Author:
Montout, Axel X

Title:
Prediction of poor health in small ruminants and companion animals with accelerometers and machine learning

Prediction of poor health in small ruminants and companion animals with accelerometers and machine learning

By

AXEL XAVIER MONTOUT

Bristol Veterinary https://www.overleaf.com/project/5f8426fd576a7a00017668e5School
UNIVERSITY OF BRISTOL

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of DOCTOR OF PHILOSOPHY in the Bristol Veterinary School.

MARCH 2023

Word count: 29481

Abstract

Global warming is one of the biggest challenge of our times, and significant efforts are being undertaken by academics, industries and other actors to tackle the problem. In the agricultural field precision farming is part of the solution to environmental sustainability and has been researched increasingly in recent years. Indeed, it has the potential to effectively increase livestock yield and decrease production carbon footprint while maintaining welfare. The thesis begins with a review of developments in automated animal monitoring and then moves on to a case study of a health monitoring system for small-ruminant in South Africa. As a demonstration and validation of the potential use case of the system, the method we propose is then applied to another study which aims to study feline health. Lower and Middle Income countries will be strongly affected by the changing climate and its impacts. We devise our method based on two South African small scale sheep and goat farms where assessment of the health status of individual animals is a key step in the timely and targeted treatment of infections, which is critical in the fight against anthelmintic and antimicrobial resistance. The FAMACHA scoring system has been used successfully to detect anaemia caused by infection with the parasitic nematode *Haemonchus contortus* in small ruminants and is an effective way to identify individuals in need of treatment. However, assessing FAMACHA is labour-intensive and costly as individuals must be manually examined at frequent intervals. Here, we used accelerometers to measure the individual activity of extensively grazed small ruminants exposed to natural *Haemonchus contortus* worm infection in southern Africa over long time scales (13+ months). When combined with machine learning for missing data imputation and classification, we find that this activity data can predict poorer health as well as those individuals that respond to treatment, with precision up to 80%. We demonstrate that these classifiers remain robust over time. Interpretation of trained classifiers reveals that poorer health can be predicted mainly by the night-time activity levels in the sheep. Our study reveals behavioural patterns across two small ruminant species, which low-cost biologgers can exploit to detect subtle changes in animal health and enable timely and targeted intervention. This has real potential to improve economic outcomes and animal welfare as well as limit the use of anthelmintic drugs and diminish pressures on anthelmintic resistance in both commercial and resource-poor communal farming. The validation of the proposed techniques with a different study group will be discussed in the latter part of the thesis. We used the accelerometry data of indoor cats equipped with wearable accelerometers in conjunction with their health status to detect signs of degenerative joint disease, and adapted our machine-learning pipeline to analyse bursts of high activity in the cats. We were able to classify high-activity events with precision up to 70% despite the relatively small dataset adding further evidence to the viability of animal health monitoring with accelerometers.

Dedication and acknowledgements

My desire to undertake challenges with potential real-world impact led me to start this PhD 4 years ago after spending some time working for various industries. To everybody who supported me during this endeavour, thank you.

I wish to, particularly thanks my supervisor Professor Andrew Dowsey who has given me guidance and support throughout the up and downs of this journey, from my very first middle school teacher to my early years as an academic researcher he has been the kindest and most knowledgeable professor I had. It has been a pleasure to work alongside him and my other supervisors Tilo Burghardt and Christos Ioannou.

I wish to also thank my colleague Ranjeet Bhamber who was always able to fix bugs in my code thanks to his incredible coding experience. My good friend and colleague Lucy Vass who started her PhD the same day I started mine for introducing me to the UK customs and guiding me around the City and the University of Bristol.

I must thank senior veterinarian Jan A. Van Wyk who made this project possible by collecting the data for this thesis in 2012. Despite not knowing the usability of the data he had the insight to think that eventually, someone could analyse it in depth. I had the pleasure and luck to have him as my guide during my field trip in South Africa where I met his associate Doreen Z. Ndlovu and resource-poor farmers who were all very welcoming and helped me to understand the on-the-field implications and challenges of my research.

In addition to my main supervisory team, I wish to thank Eric R. Morgan for his collaboration with this project.

Furthermore, I wish to thank Professor Emily Blackwell and Professor Melanie Hezzell for making Chapter 5 (companion animal) possible and for their valuable veterinary knowledge to guide me along the way.

Finally, thank you to all my friends and family.

This research has been funded by the University of Bristol School of Veterinary Sciences and the Jean Golding Institute.

Author's declaration

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

SIGNED: ..... DATE: .....

