest, and Ada Boost.
4. The unlabelled data file specified as a parameter has features added it to the level that was used to train the classifier.
5. The saved classifier is used to predict the activities of the data file specified as a parameter. The labels predicted are saved as a CSV.

A full usage guide is included in Appendix B.

8.2.2 Structure

The Command Line Tool is a Python script, designed to be run from a command line terminal, and interacts with files in specific locations in it's directory. The usage prompt can be seen in Figure 8.2.

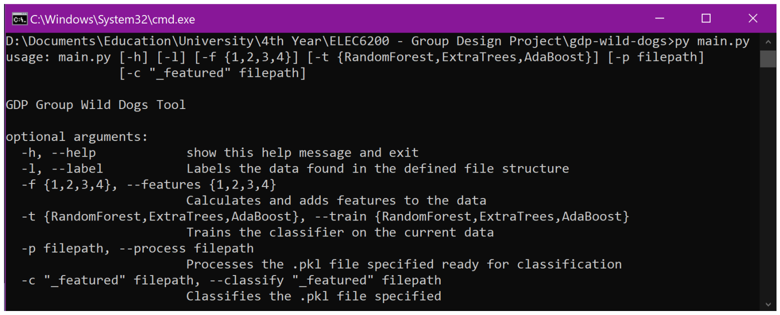


Figure 8.2: CLI Tool Usage

Before using the command line tool, an "environment.yml" can be ran using conda environment management system. This creates an environment with all required libraries and depencies for all the scripts to run smoothly.
The file that is actually running, "main.py", contains the command line options and calls functions from other files (modules). The "argparse" module was used to turn this script into a command line tool, that can be run with options and parameters. The rest of the code is found in label.py, features.py, train.py, and classify.py. The majority of this code is taken directly from the Python Jupyter Notebooks from this project, and is modified to work with the more modular architecture of the tool (detailed below), and certain files in the directory.

In order to make the tool very modular in functionality, and make it less memory intensive, it was decided that each function would load data from disk, and store it again on disk. This meant that carrying out multiple functions at one time would take longer due to this loading, but would require less data to be stored in RAM and so would be more likely to complete on low memory systems. This also made system design easier when allowing these functions to be run independently. For example labelling data would load an unlabelled session one at a time, label it, and store the labelled data back to disk.

For this kind of architecture, other components needed to also be saved to disk. The first of these was the classifier, to allow the user to train this once, and use it multiple times to classify. The "pickle" module was used here, at the end of the "train" function, the trained classifier object is stored on disk. This can then be loaded when the "classify" function is run. The second component was the feature level used to train the current classifier. This was also stored using pickle, and is loaded in the "process" function, to add the same level of features to the unlabelled data as that which was used to train the classifier.

Chapter 9

Project Management

The success of the project was mainly due to the management techniques implemented over the project. This involved identifying and mitigating potential risks, allocating each task correctly, managing and meeting the deadlines set, using a file/code management system, resolving conflicts, and working as a team.

9.1 Risk Assessment

In any project, problems and challenges were expected. A risk assessment was performed when starting the project to identify problems that could arise, how likely and influential they would be and how they would be dealt with. Table 9.1 shows the constructed risk assessment. This risk assessment allowed the group to plan for the inevitable impacts of COVID-19, ensuring that the project was complete on time and to the best standards.

Risk	Likelihood	Impact	Prevention & Correction
National Lockdown	**Moderate**	**High**	*Ensure all non-remote work is completed as early on as possible.* *Ensure essential hardware stays with a single team member.* *Use cloud based services for simultaneous file and code management.* *Obtain all required equipment as early on as possible.* *Communicate via email and services such as Teams.* *Allow for extra time for each section in the Gantt Chart.*
COVID Related Illness	**Moderate**	**High**	*Aim to have minimal co-dependence between team members.* *Allow for extra time for each section in the Gantt Chart.* *Well document all work so other team members can continue with it.* *Communicate via email and services such as Teams.* *Take on more work in the later stages to ease the work load.* *Apply for a COVID extension.*
General Illness/Injuries	**Low**	**Moderate**	*Allow for extra time for each section in the Gantt Chart.* *Communicate the severity with team members to plan around it.*
Personal Hardware Failure	**Moderate**	**High**	*Ensure all GDP related work is on a cloud based system.* *Have spare hardware available, through the university.*
Data Loss/Corruption	**Moderate**	**High**	*Ensure all GDP related work is on a cloud based system.* *Perform personal backups on a USB.* *Save and upload all work on a daily basis, to avoid wasted time.*
Falling Behind Schedule	**Moderate**	**High**	*Allow for extra time for each section in the Gantt Chart.* *Alert team members to other commitments and deadlines.* *Suggest a new split of the workload amongst team members.* *Contact supervisor if feeling stressed and overwhelmed.*
Unable to Contact Supervisor	**Low**	**Moderate**	*Send emails well before a response is required.* *Be clear and concise when sending emails.* *Email Nick Harris if there is no response after 2 weeks.*
Unable to Contact Customer	**Moderate**	**High**	*Send emails well before a response is needed.* *Be clear and concise when emailing.* *Have one team member dedicated to emailing the customer.* *Contact supervisor for support and guidance.* *Contact Nick Harris if problem persists.*
Unable to Collect Dog Data	**Low**	**High**	*Contact a variety of sources to ask for access to large dogs.* *Research existing datasets that involve large dog data.* *Contact researchers for their datasets on this topic.* *Contact supervisor and customer for a change in project scope.*

Table 9.1: GDP Initial Risk Assessment

9.2 Task Allocation

When allocating both technical and non-technical tasks to members of the team, several factors were considered. These were the conclusions drawn from the skills audit, DISC Personality Test and the Team Roles Test taken at the beginning of the semester. This was a way to be fair with task allocation as these tests were an independent decision maker, resolving any potential conflicts.

From analysis of each team member's test results it was decided that the non technical roles would be split as:

- Charlie Steptoe – Team Lead, Project Manager
- Aran McConnell – Ethics Officer, Deadlines, Tech Coordinator
- Yue Kew – Finance Officer, Participant Contact, Tech Coordinator
- Rohan Bungre – Scheduling, External Contact

At the start of the project it was too soon to split the technical roles, as this required understanding of the data which still needed to be collected. The results of these tests were used to give each team member an initial responsibility for a single phase:

- Charlie Steptoe – Device Mount Creation
- Aran McConnell – Data Labelling
- Yue Kew – Data Collection
- Rohan Bungre – AI Modelling

9.2.1 Skills Audit

A skills audit is a useful tool for mapping out the skills and expertise of each team member. This would make sure that there was the right mix of skills and to identify any gaps in the group's overall skills. A skills audit, shown in Table 9.2, was performed to understand what tasks undertaken during the project was best tailored for each team member's strengths, skills and interests.

Skills	**Coding**	**Machine Learning**	**Data Analytics**	**Scripting**	**3D Modelling**	**3D Printing**
Rohan	7/10	8/10	9/10	7/10	3/10	1/10
Charlie	8/10	9/10	8/10	8/10	7/10	6/10
Yue	8/10	9/10	8/10	7/10	2/10	1/10
Aran	7/10	8/10	8/10	9/10	4/10	5/10

Table 9.2: Skills Audit

The skills audit showed that the AI modelling and data aspects of the project could be split amongst all team members. There were some tasks such as 3D modelling and printing that only Charlie was best suited for. This put the team in good stead for the rest of the project.

9.2.2 DISC Personality Test

DISC is a behavior assessment tool based on the four different personality traits which are Dominance(D), Inducement(I), Submission(S), and Compliance(C). The tool produced a personality profile which was used to understand how each team member behaves in various situations. For example how each team member would respond to challenges, their influence on others, their preferred working pace and how they would respond to rules and procedures.

The detailed reports for each team member can be found in the project archive, as laid out in Appendix F. Analysis of the reports showed that the team had a healthy mix of the four different personality traits. There was a dominance of the inducement trait, which related to a strong ability to work in a team. This bode well for the duration of the project.

9.2.3 Team Roles Test

The Team Roles test helped determine which specific team roles are best suited for each team member. These roles could have been functional, organizational, personal or even skillful. It was clear to see that the group should consist of different team roles, depending on the specific goals the team wanted to achieve.

The detailed reports for each team member are again to be found in the project archive. Analysis of the reports showed the dominant role of each team member. Charlie obtained the Executive trait, perfect for a leader role. Rohan obtained the Expert trait, perfect for complex problem solving. Aran obtained the Explorer trait, perfect for experimenting and creative thinking. Finally Yue obtained the Team Player trait, perfect for communication and conflict resolution.

9.3 Time Scheduling

At the onset of the project a Gantt chart was produced in order to plan out each stage of the project. This Gantt chart plans out each of the four phases of the project, along with other tasks that don't fit into a phase. Within these phases, the different tasks which the phases consist of are also planned out, with dependencies between tasks and phases illustrated. This Gantt chart can be found in Appendix C.

Whilst the project was planned out in great detail with a lot of thought, some tasks naturally drifted off this schedule due to unforeseen circumstances. In order to illustrate the changes that occurred in our schedule, a second, revised Gantt chart was created (see Appendix C).

9.3.1 Changes to Schedule

The first two sections laid out in the Gantt chart mostly went unchanged. The preparation section was completely mostly as planned, with some issues with the Ethics submission as described in section 9.5. Despite this, the actual creation of the Ethics documents did not stray from the outlined schedule. The Device Mount Creation phase also went largely as planned.