Table of Contents

List of Abbreviations

LMIC	Low and middle income countries
RP	Resource poor
ML	Machine learning
DB	Decision boundary
KNN	K nearest neighbour
LREG	Logistic regression
DTREE	Decision tree
SVM	Support vector machine
CNN	Convolutional Neural Networks
FFT	Fast fourier transform
CWT	Continuous wavelet transform
DSP	Digital signal processing
ADC	Analogue to digital converter
MEMS	Micro electro-mechanical system
ECG	Electrocardiogram
GPRS	General Package Radio Services
HDF5	Hierarchical Data Format 5
NN	Neural Network
RNN	Recurrent neural network
FC	Fully connected

LSTM	Long short term memory
GRU	Gated Recurrent Units
RMSE	Root mean square error
AUC	Area under the curve
ROC	Receiver operating characteristic
FP	False positive
TP	True positive
FPR	False positive rate
TPR	True positive rate
PLS	Partial Least Square
QN	Quotient normalisation
M-RNN	Multi recurrent neural network
GAIN	Generative adversarial imputation nets
LI	Linear interpolation
RBF	Radial basis function
FAMACHA	Faffa Malan chart
FMPI	Feline Musculoskeletal Pain Index
DJD	Degenerative joint disease
LOOCV	Leave-one-out cross-validation
EDA	Exploratory data analysis
PCA	Principal component analysis
NPL	Natural language processing

List of Tables

Table **Page**

List of Figures

Figure **Page**

CHAPTER 1

INTRODUCTION

Accelerometers are sensors that measure the acceleration of an object, animal, or person. They are widely employed in wearable devices, such as smartphones and fitness trackers, to track physical activity and movement. In recent years, machine learning (ML) has gained increasing popularity as a subset of artificial intelligence, enabling algorithms to recognise patterns in data. Together, accelerometers and ML are transforming wellbeing and health tracking in humans, facilitating accurate and personalised monitoring of physical activity, sleep patterns, and other health-related metrics. For instance, accelerometers can detect falls, a crucial feature for elderly individuals or those with mobility impairments. ML algorithms can be trained to recognise patterns in movement data and identify when a fall occurs, triggering alerts for emergency services or caregivers [96] [116] [108]. Accelerometers are also valuable in monitoring motor symptoms in Parkinson's disease patients, with ML algorithms analyzing this data to identify patterns in symptom progression and response to treatment [80]. As these technologies continue to advance, we anticipate even more sophisticated applications for health and wellbeing monitoring in the future.

There is a growing interest in applying accelerometers and ML to animal health, welfare, and food security. They can revolutionise these fields, including animal behavior monitoring, where accelerometers can be attached to animals to monitor their activity levels and movements. This allows accurate monitoring of behavior and identification of potential health issues or welfare concerns. ML algorithms can analyse this data to recognise patterns in animal behavior, enabling targeted interventions and management practices [45]. For disease detection and monitoring in animals, accelerometers can be used to identify potential signs of disease earlier than traditional diagnostic methods. By monitoring changes in activity levels and behavior patterns, machine learning algorithms allow for timely interventions and treatment [93]. They can also be employed in precision livestock farming practices, where individual livestock are

monitored and managed based on their specific needs and behavior patterns. This can enhance animal welfare and reduce environmental impacts. By improving animal health and welfare monitoring, accelerometers and ML contribute to ensuring food security by reducing the risk of disease outbreaks in livestock populations and improving the efficiency of food production [131]. Now is the opportune time to translate this approach for animal health, welfare, and food security due to the rapid advancements in sensor technology and ML algorithms, making these technologies more affordable and accessible.

Accelerometer data can be analysed using various techniques, including digital signal processing (DSP) and machine learning (ML). DSP involves applying mathematical algorithms to the raw accelerometer data to extract relevant features such as frequency, amplitude, and phase. These features can then be used to identify specific types of motion, such as walking, running, or jumping. DSP techniques have been widely used in activity-tracking applications, such as pedometers and fitness trackers, to estimate the number of steps taken, distance travelled, and calories burned. In recent years, the use of ML techniques for accelerometer data analysis has become increasingly popular. ML algorithms can automatically learn and extract complex patterns from large datasets, without the need for manual feature extraction. This makes ML particularly useful for applications where the underlying patterns may be difficult to characterise using DSP techniques alone. For example, ML algorithms have been used to classify different types of physical activities, such as walking, cycling, and climbing stairs, based on accelerometer data. Human and animal activity tracking is one area where accelerometers and ML have been widely used. For example, wearable devices with built-in accelerometers have been used to monitor the physical activity levels of individuals and to track their sleep patterns. These data can be used to identify risk factors for chronic diseases such as obesity and diabetes and to develop personalised health interventions. In animal activity tracking, accelerometers have been used to study the behaviour of a wide range of species, from birds and bats to elephants and whales. ML algorithms have been used to analyse the accelerometer data and identify specific behaviours, such as flying, diving, or feeding. This information can be used to better understand the ecology and behaviour of the animals and to develop conservation strategies. In summary, accelerometers are sensors that measure acceleration and output data in the form of electrical signals. DSP and ML techniques can be used to analyse accelerometer data and extract useful information about motion, behaviour, and activity levels. Accelerometers and ML are widely used in human and animal activity tracking applications, with the potential for a wide range of other applications in industry, healthcare, and research.