Upon reaching the Data Collection phase, this is when delays began to appear. Initially finding dog owning participants was very challenging, in part due to the COVID-19 Pandemic. Thus this increased the length of the "Find Suitable Dogs" task. Once appropriate dogs were found, another delay was faced due to the Ethics process. This is discussed in section 9.5. This increased the length of the "Host sessions and collect data" task. Overall data was collected between the 5th of November and the 3rd of December, whereas it was initially planned to task place between the 2nd and 15th of November. This delay pushed back subsequent phases due to their dependency on data collection.

The next phase that strayed from the initial schedule was the data labelling phase. The first of these changes was, this task was greatly extended as data collection took a lot longer than expected. Each session was labelled as soon as it was completed, as opposed to all sessions being labelled once data collection was completed, as initially planned. The labelling task also took a lot longer than expected. The number of hours required to manually label a session was greatly underestimated to begin with.

Finally, largely due to delays in the previous two phases, the "Create ML Architecture" task was delayed by a week. Whilst work on this could be started with only a single session of data, the delay in collecting a significant dataset meant that quantitative analysis was pushed back. Further to this, many of our team were occupied with collecting and labelling data early on, and could only commit time to this phase later into the project.

9.4 Management Techniques

For a successful project, it was important that the team implemented tried and tested management techniques, ensuring that the project would stay on schedule. During the project both Waterfall and Agile management techniques were implemented for an optimal overall project management style.

9.4.1 Waterfall Management

Waterfall project management involved splitting the project into sequential phases, where each new phase could only begin when the previous phase had been completed [7]. The waterfall system is a method for managing a project, where team members work in a linear fashion towards a set end goal. Each member had a clearly defined role and none of the phases or goals were expected to change. This method was ideal for managing the phase structure of our project. Each of the four phases had to happen in sequential order for the project to be successful and there was fixed goals set from the beginning by the external customer that could not be changed during the project.

9.4.2 Agile Management

Agile project management involved having short development cycles called "sprints" to focus on continuous improvement during the project [5]. The agile approach was very helpful to use during each individual phase of the project, especially the AI model generation. This was because it allowed the team to develop models that constantly improved at a fast rate. There was also no fixed constraints to the AI model, so its architecture could be adapted to meet the required accuracy as time went on, aligning with the principles of an agile approach.

9.4.3 Team Meetings

Meeting up as a team was essential to the success of this project. It was decided that the team would meet up weekly to discuss what everyone had been up to over the past week; how successful was the work done over the past week; what needed to be done next and finally if there were any concerns. This was accompanied by Charlie taking minutes for each meeting.

We also aimed to have monthly meetings with our supervisor to ensure that they were happy with our progress and that we were moving in the correct direction. These meetings were also accompanied with regular emails when minor concerns arose. We only had 3 meetings with the external customer due to busy schedules. These were to introduce ourselves, to understand the problem and what Mafic wanted and finally to learn about their labelling tool. We were able to email them when issues arose which will be discussed more in the customer interaction section.

9.4.4 Conflict Resolution

As with any group project, conflict was bound to arise in a situation where people were required to interact with each other. Managing this conflict was important for the success of the project. It was decided to use the Universities "Five Methods for Managing Conflict" [6] when conflict happened. Luckily this only happened once in a minor fashion, when there was a 50/50 split on whether to collect more dog data or to focus on the AI model. Both arguments had their merits, but by using the Compromise method this issue was resolved.

9.5 Animal Ethics

As the Data Collection phase (see Chapter 5) of the project involved working with Animal Participants, we were required under University policy to apply for Ethics approval through the University "ERGO 2" system.

Applying for an "Animal Ethics" approval through this system involved the creation of an Ethics Form, Risk Assessment, Participant Information Sheet, and a Consent Form for our Data Collection activities. These documents are included in appendix A. In addition to this a questionnaire was filled out to assess the category of our application. Due to our work not involving the collection of any identifiable or special category data from the human participants (dog owners), or causing any harm to the animals involved our application was classed as Category C. Our work also did not involve any procedures laid out in the Animals (Scientific Procedures) Act 1986, and complied with the Animal Welfare act 2006.

9.5.1 Issues

The ethics procedure caused a large delay in the project overall. Whilst the application was submitted in good time (Early November 2020), there was a large delay in it being approved. This delay caused a small delay in the data collection phase. Ultimately it was decided data collection would have to start before the Ethics Application was approved, or no data would be due to the short time frame of the project. The team had confidence that due to the straightforward nature of the Ethics Application, it would be approved in the due course. The data collection phase was therefore carried out as planned after a short delay, following the procedure described in our Ethics documents. Overall the team believe this was a good decision to begin data collection before the Ethics Approval, as it was not fully approved until the 15th of December. If we had waited until this date to collect data, we would not have had a project at all due to where data collection fell on the project's critical path.

9.6 Source Code and File Management

It was decided during the early stages of the project, that all files relevant to the GDP would be stored in an online cloud environment that could be accessed by all members of the group at the same time. This was done primarily due to the risk of hardware failure identified in the risk assessment. Doing this also allowed for communal documents that were able to be edited simultaneously, simplifying the process of report writing. It was also important that the team should use university accounts were possible to ensure a separation between personal and university files.

9.6.1 File Management

Due to the university providing Office 365 accounts for all students, it was decided to use OneDrive as the cloud file management application. Files relevant to the GDP were stored in a shared location where each team member was able to view and edit each file. Due to the Office 365 integration, documents could be created and edited within a Teams call, and still remain a part of the OneDrive.

As several of the reports needed to be written in Latex, the decision was made to use Overleaf. This was because Overleaf allowed the team to work on a single online document that could be simultaneously edited. This both improved efficiency and reliability whilst producing the reports.

9.6.2 Code Management

Several methods of code management were proposed during the project, however GitLab was decided on due to the university providing us with accounts. GitLab is a web-based tool that provides a Git-repository manager for source code management. Using Git allowed each team member to track the changes made to each file, producing a record of what has been done, and what could be reverted when needed. Git also made collaboration easier, allowing changes by multiple team members to all be merged into one source. GitLab also stores files on a University server, solving any issues around security.

Using git, each team member was able to work on the Dev branch of the project from a personal machine. To keep updated, each member would have to push and pull data from the main repository. When a section was complete, it would be merged with the master branch which wasn't changed again. This allowed for reliable code management throughout the project.

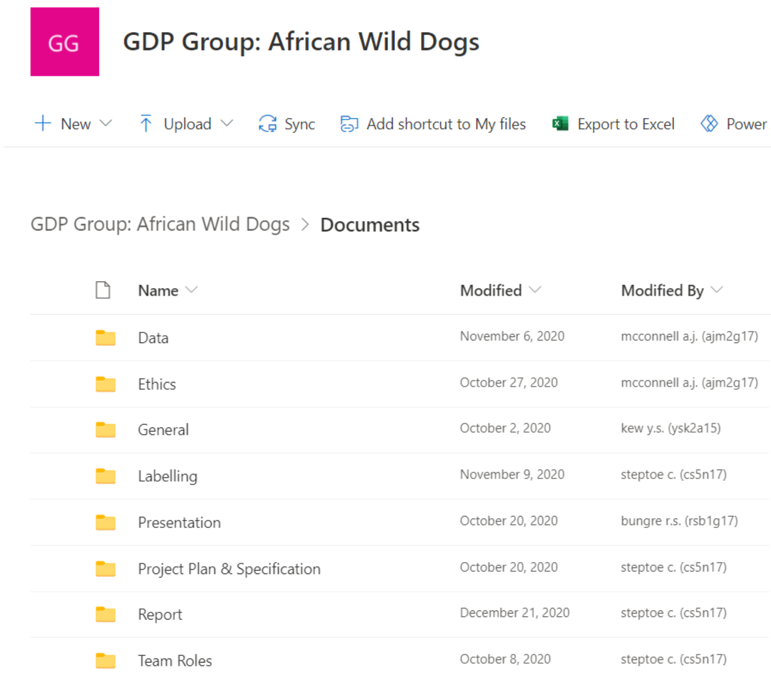


Figure 9.1: OneDrive Usage

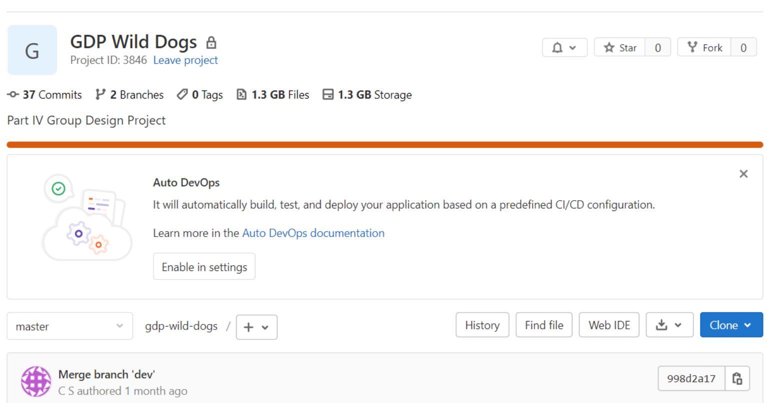


Figure 9.2: GitLab Usage

9.7 Customer Interaction

Overall, interactions between the customer and the team have been good. Emails were the main medium of communication, as preferred by the customer. At times, it would be more difficult to open communication with our contacts, though once lines were established, a video meeting on Zoom was typically arranged to resolve any ongoing issues. Understandably, the response times between the customer and the project team would vary, potentially indicating an area for improvement;

opening a direct line of communication would have been significantly more efficient and remove a number of blockers in the project timeline. Primarily, issues regarding retrieval of data and access/execution of a proprietary labelling tool posed a large obstacle to the project schedule. With this said, both customer and project team were unfamiliar with working alongside one another, with these issues being resolved or avoided altogether after further discussions.

Sensor Device

The team have requested an extra device from the first meeting with the customer under the motivation of collecting data on more than one subject at once. At the time, they did not have extra devices ready to be sent off, therefore the team worked with one device throughout most of the project. During data collection, a few problems were experienced by the team. There were times where a member of the team would go out for an arranged session with a volunteer, only to find out that the device was not logging at all, resulting in lost sessions, reducing our overall dataset. The feedback we have received from the customer was that this was due to the sensor device being switched on for too long so the battery was drained.

As a result, the team received two more sensor devices with an optimised battery life that lasts 72 hours on a full charge. However, new devices were only received approaching the end of the project, hence they was barely used. If the team were able to check whether the sensor device is indeed working, or if it is reading the correct values in real-time, lost sessions could be avoided.

Data Uploading and Retrieval

Data uploading and retrieval were always a problem for the team. As per the last paragraph on the sensor device, the team have received a total of three sensor device. However, sensor devices were programmed differently with two dissimilar WIFI SSID and password. This was identified through a series of emails and video meetings in which Will from Mafic walked the team through the set up of new SSID configurations that works with the old configurations. The team had to retrieve data for each session through emails because the AWS server hosted by our customer is only used internally within the company.

Due to issues with data uploading, the team experienced missing data on dates where sessions were carried out, be it data not uploaded to their server due to wrong WIFI configurations, sensor device not actually logging for those dates, or data lost on the server. The customer and team have been in touch with on this matter throughout the project. From data uploading to data retrieval for each session (sometimes multiple dates are requested at once), it would take a week up to a month to solve. It is decided that the team do not pursue resolving issues regarding data upload and data retrieval for the final few sessions after great efforts for 3 weeks, but instead work with existing data that were retrieved thus far. These cascading problems have caused a significant effect on the progress of the project. If data were unable to retrieve, data can not be labelled. This limits the amount of data that the team are able to work with, resulting in Phase 4 of the project, 'AI Model Generation', being delayed greatly. All phases of the project were highly dependent on previous phases.

Data on the AWS was identified by the M.A.C. address of sensor device. If there is a channel that allows direct access for third parties with defined M.A.C. address and defined port, this would help ensure that the sensor devices are working correctly. Also, the uploading and retrieval process would be a lot smoother without going through the customer.

Labelling Tool

With the labelling tool, the team had two main issues which resulted in an inability to label existing data that was retrieved. The Labelling tool is developed by the customer, written in Python and has only been used internally, therefore unforeseeable issues were experienced. A member of the team was unable to load the labelling tool, while it took approximately 5 minutes to launch for the rest of the team. The first issue was that traces could not be loaded into the labelling tool. The problem identified here was that the pickle version used to convert the data from AWS was different from the pickle version used in the labelling tool. There is a lot of library dependency issues for pickle and Python. The customer aided in reconverting data files into appropriate pickle version upon identification of the problem by the team.

Next issue was an inability to load video recordings. The reason was identified after a couple of video meetings with the customer. The problem identified was that audio and video files required a codec in order to be played in the tool. Once this codec was installed, the issue was resolved, the customer then walked the team through the procedures for labelling traces.

9.8 Effects of the Covid-19 Pandemic

The Covid-19 pandemic affected the project in several way over the semester. As the majority of the project's phases were dependant on each previous stage, a delay in any section due to Covid was propagated throughout the whole project. Even though some aspects of Covid was included in the risk assessment, this was not enough to stop the project from experiencing major delays.

Device Mount

Due to the social distancing measures in B16, the 3D printed mounting device took longer than expected to be produced. Even after emails were sent to book a lab session, this never happened. Eventually one of the B16 technicians was able to print the mount.

Data Collection

Due to the government's advice on social distancing and households meeting up outdoors, it was difficult to find willing participants for the data collection phase. This was further worsened by the national lockdowns where no data was allowed to be collected. Luckily a PhD student and the Hampshire Dog Club were happy to participate in a Covid safe way.

AI Model Generation

Although this phase was done over the Christmas break at home, one of the team members contracted Covid. This meant the team member was bedridden for 2 weeks over a crucial period. Fortunately the University provided the group with a week extension to help with the delays caused by Covid.

9.9 Report Breakdown

To ensure an even workload whilst writing the report, the team decided to split the total 20,000 word limit into an individual 5000 word contribution for each team member. As some team members had more relevant work to contribute towards the technical chapters, the others would contribute more to non-technical chapters. Whilst achieving a perfect split was difficult, the team were able to find a split that satisfied everyone. The individual report contribution is shown in table 9.3.

Team Member	**Report Contribution**	**Team Member**	**Report Contribution**
Charlie	3 - Device Mount Creation 6.1 - Dataset Analytics 6.2 - Feature Generation 6.3 - Supervised Learning 7.1 - Supervised Learning 8.1 - Device Mount 3D Model 10 - Conclusion	**Aran**	Abstract Statement of Originality 5 - Data Labelling 6.4.2 - K Means Clustering 6.5 - Semi Supervised Learning 7.2.2 - K Means Methods 7.3 - Semi Supervised Learning 8.2 - Command Line Tool 9.3 - Time Scheduling 9.5 - Animal Ethics Appendix
Rohan	Abstract Acknowledgements 1 - Introduction 2 - System Design & Specification 6 - AI Model Generation 6.4.1 - Density Based Clustering 6.7 - Comparison with Specification 7.2.1 - Density Based Methods 9 - Project Management	**Yue**	Abstract 4 - Data Collection 6.2.4 - Wavelet Transform 6.6 - Deep Learning 7.4 - Deep Learning 9.7 - Customer Interaction

Table 9.3: Individual Report Contribution

Chapter 10

Conclusion

10.1 Further Work

The prevailing phase of the project that requires a substantial quantity of further work is data collection. Regrettably, due to a limited number of those dog owners willing to participate in the study, and thus provide substantially more data to be used for training or testing purposes, as well as corruption or retrievability issues when using the current pipeline, only five full sessions were able to be labelled and used for machine learning. This number is much lower than was initially desired and it is the team's unanimous opinion that a larger selection of participant dogs would have improved classifier accuracy markedly. Fortunately, this project is due for deployment on African Wild Dogs in conjunction with conservation trusts where a much larger volume of data will be gathered and labelled for training purposes.

Additionally, it is recommended that, in order to reduce issues of data corruption in the future, the customer allows for local storage of data recovered from the IMU device. This improvement will not only ensure a backup is available on a client system if applicable, but will also help to facilitate a smooth transition to full system deployment, especially for a scenario in which internet capability may not be assumed.

Aside from limited data collection, the project was able to study the accuracies of a variety of classifier models on a somewhat limited training set. Although these studies are considered robust, due to severe time restrictions on the project, the team was not able to perform a full analysis of all classifiers considered. Additionally, the data being used for the testing over the course of this project is that of domestic dogs. Where the performance of classifiers may differ drastically when trained on data gathered from African Wild Dogs, it is the opinion of the project team that the customer should conduct additional analysis of the classifiers suggested to ensure robustness in the new scenario.

Finally, throughout the early stages of the project, the potential of using purely unsupervised or semi-supervised was discussed in great detail with the customer. Despite unsupervised learning techniques tested during the project having performed poorly, the semi-supervised learning and deep learning techniques analysed have been shown to have potential in the field of canine activity recognition. The team encourages the customer, and indeed others in the field, to pursue these possibilities further as such technology may bear impressive results in generalised ethology construction.

10.2 Project Summary

In conclusion, over the four month duration of this project, the team has been able to demonstrate an efficient and effective pipeline for the gathering of movement data of domestic dogs, labelled such data according to a reasonable schema and constructed machine learning models to classify data into this schema. During these phases, the project has: delivered on creating a suitable and reusable mounting mechanism for a proprietary device, successfully conducted over ten data collection sessions with domestic dogs and their owners, fully labelled five of these sessions (all sessions that were able to be retrieved) and assessed many fully-supervised, semi-supervised, unsupervised machine learning and deep learning techniques on a variety of metrics.

A particular benchmark by which to assess the success of this project has been to compare the eventual maximum obtained accuracy of any tested classifier against the 80% set out in the initial project specification. When inspecting intra-session activity classification, as many as 6 out of the 7 supervised classifiers tested surpassed this accuracy. Although this metric may be considered an absolute upper bound for the accuracy, it should be regarded with some caution as it may, on its own, be representative of over-fitting to some session. When taking inter-session accuracy into account, regrettably, no classifiers attained an average classification accuracy above 70%. Despite this, a maximum accuracy of roughly 64% on a significantly reduced quantity of training data was obtained.

With this result considered, the ultimate accuracy of such a complete system as that set out in this report has been shown to depend heavily on the quality and quantity of the data obtained. Hence, as this project moves forward with the customer to aid in identifying African Wild Dog behaviours, an emphasis must be placed on collecting a sizable set of labelled data. With this augmented data set, the benchmark accuracy attempted may very possibly be achieved.