1.1 Livestock farming

Agriculture is one of the building blocks of modern human civilisation. From the early stages of the development of our specie, cultivating and controlling access to food has historically been a

fundamental priority in the success of a nation; a regular diet that includes meat has been key to producing stronger and healthier people. The first sign of livestock farming can be traced over 10,000 years ago when pigs, sheep and cattle were domesticated. We can define "livestock farming" as the culture of animals for human consumption of said animal or any of their consumable derivative products. Depending on the region of the world the raised animals may include, dairy cattle, pigs, sheep, goats, horses, mules, buffalo and camels. The raising of birds for meat and egg consumption is classified as Poultry farming [63](Fig. 1.1). In modern times, the growing world population alongside urbanisation and income growth are some of the main factors which explain why the consumption of meat has increased in the developing and developed world alike. The developing world experienced a 70 million metric tons increase in the demand for meat between the beginning of the 1970s and the mid-1990s which was a 3 fold increase compared to the demand in the developed world [29]. The demand is closely tied to the human population growth, it is expected to observe a continued increase in meat demand. However, livestock production has a significant impact on the environment, contributing to greenhouse gas emissions, deforestation, and water pollution [46]. Achieving the goal of 'net-zero' emissions from livestock production is extremely difficult due to the complex nature of the industry and the high demand for meat. While reducing meat consumption in the developed world is necessary, it may not be feasible in many parts of the world where alternative sources of protein are limited. Instead, efforts should focus on promoting more efficient and sustainable meat production in the developing world through better animal management practices, reducing waste, and improving feed quality. This can help to reduce the environmental impact of livestock production while ensuring that people in these regions have access to important sources of protein. Such growth will continue to affect the lives of many and present numerous challenges for livestock farming.

From old methods in livestock management to modern precision farming technologies, the era greatly influences preferred practices. This is particularly notable in the developed world, where accessing the latest technologies is more affordable. For example, in the developed world the efficiency of the production of beef has drastically improved thanks to the use of scientific methods to evaluate optimal housing and equipment management [19] [82] [75]. Offspring production also benefits while various fertility checks, preventative medicine, protection against insects and parasites, and diet monitoring are some of the methods that have allowed better yield [127]. Analysis of the weather is also taken into account as calving events are arranged to correspond to the seasons which offer higher-quality pastures. Furthermore, artificial insemination permits the production of large quantities of genetically fit animals [63]. In recent years precision farming technologies such as accelerometry monitoring with animal-mounted sensors and automatic video monitoring have emerged and can also contribute to better production outcomes. For example, Qiao et al [119] successfully developed an accurate camera-based cattle identification software that can facilitate the non-intrusive monitoring of cattle applicable to precision farming and surveillance for automated productivity, health and welfare monitoring, and veterinary

research such as behavioural analysis, disease outbreak tracing, and more. Similarly, Sheep and Goat farming has seen the advent of sensor-based monitoring systems for animal behaviour monitoring. Pork production in an industrialised nation is often mechanised which reduces the need for expensive labour. It is commonplace for large-scale commercial producers to use indoor slotted floor farms with various environmental control such as air temperature and self-feeders.

While the demand in the developing world is growing at a fast pace, in practice the use of state-of-the-art precision farming techniques is generally not feasible unless new affordable technologies were to be developed, which is what this thesis aims to explore. This contrast explains the different sets of challenges faced by developed and developing countries. Indeed, while the developing countries still face high livestock mortality rates [39] due to poor health management practices and limited access to modern disease control, which often lead to serious problems such as antibiotic resistance, the developed world faces much lower moralities rates not only because of better overall practices but also thanks to the assistance of advanced technologies and medical treatment.