Bibliography

[1] Agility Pet ID Tag. URL https://petidtagsexpress.co.uk.

[2] pickle — Python object serialization. URL https://docs.python.org/3/library/pickle.html.

[3] African Wild Dog Facts. URL https://www.worldwildlife.org/species/african-wild-dog.

[4] Canine Grooming Behavior, July 2016. URL https://veteriankey.com/canine-grooming-behavior/.

[5] Agile Project Management, 2021. URL https://www.apm.org.uk/resources/find-a-resource/agile-project-management/#:~:text=Agile%20%E2%80%93%20a%20project%20management%20approach,and%20eXtreme%20Programming%20(XP).

[6] Conflict Resolution, 2021. URL https://www.southampton.ac.uk/~assets/doc/hr/Five%20methods%20for%20managing%20conflict.pdf.

[7] Waterfall Project Management, 2021. URL https://www.wrike.com/project-management-guide/faq/what-is-waterfall-project-management/.

[8] Tianqi Chen and Carlos Guestrin. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4232-2. doi: 10.1145/2939672.2939785. URL http://doi.acm.org/10.1145/2939672.2939785. event-place: San Francisco, California, USA.

[9] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2020. URL http://www.blender.org.

[10] F D. Simple explanation of semi-supervised learning and pseudo labeling, 2017. URL https://towardsdatascience.com/simple-explanation-of-semi-supervised-learning-and-pseudo-labeling-c2218e8c769b

[11] Imad Dabbura. K-means clustering: Algorithm, applications, evaluation methods, and drawbacks, 2018. URL https://towardsdatascience.com/.

[12] Hoang Anh Dau, Eamonn Keogh, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, Yanping, Bing Hu, Nurjahan Begum, Anthony Bagnall, Abdullah Mueen, Gustavo Batista, and Hexagon-ML. The ucr time series classification archive, October 2018. https://www.cs.ucr.edu/~eamonn/time_series_data_2018/.

[13] Florenc Demrozi, Graziano Pravadelli, Azra Bihorac, and Parisa Rashidi. Human activity recognition using inertial, physiological and environmental sensors: A comprehensive survey. *IEEE Access*, 8:210816–210836, 2020. ISSN 2169-3536. doi: 10.1109/access.2020.3037715. URL http://dx.doi.org/10.1109/ACCESS.2020.3037715.

[14] Hassan ISMAIL FAWAZ. Deep Learning for Time Series Classification, January 2020. URL https://github.com/hfawaz/dl-4-tsc/blob/master/classifiers/cnn.py.

[15] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Deep learning for Time Series Classification: A review. *CoRR*, abs/1809.04356, 2018. URL http://arxiv.org/abs/1809.04356.

[16] Felipe Garcia. Deepconvlstm, January 2019. URL https://github.com/felipeagarcia/deepConvLSTM.

[17] Linda Gerencsér, Gábor Vásárhelyi, Máté Nagy, Tamas Vicsek, and Adam Miklósi. Identification of Behaviour in Freely Moving Dogs (Canis familiaris) Using Inertial Sensors. *PLoS ONE*, 8(10):e77814, October 2013. ISSN 1932-6203. doi: 10.1371/journal.pone.0077814. URL https://dx.plos.org/10.1371/journal.pone.0077814.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. *CoRR*, abs/1512.03385, 2015. URL http://arxiv.org/abs/1512.03385.

[19] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 3149–3157, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 978-1-5108-6096-4. event-place: Long Beach, California, USA.

[20] Cassim Ladha, Nils Hammerla, Emma Hughes, Patrick Olivier, and Thomas Ploetz. Dog's life: wearable activity recognition for dogs. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 415–418, Zurich Switzerland, September 2013. ACM. ISBN

978-1-4503-1770-2. doi: 10.1145/2493432.2493519. URL `https://dl.acm.org/doi/10.1145/2493432.2493519`.

[21] Jing Lin and Liangsheng Qu. Feature Extraction Based on Morlet Wavelet and its Application for Mechanical Fault Diagnosis. *Journal of Sound and Vibration*, 234(1):135–148, June 2000. ISSN 0022460X. doi: 10.1006/jsvi.2000.2864. URL `https://linkinghub.elsevier.com/retrieve/pii/S0022460X00928645`.

[22] Mohan. Activity Detection Using IMU Sensor, April 2019. URL `https://github.com/mohan-mj/Activity-Detection-using-IMU-sensor/blob/master/Human_Activity_Detection-1D%20CNN.ipynb`.

[23] Ronald M Nowak. *Walker's Carnivores of the World*. JHU Press, 2005.

[24] Helen O'Neill and Dani Rabaiotti. A dog's life: Using conservation technology to monitor African wild dogs, August 2017. URL `www.zsl.org`.

[25] Francisco Ordóñez and Daniel Roggen. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors*, 16: 115, 01 2016. doi: 10.3390/s16010115.

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

[27] Joshua Poduska. Density Based Clustering, 2021. URL `https://blog.dominodatalab.com/topology-and-density-based-clustering/`.

[28] M. J. Shensa. The discrete wavelet transform: wedding the a trous and mallat algorithms. *IEEE Transactions on Signal Processing*, 40(10):2464–2482, 1992. doi: 10.1109/78.157290.

[29] Amy Shojai. Canine Hunting Behavior, October 2019. URL `https://www.thesprucepets.com/canine-hunting-behavior-2804561`.

[30] Zhiguang Wang, Weizhong Yan, and Tim Oates. Time Series Classification from Scratch with Deep Neural Networks: A strong baseline. *CoRR*, abs/1611.06455, 2016. URL `http://arxiv.org/abs/1611.06455`.

[31] P. Zappi, T. Stiefmeier, E. Farella, D. Roggen, L. Benini, and G. Troster. Activity recognition from on-body sensors by classifier fusion: sensor scalability and robustness. In *2007 3rd International Conference on Intelligent Sensors, Sensor Networks and Information*, pages 281–286, 2007. doi: 10.1109/ISSNIP.2007.4496857.

[32] Bendong Zhao, Huanzhang Lu, Shangfeng Chen, Junliang Liu, and Dongya Wu. Convolutional neural networks for time series classification. *Journal of Systems Engineering and Electronics*, 28(1):162–169, 2017.

Chapter A

Ethics Documents

UNIVERSITY OF Southampton

ERGO Animal application form – Ethics form

All mandatory fields are marked (M*). Applications without mandatory fields completed are likely to be rejected by reviewers. Other fields are marked "if applicable". Help text is provided, where appropriate, in italics after each question.

1. APPLICANT DETAILS

1.1 (M*) Applicant name and email:	Rohan Bungre (rsb1g17), Aran McConnell (ajm2g17), Yue Kew (ysk2a15), Charlie Steptoe (cs5n17)
1.2 (M*) Supervisor name and email:	Dr Srinandan Dasmahapatra - sd@ecs.soton.ac.uk
1.3 Other researchers/collaborators and external personnel involved in study (if applicable): *Name, address, email, telephone*	Mafic Ltd, Woodford Green, Essex

2. STUDY DETAILS

2.1 (M*) Title of study:	Recognising Activities of African Wild Dogs using Machine Learning
2.2 (M*) Type of study (*e.g. Undergraduate, Doctorate, Masters, Staff*):	Undergraduate
2.3 i) (M*) Proposed start date (allow at least 1 month):	19/11/2020
2.3 ii) (M*) Proposed end date:	14/01/2020

2.4 (M*) What are the aims of this study?
To collect training and test data, in order to develop and Artificial Intelligence model that can recognise the activities of African Wild Dogs.

2.5 (M*) What are the objectives of this study?
To collect accelerometer and gyroscope data from a device attached to a dogs collar, whilst the dog carries out a range of standard behaviours. To record video of the dog performing these behaviours. To collect this data for a number of dogs to build up a good dataset. Finally, to use the video to label the recorded data and thus use It to train and test an Artificial Intelligence model.

2.6 (M*) Background to study (*a detailed and thorough rationale for conducting the study, listing all relevant publications*)**:**
This project involves working with an external company, Mafic Ltd, who have been approached by the makers of a TV documentary on African Wild Dogs. They want to use a device attached to the dog's collar, giving accelerometer readouts, to understand the

behaviour of the dogs. Using the readouts from the device and a Artificial Intelligence framework, different common activities of the dogs should be recognised and aid in understanding the behaviour of these dogs. To develop the Artificial Intelligence framework for this solution, we are using similar data from large domestic dogs to train the framework, as it is not possible to collect such data from African Wild Dogs. The behaviour of such large dogs should be similar enough to get a model of a reasonable accuracy. To train this model we need to collect data from domestic dogs, whilst carrying out basic tasks such as running, trotting, galloping, barking, sleeping, grooming, and eating. The dogs must be filmed for data collection, so the data can be labelling to each activity later on.

2.7 (M*) Has this work been done before? If so, what are you adding to previously published work?
Has not been carried out before to the best of our knowledge.

2.8 (M*) What are the benefits of this study?
When the Artificial Intelligence framework is complete is will act as a behavioural classifier to be used by documentary producers and conservationists. The use of this will reduce unnecessary intrusion into wild habitats, and reduce the expenditure for documentary producers and conservationists by having to spend time in the field.

2.9 (M*) Study design and detailed protocol (*Give a clear detailed protocol*) *Outline what approach is being used, why certain methods have been chosen and include statistical design.*
The approach used has been chosen to cause the least disruption to the subjects. The dogs will have a device attached their collar, but otherwise will be left to the control of the dogs owner.

3. SAMPLE AND SETTING

3.1 (M*) State numbers (or predicted numbers) to be used for study:
As many as practical in time frame, around 10-20

3.2 (M*) What species is the proposed sample and where is it located (e.g private land, university land, overseas, specific location)?
Private land likely, at the location of dog owning organisations. If that is not possible, an outdoor public space such a Southampton Common. If this takes place on private land permission will be obtained from the owning organisation.

3.3 (M*) Are endangered or protected species involved (intentionally or possibly inadvertently)
No

3.4 (M*) If so has relevant permission and licence been obtained?
No relevant license, permission will be obtained from dog owners before any data collection occurs.

3.4 (M*) Please list and upload licences required and name of person holding it.

UNIVERSITY OF Southampton

N/A

3.5 (M*) Which laws apply?
Animal Welfare Act 2006

3.6 (M*) What is the relationship between researchers and external funding organisation if any?
External organisation defines aims of the project.

4. RESEARCH PROCEDURES, INTERVENTIONS AND MEASUREMENTS

4.1 (M*) Give a brief account of the procedure as experienced by the participant *(Make clear who does what, how many times and in what order. Make clear the role of all assistants and collaborators.*
The researchers role in this is to attach the device to the dogs collar (this can also be done by the dog's owner if required), and to film the whole procedure. The dog's owner will be tasked with persuading the dogs to carry out the specified activities, as described in section 2.6. Once the collar is on the dog, and the researcher is filming, the dogs will be persuaded to carry out an activity "circuit" of all the required behaviours. This circuit will be repeated a number of times as long as time constraints allow this.

5. ANIMAL WELFARE

5.1 (M*) Will the animal be exposed to psychological or physical discomfort and/or distress?
Only to light discomfort of having a small device attached to their collar.

5.2 (M*) Explain how you intend to alleviate any psychological or physical discomfort and/or distress that may arise? (if applicable)
The device attached to the collar will be made as small and comfortable as possible.

5.3 Explain how you will care for any living organisms in the study (if applicable)?
Other than having the device attached to their collar, the Dogs will be left alone to be treated by the owner in the same manner as they normally would. As the device on the collar does not cause any suffering this should be sufficient.

5.4 What is the fate of the organisms at the end of the study?
They will be returned to their owners.

5.5 Have you undertaken any animal handling training (if applicable)?
No. This shouldn't be necessary as handling of dogs will be left to their owners.

5.6 (M*) How will data from this study be used? *Researchers should be aware of, and compliant with, the Data Protection policy of the University. You must be able to demonstrate this in respect of handling, storage and retention of data.*
The data will take the form of accelerometer and gyroscope readings with no identifiable components. These will be sent to a secure server and later downloaded on to personal

computers. The video component will be stored on password protected personal computers. The data will be used as described in section 2. Video will be deleted at the end of the project.

N.B. – Before you upload this document to your ERGO submission remember to:

1. Complete ALL mandatory sections in this form

2. Upload any letters of agreement referred, land permissions or licences required to undertake your study.

Participant Information Sheet

Study Title: Recognising Activities of African Wild Dogs using Machine Learning

Researcher: Rohan Bungre, Aran McConnell, Yue Kew, Charlie Steptoe
ERGO number: 61647

You are being invited to take part in the above research study. To help you decide whether you would like to take part or not, it is important that you understand why the research is being done and what it will involve. Please read the information below carefully and ask questions if anything is not clear or you would like more information before you decide to take part in this research. You may like to discuss it with others but it is up to you to decide whether or not to take part. If you are happy to participate you will be asked to sign a consent form.

What is the research about?
This research is part of a 4th year project for a group of Electronic Engineers and Computer Scientists.
This project involves working with an external company, Mafic Ltd, who have been approached by the makers of a TV documentary on African Wild Dogs. They want to use a device attached to the dog's collar, giving accelerometer readouts, to understand the behaviour of the dogs. Using the readouts from the device and a Artificial Intelligence framework, different common activities of the dogs should be recognised and aid in understanding the behaviour of these dogs. The final aim of this product is to be able to monitor and understand the behaviour of African Wild Dogs better, and in an nonintrusive way. This has applications in the production of documentaries and in conservation.
To develop the Artificial Intelligence framework for this solution, we are using similar data from large domestic dogs to train the framework, as it is not possible to collect such data from African Wild Dogs. The behaviour of such large dogs should be similar enough to get a model of a reasonable accuracy. To train this model we need to collect data from domestic dogs, whilst carrying out basic tasks such as running, trotting, sleeping, grooming, and eating. The dogs must be filmed for data collection, so the data can be labelling to each activity later on.

Why have I been asked to participate?
You have been asked to take part as you are an organisation or party that we believe may have access to dogs suitable for our research. You are also local to us, facilitating easier data collection.

What will happen to me if I take part?
A small device will be attached to the collar of your dog, which will record some basic movement data. A camera will also be set up to film the dog for the whole process. Then you will be asked to encourage your dog to carry out a circuit of "behaviours", consisting of;

- Walking
- Trotting
- Sleeping
- Grooming
- Eating
- Galloping
- Barking

This circuit will then be repeated if time allows. If any activity is difficult to encourage it can be skipped. This whole activity can take as long as the participant is happy with, but the more activities that can be recorded, the better.

The video recording of the dog is essential to this project, and will be used to label the data from the device on the dogs collar.

Are there any benefits in my taking part?

Taking part of this study will contribute data to a project which will develop a tool to help those producing wildlife documentaries, and conservationist carry out their work more effectively and with less disruption to African Wild Dogs.

Are there any risks involved?
No real risks identified.

What data will be collected?
Two main types of data will be collected. The device on the dogs collar will simply collected reads of the acceleration it experiences and the rotation it experiences. This will be stored securely on a server.
We will also take a video of the dog performing the activities. This video can be designed to only show the dog if required.

Will my participation be confidential?
Your participation and the information we collect about you during the course of the research will be kept strictly confidential.

Only members of the research team and responsible members of the University of Southampton may be given access to data about you for monitoring purposes and/or to carry out an audit of the study to ensure that the research is complying with applicable regulations. Individuals from regulatory authorities (people who check that we are carrying out the study correctly) may require access to your data. All of these people have a duty to keep your information, as a research participant, strictly confidential.

The study itself will collect no identifiable data about the participants, other than a video of the participant's dog. This video will be stored securely on password protected laptops, or the University servers on their OneDrive.
Consent forms will be scanned and kept electronically in the same location as the videos. The originals will then be destroyed.

Do I have to take part?
No, it is entirely up to you to decide whether or not to take part. If you decide you want to take part, you will need to sign a consent form to show you have agreed to take part.
If you do not wish to take part, communicate this verbally and do not sign the associated consent form.

What happens if I change my mind?
You have the right to change your mind and withdraw at any time without giving a reason and without your participant rights being affected.
You must communicate this verbally to a researched during the session, or by email before a session commences.

If you withdraw from the study, we will keep the information about you that we have already obtained for the purposes of achieving the objectives of the study only.

What will happen to the results of the research?
Your personal details will remain strictly confidential. Research findings made available in any reports or publications will not include information that can directly identify you without your specific consent.

Raw accelerometer data will be used to test and train a machine learning model, but contains no identifiable information at all. The project in general will be written up in an academic report as part of the completion of the Student's degrees. Video will be kept for the length of the project itself.

Where can I get more information?
Contact: Rohan Bungre (rsb1g17@soton.ac.uk),
Aran McConnell (ajm2g17@soton.ac.uk),
Yue Kew (ysk2a15@soton.ac.uk),
Charlie Steptoe (cs5n17@soton.ac.uk)

What happens if there is a problem?
If you have a concern about any aspect of this study, you should speak to the researchers who will do their best to answer your questions.
If you remain unhappy or have a complaint about any aspect of this study, please contact the University of Southampton Research Integrity and Governance Manager (023 8059 5058, rgoinfo@soton.ac.uk).

Contact: Rohan Bungre (rsb1g17@soton.ac.uk),
Aran McConnell (ajm2g17@soton.ac.uk),
Yue Kew (ysk2a15@soton.ac.uk),
Charlie Steptoe (cs5n17@soton.ac.uk)

Data Protection Privacy Notice
The University of Southampton conducts research to the highest standards of research integrity. As a publicly-funded organisation, the University has to ensure that it is in the public interest when we use personally-identifiable information about people who have agreed to take part in research. This means that when you agree to take part in a research study, we will use information about you in the ways needed, and for the purposes specified, to conduct and complete the research project. Under data protection law, 'Personal data' means any information that relates to and is capable of identifying a living individual. The University's data protection policy governing the use of personal data by the University can be found on its website (https://www.southampton.ac.uk/legalservices/what-we-do/data-protection-and-foi.page).

This Participant Information Sheet tells you what data will be collected for this project and whether this includes any personal data. Please ask the research team if you have any questions or are unclear what data is being collected about you.