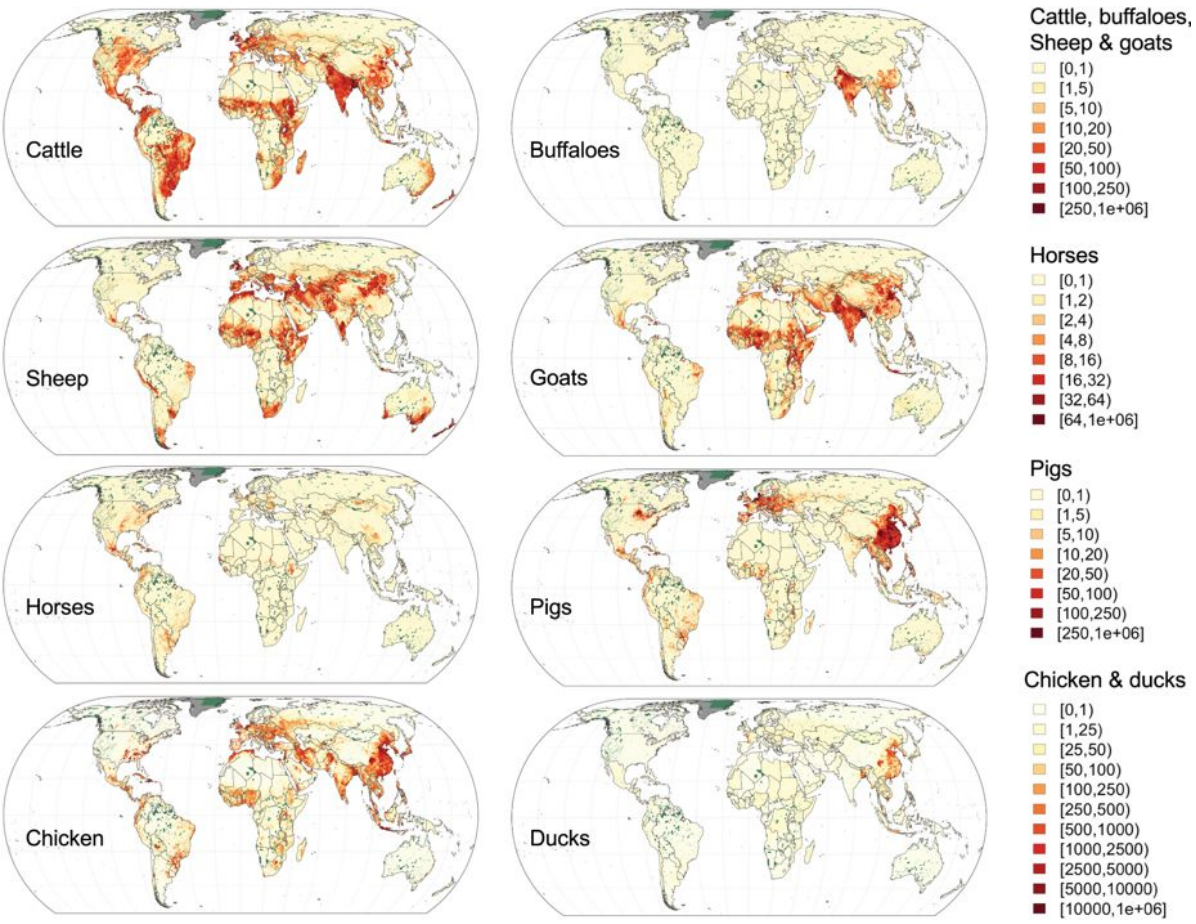


Figure 1.1: **Global distribution data for cattle, buffaloes, horses, sheep, goats, pigs, chickens and ducks in 2010.** World map of the geographic distribution of different farm animals, source [48].

1.2 Environmental sustainability and net-zero

Climate change is a significant challenge for the livestock farming industry as it contributes to greenhouse gas emissions that can exacerbate the effects of global warming. Livestock farming is a significant contributor to greenhouse gas emissions, accounting for around 18% of all anthropogenic greenhouse gas emissions worldwide [107]. The primary source of greenhouse gas emissions from livestock farming is enteric fermentation, which occurs during digestion in the stomachs of ruminant animals such as cows, sheep, and goats. Additionally, manure management, feed production, and transportation of livestock and feed also contribute to greenhouse gas emissions. Climate change can also have indirect effects on livestock farming, including changes in weather patterns, increased frequency of extreme weather events, and changes in the availability of water and other resources. These changes can impact animal health and welfare, and also affect the productivity of farms and the availability and cost of feed and other inputs. To address

the challenge of climate change in the context of livestock farming, various strategies can be employed, such as improving animal nutrition and genetics [100], reducing waste and improving manure management, and using renewable energy sources. Additionally, there is growing interest in developing alternative protein sources, such as plant-based and cell-based meats, which have a lower environmental impact than traditional livestock farming [139]. Flexitarian diets are plant-based diets that emphasise the consumption of whole grains, fruits, vegetables, and legumes, while allowing for limited amounts of animal products. While such diets can provide health benefits and help reduce the environmental impact of food production [98], it may not be realistic or feasible in all parts of the world.

In low and middle income countries (LMICs) and regions with limited resources, there may be several factors that make it difficult to transition to a flexitarian diet [135]. One of the primary reasons is that the climate and soil may not be conducive to growing a diverse range of crops. In some areas, the soil may be too acidic, nutrient-poor, or contaminated with pollutants, making it difficult to grow crops that are essential for a plant-based diet. Furthermore, certain regions may experience droughts or flooding, making it difficult to consistently produce crops. In addition, access to fresh produce and other plant-based foods may be limited in LMICs. This can be due to a lack of infrastructure, such as refrigeration or transportation, or limited access to markets that sell fresh produce. In some areas, food insecurity may also be a major concern, which can make it difficult to prioritise a plant-based diet. Finally, cultural and societal factors may also play a role in making it difficult to transition to a flexitarian diet. In some regions, meat consumption is an important part of cultural traditions, and it may be challenging to convince people to shift their dietary habits. Additionally, there may be a lack of education about the benefits of plant-based diets, and people may not have the resources or knowledge to prepare and cook plant-based meals. In conclusion, while a flexitarian diet can be beneficial for health and the environment, it may not be a realistic option for everyone. In LMICs and regions with limited resources, factors such as soil quality, access to fresh produce, food insecurity, and cultural and societal factors may make it difficult to transition to a plant-based diet.

1.3 Subsistence and resource-poor farming

Delgado et al. (1999c) popularised the term "Livestock revolution" which defines the accelerated growth in the demand for livestock-based products (for example, meat, milk, eggs etc...) in the developing world. The root of this "revolution" is the fast increasing human population, rising incomes, advancing urbanisation and change in the diet habits in the developing part of the world. This phenomenon has profound implications for human health, the environment and the economic outcome of the concerned region.

Livestock farming in resource-poor (RP) communities presents multiple challenges. Sheep and goat farming in developing countries suffers from tremendous economic losses from a variety

of diseases, including parasitic helminth infections [39]. Optimal helminth management is imperative for a farmer to achieve, but is complex and especially difficult without access to expert help.

1.3.1 Helminth infection in small ruminants

Helminth infection is a leading cause of death in livestock farming, especially in poor tropical regions of the world, where the combination of recurrent rainfall and poor husbandry management causes significant economic loss in those regions [39]. The life of most Helminth species starts with the female laying up to 10,000 eggs depending on the specie [125] inside the host animal, which will be passed out in the faeces. The development of the egg into larvae is dependent on exogenous factors such as the moisture and the living bacteria in the faeces, dry conditions reduce the chances of development of the juvenile larvae. While grazing, ruminants ingest larvae that can complete their development inside the animal where they can feed on its blood and produce new eggs.

The gastrointestinal nematode *Haemonchus contortus (H. contortus)* has a particularly heavy impact on small ruminants in tropical and subtropical regions, as these regions provide a favourable environment for its development. Each female *H. contortus* produces up to 10,000 eggs per day [125], and these develop into infective larvae in a few days under warm and moist conditions. Re-infection can result in high parasite burdens and acute disease outbreaks often leading to death, especially among young animals [51]. The disease is primarily the result of blood-feeding by adult worms in the abomasum, leading to anaemia, protein loss, and associated consequences for health, growth and fertility [13]. The economic loss due to helminth infection in sheep and goat production is substantial, for example, an estimated $40 million per annum in the Kano area of northern Nigeria and $26 million per annum in Kenya [39].

Rainfall plays a pivotal role in the life cycle and development of Haemonchus contortus parasites. Because the life cycle of Haemonchus contortus involves a free-living stage in the environment, where larvae develop within the faecal matter, rainfall is crucial as it creates a conducive environment for the survival and movement of the larvae, allowing them to migrate onto pasture where they can be ingested by grazing animals during foraging [104] [105] [11]. Precipitation events, contributes to the sporadic distribution of infective larvae on pasture, increasing the risk of exposure for susceptible hosts. Temperature and seasons also have significant influence over the life cycle of Haemonchus contortus parasites [78]. Warmer temperatures generally accelerate larval development, promoting a faster transition from one stage to another. In contrast, colder temperatures can slow down this process. During warmer seasons, the environment becomes more favorable for larval survival on pasture, increasing the risk of infection for grazing animals. In colder seasons, the opposite occurs, with decreased larval survival [126]. Consequently, understanding the interplay between rainfall, temperature patterns and parasite development is vital for implementing effective parasite control strategies in livestock management.

Although multiple worm control strategies for RP farmers exist [52], including chemical dewormers, vaccination, anthelmintic drugs, grazing management, specific diets and ethnoveterinary remedies, they all require high manual labour and other expenses [89, 144]. In addition, widespread use of anthelmintic drugs has led to the high prevalence of anthelmintic resistance (AR) in countries such as South Africa [141, 145, 147]. This is due to farmers relying on anthelmintics as the sole method of control against helminth infection, and poor practices such as treating the entire herd when only a few individuals are affected. Although helminth infections are curable, they are common and have high ongoing costs relative to other diseases due to the complexity and operational difficulties of effective and sustainable management [114]. Indeed, farmers in Sub-Saharan Africa rank helminths as the most important disease in small ruminants, in spite of more visible (*e.g.* ectoparasitic) and ostensibly more damaging (*e.g.* foot and mouth disease virus) pathogens [113].

1.4 Companion animal health

Companion animal health refers to the overall health and well-being of pets, including dogs, cats, and other domestic animals that are kept as companions. This includes physical health, such as nutrition, exercise, and disease prevention, as well as mental and emotional health, such as socialisation, behaviour, and stress management. Physical health is an important aspect of companion animal health. Proper nutrition and exercise are essential for maintaining a healthy weight and preventing obesity, which can lead to a range of health problems such as diabetes, heart disease, and joint problems [14] [69]. Regular veterinary check-ups and preventative care, such as vaccinations and parasite prevention, can also help maintain a pet's physical health and prevent diseases. Mental and emotional health are also important for companion animals. Socialisation with other pets and people, as well as training and positive reinforcement, can help prevent behavioural issues and promote mental well-being. Stress management techniques, such as environmental enrichment and relaxation training, can also help pets cope with stress and anxiety [28]. Healthy companion animal health can improve the quality of life for both the pet and its owner [44]. By maintaining physical and mental health through proper nutrition, exercise, and stress management, as well as utilising emerging technologies and veterinary care, pets can live happy, healthy lives with their owners.