Our privacy notice for research participants provides more information on how the University of Southampton collects and uses your personal data when you take part in one of our research projects and can be found at http://www.southampton.ac.uk/assets/sharepoint/intranet/ls/Public/Research%20and%20Integrity%20Privacy%20Notice/Privacy%20Notice%20for%20Research%20Participants.pdf

Any personal data we collect in this study will be used only for the purposes of carrying out our research and will be handled according to the University's policies in line with data protection law. If any personal data is used from which you can be identified directly, it will not be disclosed to anyone else without your consent unless the University of Southampton is required by law to disclose it.

Data protection law requires us to have a valid legal reason ('lawful basis') to process and use your Personal data. The lawful basis for processing personal information in this research study is for the performance of a task carried out in the public interest. Personal data collected for research will not be used for any other purpose.

For the purposes of data protection law, the University of Southampton is the 'Data Controller' for this study, which means that we are responsible for looking after your information and using it properly. The University of Southampton will keep identifiable information about you for 0 years after the study has finished after which time any link between you and your information will be removed.

To safeguard your rights, we will use the minimum personal data necessary to achieve our research study objectives. Your data protection rights – such as to access, change, or transfer such information - may be limited, however, in order for the research output to be reliable and accurate. The University will not do anything with your personal data that you would not reasonably expect.

If you have any questions about how your personal data is used, or wish to exercise any of your rights, please consult the University's data protection webpage (https://www.southampton.ac.uk/legalservices/what-we-do/data-protection-and-foi.page) where you can make a request using our online form. If you need further assistance, please contact the University's Data Protection Officer (data.protection@soton.ac.uk).

Thank you for reading this information sheet and considering taking part in our study.

UNIVERSITY OF Southampton

CONSENT FORM

Study title: Recognising Activities of African Wild Dogs using Machine Learning

Researcher name: Rohan Bungre, Aran McConnell, Yue Kew, Charlie Steptoe
ERGO number: 61647

Please initial the box(es) if you agree with the statement(s):

I have read and understood the information sheet and have had the opportunity to ask questions about the study.	
I agree to take part in this research project and agree for my data to be used for the purpose of this study.	
I understand my participation is voluntary and I may withdraw (at any time) for any reason without my participation rights being affected.	
I understand that taking part in the study involves video recording for the purposes set out in the participation information sheet.	

Name of participant (print name)……………………………………………………………………………

Signature of participant…………………………………………………………………………………….

Date…………………………………………………………………………………….. ………………….

Name of researcher (print name)……………………………………………………………………………

Signature of researcher …………………………………………………………………………………….

Date………..

UNIVERSITY OF Southampton

Health & Safety Risk Assessment Template

Work task / activity	Recognising Activities of African Wild Dogs using Machine Learning

Assessor	Aran McConnell	**Responsible Manager**	Srinandan Dasmahapatra	**Date**	29/10/2020

Faculty / Service	Electronics and Computer Science	**Academic Unit / Team**	GDP Group 16	**Location**	Private land of organi-sations or public parks.

Brief description of task / activity	Collecting data from dogs by attaching a small device to their collar and filming them carry out standard behaviours such as Running, Trotting, Sleeping, Grooming, Eating, Galloping, and Barking.

Reasonably foreseeable hazards	Inherent risk	Controls	Residual risk
COVID-19 Transmission during procedure	Med	Researchers and participants will wear a mask during the procedure, and social distancing will be respected. Anything that multiple people might touch will be disinfected and hand sanitiser will be used before use. The groups and individuals that we will be working with will be made aware of this COVID protocol.	Low
COVID-19 Transmission during Travel	Low Med High	We will try to use safer means of transport such as personal cars or cycling. If we are required to use public transport we will wear masks, use hand sanitiser and practice social distancing.	Low Med High
	Low Med High		Low Med High

Risk assessment checklist

- ☑ Risk assessments must be ‘suitable and sufficient’, that is, should cover all relevant issues and include enough detail.
- ☑ Work tasks & activities should be risk assessed, and not, as such, substances (but rather use of substances), or equipment (but rather use of equipment), or locations (but rather activities therein), or people (but rather what they do).
- ☑ This template is for ‘general’ risk assessments, and is suitable for most hazards, but certain hazards require additional regulatory and technical detail, such as ionising radiations, biological agents, genetic modification, noise, hazardous chemicals, etc.
- ☑ Risk assessments can be generic, provided they are ‘suitable and sufficient’, that is, identify all reasonably foreseeable hazards, meaningfully estimate risk, and delineate effective controls.
- ☑ ‘Hazards’ are things with the potential to cause harm.
- ☑ The qualification ‘reasonably foreseeable’ is applied to hazards to indicate that far-fetched, improbable hazards need not be considered, and also neither need the obvious hazards of everyday life.
- ☑ ‘Inherent’ risk is that before controls are applied.
- ☑ Risk should be estimated using the matrix on the next page.
- ☑ ‘Controls’ are measures to eliminate or reduce risk.
- ☑ ‘Residual’ risk is that after controls are applied.
- ☑ The assessment should consider:
  - ☑ Any competency, training and supervision that may be necessary.
  - ☑ Reasonably foreseeable emergencies, and include suitable contingency plans.
  - ☑ Any health surveillance that may be necessary.
  - ☑ Any waste management or other environmental issues that may arise.
- ☑ The declaration at the end of the assessment must be signed by the responsible manager, principal investigator, project leader, etc.

UNIVERSITY OF Southampton

Risk estimation matrix

High risk – requires controls to reduce risk before activity / task can commence (or continue).

Medium risk – requires controls to reduce risk as much and as soon as is reasonably practicable.

Low risk – all risk should be reduced to this tolerable level, so far as is reasonably practicable.

Reasonably foreseeable consequence severity Likelihood [3] of consequence	**Minor** superficial injury; or slight and temporary health effect; or trivial damage to equipment / building; or minimal disruption to work activities	**Moderate** significant injury or illness [1]; or temporary minor disability; or minor damage to equipment / building; or slight disruption to work activities	**Major** serious injury or illness [2]; or significant or permanent disability; or significant damage to equipment / building; or significant disruption to work activities	**Critical** fatal injury or illness; or substantial and permanent disability; or severe damage to equipment / building; or extensive disruption to work activities	**Catastrophic** fatal injury or illness for multiple persons; or enormous damage to equipment / building; or disastrous disruption to work activities
Almost certain very high probability, at or approaching 100%	high risk	high risk	high risk	high risk	high risk
Likely high probability, 1 in 10 chance or higher, once in two weeks or longer for activities on a daily basis	medium risk	high risk	high risk	high risk	high risk
Possible significant probability, 1 in 100 chance or higher, once in six months or longer for activities on a daily basis	low risk	medium risk	high risk	high risk	high risk
Unlikely low probability, 1 in 1,000 chance or higher, once in four years or longer for activities on a daily basis	low risk	low risk	medium risk	high risk	high risk
Rare very low probability, 1 in 10,000 chance or higher, once in a decade or longer for activities on a daily basis	low risk	low risk	low risk	medium risk	high risk
Almost never extremely low probability, less than 1 in 100,000 chance, once in a century or longer for activities on a daily basis	low risk	low risk	low risk	low risk	medium risk

[1] 'Significant injury' could include, for example, laceration, burn, concussion, serious sprain, minor fracture, etc.
'Significant illness' could include, for example, dermatitis, minor work-related musculoskeletal conditions, partial hearing loss, etc.

[2] 'Serious injury' could include fracture or dislocation (other than fingers, thumbs or toes), amputation, loss of sight, penetration or burn to eye, serious electric shock, asphyxia, or any injury leading to unconsciousness or requiring resuscitation or admittance to hospital for more than 24 hours. 'Serious illness' could include, for example, requiring medical treatment after chemical or biological or radiological exposure, severe musculoskeletal conditions, severe dermatitis, asthma, etc.

[3] For likelihoods in between the listed values, use the higher likelihood to estimate risk.

Chapter B

Command Line Tool Guide

File Structure

```
Directory
├─ environment.yml
├─ main.py
├─ label.py
├─ features.py
├─ train.py
├─ classify.py
├─ classifier.pkl
├─ featureLevel.pkl
├─ environment.yml
└─ Data
   ├─ "Session Date"
   │  ├─ raw.pkl
   │  ├─ labelled.pkl
   │  ├─ dataFeatured.pkl
   │  └─ labels.csv
   └─ "Session Date"
      └─ etc...
```

Tool Usage

General Usage

If using the conda environment, the following command can be used to set up the environment:

```
> conda env create -f environment.yml
```

The tool can be run from the command line using:

```
> python main.py [options]
```

Labelling

The tool can be used to label all training *raw.pkl* data files, with their corresponding *labels.csv* files using:

```
> python main.py -label
```

This will save the labelled data in each subdirectory as *labelled.pkl*

Adding Features

The tool can then be used to add different "levels" of features to the training data:

- Level 1 adds means and variance features
- Level 2 adds reciprocals of base features
- Level 3 adds Fourier variables
- Level 4 adds Wavelet function features Each level adds features from all previous levels.

```
> python main.py -features [1,2,3,4]
```

This will save the featured data as *dataFeatured.pkl* in the corresponding subdirectory.

Training a Classifier

The tool can then train a chosen classifier using the current featured training data.

```
> python main.py --train [RandomForest,ExtraTrees,AdaBoost]
```

This will save the trained classifier as *classifier.pkl* to be used for classification.