The use of technology, such as machine learning and accelerometers, is increasingly being used to monitor and improve companion animal health [3] [101]. These technologies can provide valuable insights into a pet's behaviour and activity levels, allowing pet owners and veterinarians to detect potential health problems early and develop effective strategies for improving the pet's health and well-being. In part thanks to increasing owner awareness, the companion animal market is also a growing market which presents many business opportunities [65].

1.5 Accelerometery

In 1915 Albert Einstein described the general theory of relativity which explains how Gravity affects the physical world [38]. On Earth gravity is defined as "The total force acting on a body at rest on the earth's surface is the result of gravitational force and the centrifugal force of the earth's rotation and is called gravity". The nominal "average" value at Earth's surface, known as standard gravity is, by definition, $9.80665 m/s^2$, which is also the change of velocity or acceleration of a free-falling object. Isaac Newton's second law of motion defines acceleration as the amount of force needed to move each unit of mass with the equation $a = F/m$ where a is the acceleration, F the force and m the mass. Accelerometers are devices which measure the acceleration they are subjected to.

Accelerometers are devices which measure the physical acceleration experienced by an object in its own coordinate system. Although different types of accelerometers exist, they all follow the same principle. In a closed casing, a reference mass attached to the spring will move if subject to external forces, the effect of the moving mass on the system can then be measured. A common type of mechanical accelerometer includes Piezoelectric accelerometers [83] in which the enclosed mass comes in contact with a piezoelectric material i.e. a material which accumulates electric charge when exposed to mechanical stress (Fig. 1.2). Capacitive accelerometers use the change in the distance of two metal plates as the mass moves which gives a measurement of their capacitance. Seismometers are a kind of accelerometer which use a pen attached to the moving mass, when an earthquake strikes, as the mass moves and bounces back in the casing the trace left by the pen registers the earthquake forces. Mechanical accelerometers are rarely used in common consumer electronics due to their physical size requirements. Instead, Micro electro-mechanical system (MEMS) accelerometers are more commonly used for a variety of applications ranging from fitness monitoring to video game controllers [5].

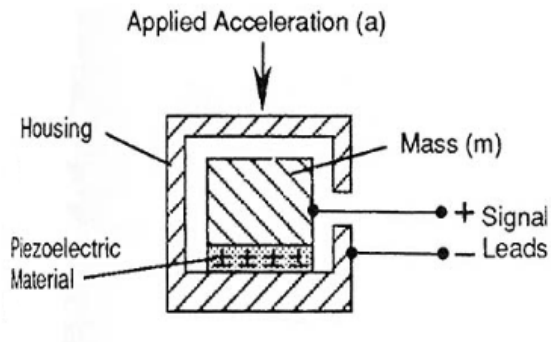


Figure 1.2: **Schematic of a basic piezoelectric accelerometer.** Image taken from source [68].

MEMS accelerometers output signals that represent the acceleration along one or more axes. The output is typically in the form of a voltage or current, which is proportional to the acceleration along the sensing axis. This analogue signal is then converted into a digital signal using an

analogue-to-digital converter (ADC). The digital signal is usually a binary code that represents the amplitude of the analogue signal at a particular point in time. A 3-axis accelerometer works by measuring the acceleration along three mutually perpendicular axes, typically labelled X, Y, and Z. The sensing element of a 3-axis accelerometer consists of a small proof mass suspended by one or more springs. The proof mass is mounted on a set of movable plates that are capacitively coupled to stationary plates. When the accelerometer experiences acceleration along one or more of the sensing axes, the proof mass moves relative to the stationary plates, causing a change in capacitance. This change in capacitance is detected and converted into a voltage signal, which is amplified and then digitised by an ADC. The digital output of a 3-axis accelerometer consists of three streams of binary data, each representing the acceleration along one of the three axes. The data is typically sampled at a fixed rate, such as 100 Hz or 1 kHz, and each sample includes the acceleration values for all three axes. This data can be further processed to extract features such as velocity, displacement, or orientation, depending on the application. In many cases, the 3-axis accelerometer data is combined with other sensor data, such as a magnetometer or gyroscope data, to provide a more complete picture of the motion or orientation of the object being measured. For example Kathpalia et al used a 3-axis accelerometer to enhance the location accuracy of a car global positioning system by predicting driving parameters [73].

1.6 Digital signal processing

The science of Digital signal processing (DSP) allows the processing of such signals in order to maximise the intrinsic information content. The time domain refers to data measured with respect to time, it shows how a signal changes over time while the frequency domain shows the frequency components of the signal across multiple frequencies. The time-frequency domain allow us to describe the evolution of the frequencies within a signal or time series over time. In practice switching to the frequency domain gives us a different perspective of the data which might reveal characteristics which were not obvious in the time domain. Techniques such as the Fast Fourier transform (FFT) [103] and the Continuous Wavelet Transform (CWT) [53], [26] are staple techniques for digital signal analysis in the time and time-frequency domains respectively, and can be used for accelerometry data analysis and interpretation.