Processing unlabelled data

Before unlabelled raw data can be classified by the tool, it must have the same level of features added to it as was used to train the classifier. This will add the required features to the file specified by "filepath".

```
> python main.py --process [filepath]
```

The featured data is then stored as *"filepath"_featured.pkl*.

Classify unlabelled data

To then classify unlabelled data using the tool:

```
> python main.py --classify [filepath]
```

By pointing this option to the filepath of the "_featured" unlablled data, it will export a *"filepath".csv* containing the timestamps in the data, and the corresponding labels as predicted by the classifier.

Chapter C

Gantt Charts

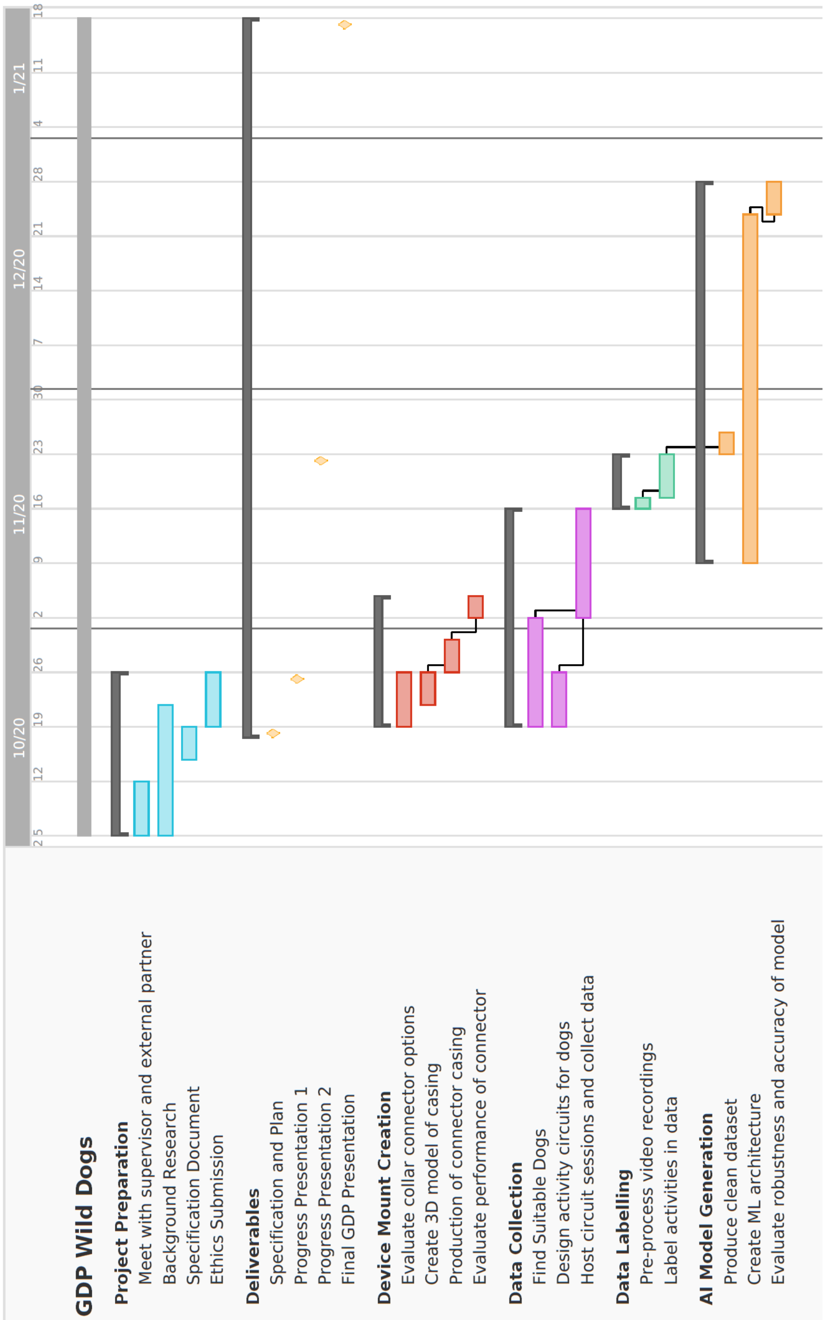


Figure C.1: Initial Gantt Chart

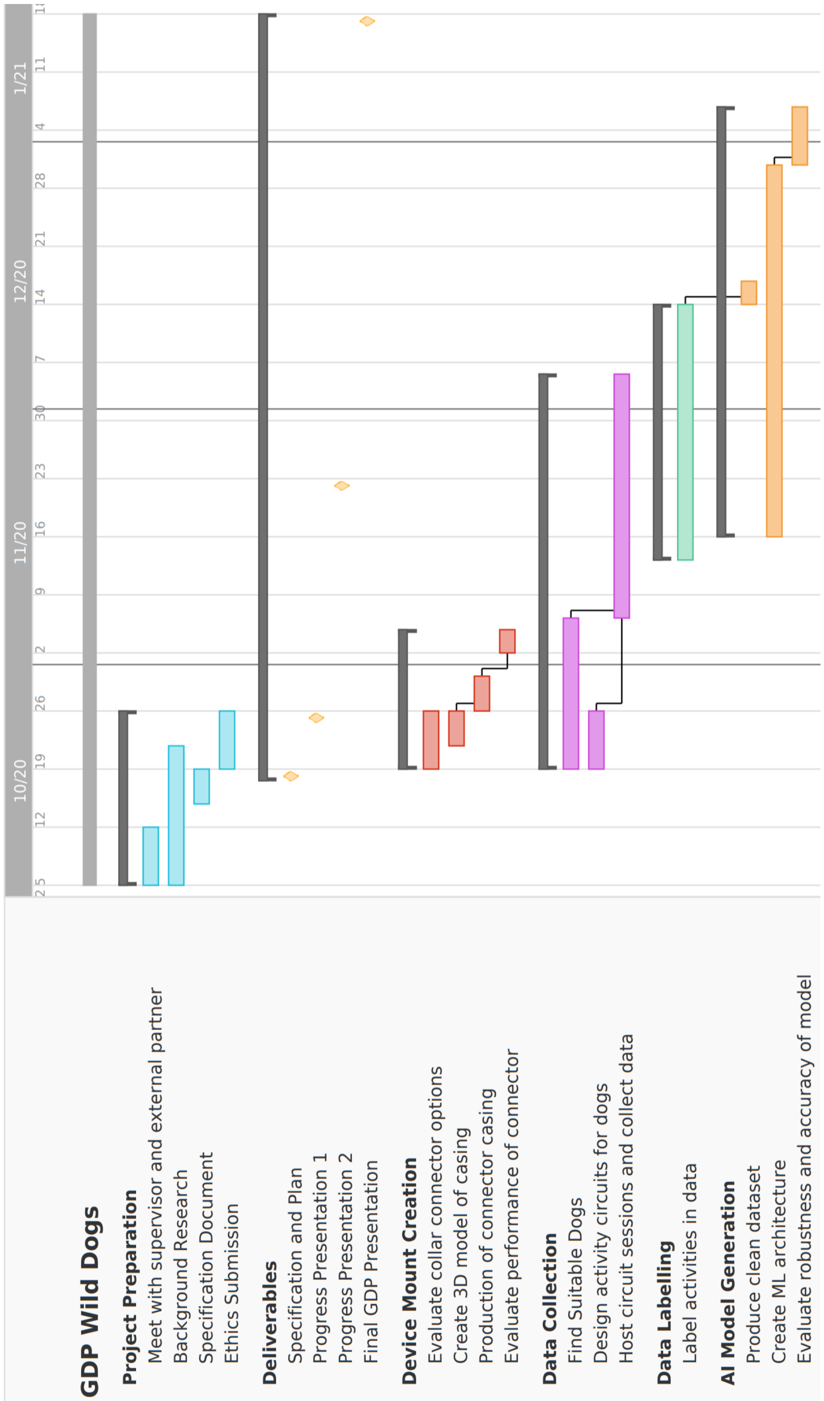


Figure C.2: Revised Gantt Chart

Chapter D

Budgeting

All GDP teams are given £400 of budget. The team have spent a total of £148.532, with £251.468 left in the budget. Item 1 and 2 are purchased through ECS Requisition site. A "Travel and Expenses Claim Form" have been submitted for the remaining items for reimbursement and is in the stage of awaiting approval from ECS and/or University Finance Team. A detailed list of expenditures is shown in Table D.1.

Item(s)/ Journey(s)	**Quantity**	**Supplier**	**Stock Code**	**Cost (£)**	**Total**
1080p KONIG Action Camera	1	OneCall	PY32098	54.564	54.564
SD Card	1	OneCall	MD01001	10.188	10.188
Taxi Journeys (round trip from home to arranged meet up point with volunteers)	2	RadioTaxi	-	29.60+28.20	57.800
Dog Collar (Large)	2	Argos	-	12.99	25.98
				Final Cost	**148.532**
				Total Budget	400.000
				Remaining Budget	**251.468**

Table D.1: Project Expenditures.

Chapter E

Activity Prediction Plots

These plots demonstrate the labelling of a typical session recorded using the IMU device attached to the dog, with supervised learning-enabled classifications plotted as blue points, against the ground truth shown in orange.