A FFT is a mathematical algorithm used to transform a time-domain signal into its frequency-domain representation. It does this by decomposing a complex signal into a sum of sinusoidal waves of varying frequencies and amplitudes. This transformation is useful because it allows the identification of the frequency components of a signal, which can reveal important information about the signal's characteristics, such as its dominant frequencies, periodicity, or spectral content. The FFT is widely used in digital signal processing, image processing, audio processing, and other applications where signal analysis is required. It is called "fast" because it can perform the transformation much more quickly than the traditional Fourier transform [18] by exploit-

ing symmetries and avoiding redundant calculations. Analysis of the frequency component of accelerometry signal via spectrogram has been extensively used to assess the state of mechanical systems and can successfully be used to monitor the remaining usable lifespan of the mechanical systems such as bearings and gas turbines, for example.

The FFT is more appropriate for stationary signals. Because activity data is not stationary, i.e. it changes over time, for example, the activity at day time will differ from the activity at night time as animals sleep, it is often appropriate to focus on the CWT which is a time-frequency transform that allows the analysis of a signal into different time and frequency resolution. The Continuous Wavelet Transform (CWT) pioneered by Ingrid Daubechies [26] decomposes a signal using Wavelets of different scales and locations. The core principle of this transform is to slide a wavelet across the input signal for a set of different wavelet properties, at every time step or location the Wavelet is multiplied by the signal for every Wavelet scale selected, and the "sliding multiplying" process is called the convolution of the signal with the Wavelet, convolution of two functions, is combining them in such a way that tracks their interaction throughout time. The wavelet transform can extract local frequency and time information simultaneously. By choosing a Wavelet shape that matches the feature of interest in the input signal the CWT is a powerful tool to detect localised events, see Fig 1.3 for examples.

For a signal or time series f which changes over time t, ϕ^* the complex conjugate of the wavelet ϕ, s is the scale parameter which is defined as the inverse of the frequency of ϕ

$$F(\tau,s) = \frac{1}{\sqrt{|s|}} \int_{-\infty}^{+\infty} f(t)\phi^*\left(\frac{t-\tau}{s}\right) dt \tag{1.1}$$

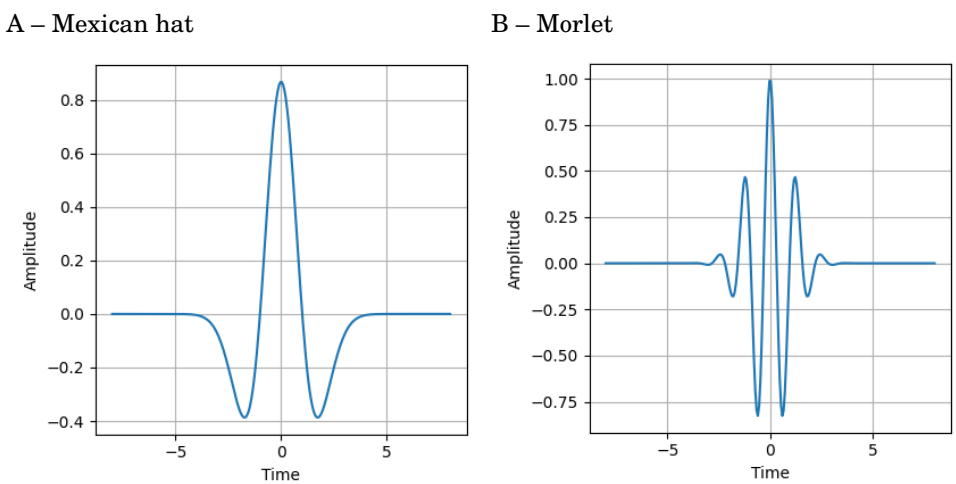


Figure 1.3: **Wavelets**. Illustration of two common continuous wavelets, the Mexican hat wavelet (A) and the Morlet wavelet (B).

A Wavelet $\Psi(t)$ can be scaled and/or shifted. Scaling expands or shrinks $\Psi(t)$ in time and also changes its frequency, with s a positive scaling factor $\Psi(\frac{t}{s})$ is a scaled wavelet, s is inversely

proportional to frequency such that the wavelet frequency is defined by $F_{eq} = \frac{C_f}{s\delta t}$ where C_f is the central frequency of the wavelet, which is the frequency of the approximated sine wave that fits the wavelet best([50]), s is the scale and δt the sampling interval. For example, scaling a wavelet by a factor of 2 results in dividing its frequency by 2. A stretched wavelet can capture the slowly varying changes in a signal (i.e the low-frequency components) while a compressed wavelet can capture the abrupt changes (i.e the high-frequencies). By constructing different scales we can look for multiple different frequencies in a signal. Shifting a wavelet means delaying or advancing it in time along the original signal time axis, it can be described. The CWT scales and shifts a wavelet across a signal to obtain its time-frequency representation and outputs coefficients that are function of frequency (or scale as it is inversely proportional to scale) and time. For k a shifting factor, s a scaling factor, $x(t)$ a time-domain signal and Ψ a Wavelet function, the CWT of x can be calculated with the formula:

$$CWT(k,s) = \frac{1}{|\sqrt{s}|}\int_{+\infty}^{-\infty} x(t)\Psi(\frac{t-k}{s})\,dt \tag{1.2}$$

To give some examples, Ayaz et al [8] used the CWT of the signal outputted by accelerometers measuring the vibration of electric motors to assess the level of damage of the hardware. A wavelet is a wave-like oscillation localised in time, a wavelet has two main properties, scale i.e., how dilated or compressed the wavelet is and location. Silva et al [133] used accelerometer data analysis with CWT for early fault detection of gas turbines by detecting rubbing elements. A demonstration of the CWT on animal activity date is given in Fig 1.4.