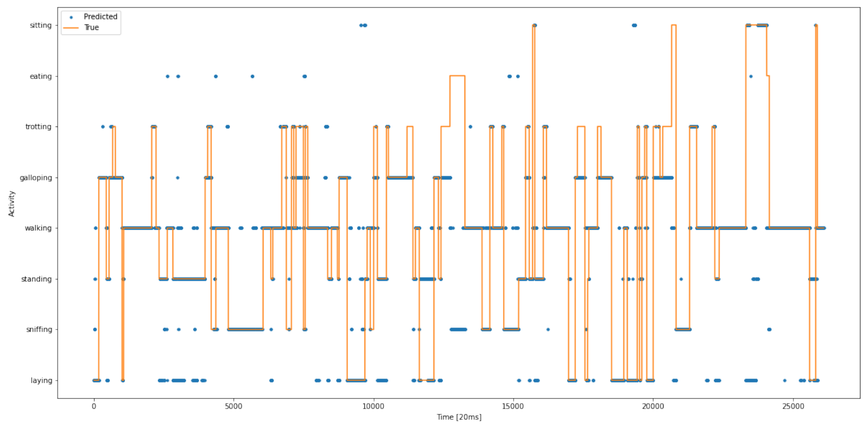


Figure E.1: XGBClassifier Activity Recognition Plot

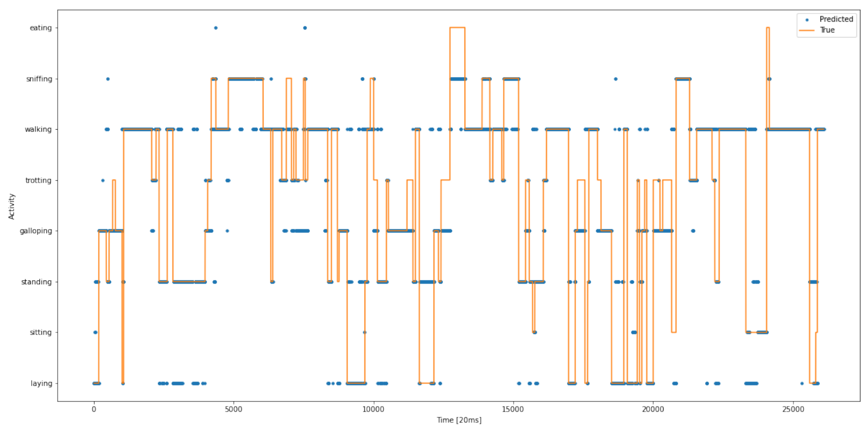


Figure E.2: RandomForestClassifier Activity Recognition Plot

Chapter F

Archive File Structure

```
Archive
├─ A - CAD
│  └─ 3D Models
│     └─ Mount.stl
├─ B - CLI Tool
│  ├─ main.py
│  ├─ label.py
│  ├─ features.py
│  ├─ train.py
│  ├─ classify.py
│  ├─ classifier.pkl
│  ├─ featureLevel.pkl
│  ├─ unlabelledTestData.pkl
│  ├─ unlabelledTestData_featured.pkl
│  ├─ unlabelledTestData_featured.csv
│  └─ Data (Copy for use with CLI Tool)
├─ C - Data
│  ├─ 2020-11-05
│  │  ├─ xyz.pkl
│  │  ├─ cut.pkl
│  │  ├─ cutoff.txt
│  │  ├─ dataFeatured.pkl
│  │  ├─ labelled.pkl
│  │  │
```

```
│    │    ├─ labels.csv
│    │    └─ raw.pkl
│    ├─ 2020-11-16
│    │    └─ etc...
│    ├─ 2020-11-23
│    │    └─ etc...
│    ├─ 2020-11-26
│    │    └─ etc...
│    ├─ 2020-11-27
│    │    └─ etc...
│    ├─ 2020-11-28
│    │    └─ etc...
│    └─ 2020-12-03
│         └─ etc...
├─ D - Ethics
│    ├─ Consent Form.pdf
│    ├─ Ethics Form.pdf
│    ├─ PIS.pdf
│    ├─ RA.pdf
│    └─ Filled Consent Forms
│         ├─ participant1.pdf
│         └─ participant2.pdf
├─ E - Notebooks
│    ├─ Basic Value Data Distribution
│    │    ├─ PartDataAnalytic1.png
│    │    ├─ PartDataAnalytic2.png
│    │    └─ PartDataAnalytic3.png
│    ├─ Feature Importance Plots
│    │    ├─ Random Forest Feature Importance.png
```

```
│   │   └─ XGBoost Feature Importance.png
│   ├─ Kmeans Results
│   │   ├─ Fourier.odt
│   │   ├─ Mean-Variance.odt
│   │   ├─ Raw Data.odt
│   │   └─ Reciprocals.odt
│   ├─ Prediction Plots
│   │   ├─ RF Activity Recognition Plot.png
│   │   └─ XGBoost Activity Recognition Plot.png
│   ├─ Important_Reduced_Features.txt
│   ├─ Session-Wise Confusion Matrices
│   │   ├─ Random Forest
│   │   │   ├─ Confusion 1.png
│   │   │   ├─ Confusion 2.png
│   │   │   ├─ Confusion 3.png
│   │   │   ├─ Confusion 4.png
│   │   │   └─ Confusion 5.png
│   │   └─ XGBoost
│   │       ├─ Confusion 1.png
│   │       ├─ Confusion 2.png
│   │       ├─ Confusion 3.png
│   │       ├─ Confusion 4.png
│   │       └─ Confusion 5.png
│   ├─ Heatmaps
│   │   ├─ Wavelet Transform.pdf
│   │   └─ Fourier Transform.pdf
│   ├─ aran_semisupervised.ipynb
```

- aran_template.ipynb
- aran_unsupervised.ipynb
- charlie.ipynb
- esther.ipynb
- esther_dl.ipynb
- esther_signalprocessing.ipynb
- rohan_init_supervised.ipynb
- rohan_unsupervised.ipynb

- F - Team Management
  - Gantt Charts
    - Gantt Initial.png
    - Gantt Revised.png
  - Team Roles
    - AranMcConnell - DISC.pdf
    - AranMcConnell - Team Roles.png
    - CharlieSteptoe - DISC.pdf
    - CharlieSteptoe - Team Roles.png
    - RohanBungre - DISC.pdf
    - RohanBungre - Team Roles.pdf
    - YueKew - DISC.pdf
    - YueKew - Team Roles.pdf

Chapter G

Archive Index

Index	Item	Description	Author	Date
A- 1	\A-CAD\3D Models	The complete and revised 3D-printable design file for the IMU device collarmount, in .stl file format for ease of alteration.	C.S	19-01-21
B- 1-5	\B-CLITool\main.py \B-CLITool\label.py \B-CLI\features.py \B-CLI\train.py \B-CLI\classify.py	The source files for the Command Line Tool. These files use code from Notebooks (Archive E), and modify it to work as a tool. The other files in this directory are generated by the tool and are included here to facilitate ease of use of the tool.	A.M	19-01-21
C	\C-Data	A collection of the device data collected from all of our sessions,along with labels from manual labelling.	A.M C.S R.B Y.K	19-01-21
D- 1-4	\D-Ethics\Consent Form \D-Ethics\Ethics Form \D-Ethics\PIS \D-Ethics\RA	Documents from the ethics proposalincluding the Ethics form, Consent form, Participant Information Sheet, and Risk Assessment.	A.M	19-11-20

E- 1	\E-Notebooks\ BasicValue Data Distribution	Data distribution tables for data collected over each session. Includes metrics for the 6 basic IMU variables: acc x, acc y, acc z, gyro x, gyro y, gyro z.	C.S	19-01-21
E- 2	\E-Notebooks\ Feature Importance Plots	Bar charts representing feature importance metrics over all 57 input variables, used in supervised learning models.	C.S	19-01-21
E- 3	\E-Notebooks\ Heatmaps	Fast Fourier Transform values over total data collected in all sessions displayed as a green-yellow heatmap (Green values represent minima, with yellow representing maxima). All 6 basic IMU values and 3 bi-dependent features are analysed for each activity classification. Discrete Wavelet Transform values over total data collected in all sessions displayed as a blue-red heatmap (blue values represent minima, with red representing maxima). The 6 basic IMU values are analysed for each activity classification.	C.S	19-01-21
E- 4	\E-Notebooks\ KMeans Results	Heatmaps and numeric results of applying KMeans to different datasets	A.M	01-01-21

E- 5	\E-Notebooks\ Prediction Plots	A pair of plots representing the activity classification across an entire session, as predicted by XGBoost and Random Forest Classifiers. Model-predicted activity classification is plotted in blue, with truth values plotted in orange.	C.S	19-01-21
E- 6	\E-Notebooks\ Important _Reduced_ Features	A complete list of the reduced features used to test scalability of XGBoost and Random Forest classifiers. The list, alongside the feature importance thresholds, by which these features are decided, is available in the attached appendix .zip file.	C.S	19-01-21
E- 7	\E-Notebooks\ Session-Wise Confusion Matrices	Confusion matrices heatmaps showing the trained activity recognition over all sessions, using sessions not tested on as training data.	C.S	19-01-21
E- 8-16	\E-Notebooks\ All .ipynb	Jupyter Notebook Files for all team members.	A.M C.S R.B Y.K	19-01-21
F- 1	\F-Team Management\ Gantt Charts	Both the initial and revised Gantt charts, shown in Appendix C.	A.M	14-12-20
F- 2	\F-Team Management\ Team Roles	Team roles and DISC test results for each team member.	A.M C.S R.B Y.K	19-10-20