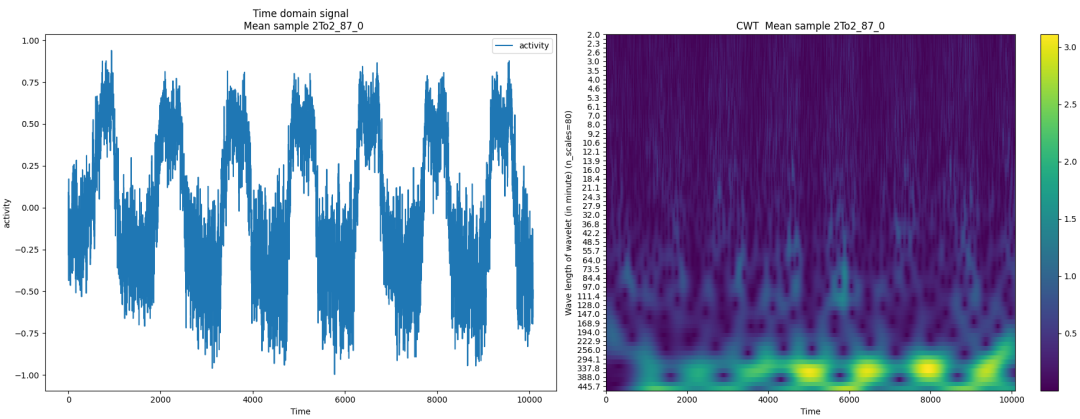


Figure 1.4: **CWT of activity data.** While the figure on the left shows a 7-day-long sheep activity signal after pre-processing and centering, the figure on the right shows its CWT, the low frequencies (bottom of y-axis) from the day/night cycle are clearly visible as well as the higher frequencies of activity.

1.7 Supervised machine learning

The term "Machine Learning" was popularised in 1959 by Arthur Samuel, it is the ensemble of computing techniques in which algorithms use data to automatically perform a task the most optimally without being directly programmed to do so.

Supervised ML is a type of ML in which an algorithm is trained to learn a mapping between input and output variables, based on a labelled dataset. In other words, the algorithm is given examples of inputs and their corresponding desired outputs, and it learns to generalise from these examples to predict the outputs for new, unseen inputs. To train a supervised ML model, a dataset is divided into two parts: a training set and a test set. The training set contains a set of input-output pairs, and the goal is to learn a mapping from the inputs to the outputs. The algorithm is presented with the input-output pairs one at a time, and it adjusts its internal parameters to minimise the difference between its predicted output and the true output. This process is known as optimisation, and it typically involves minimising a loss function that quantifies the difference between the predicted and true outputs. Once the model has been trained on the training set, it is evaluated on the test set to measure its performance on new, unseen examples. The test set contains input-output pairs that were not used in training, and the model's predictions for these examples are compared to the true outputs. The model's performance is usually assessed using metrics like AUC, accuracy, or precision. Supervised ML can be used for a wide range of tasks, including classification, regression, and sequence prediction. Examples of classification tasks include image recognition, spam detection, and sentiment analysis, while examples of regression tasks include price prediction, weather forecasting, and stock market analysis. Sequence prediction tasks include natural language processing, speech recognition, and time series forecasting. Some common algorithms used in supervised ML include decision trees, random forests, support vector machines (SVM), k-nearest neighbours (k-NN), and neural networks. These algorithms vary in their complexity, scalability, and interpretability, and the choice of algorithm depends on the nature of the problem, the size of the dataset, and the desired performance.

1.7.1 Support vector machine

As an example, the Support Vector Machine [15] (SVM) is a commonly use supervised ML technique to solve classification problems. For a set of points we wish to classify, each point is represented by a feature vector x in a finite linear D dimension space such that $x \in \mathbb{R}^D$. Because in many real-world cases data points often cannot be separated in a simple linear space, we map the previous space to a more complex non-linear higher M dimensional space ϕ such that $\phi : \mathbb{R}^D \to \mathbb{R}^M$ and $\phi(x) \in \mathbb{R}^M$ SVM aims to find a decision boundary (DB) which separates best the points into their respective classes, this separator is defined as a $M-1$ dimension hyper-plane H

such that,

$$H = w^T \phi(x) + b = 0 \tag{1.3}$$

where w and b are the hyper-plane hyper-parameters, w is called the weights and b is the bias. The distance of the hyper-plane H from a point vector x_0 is defined as d_H such that:

$$d_H(\phi(x_0)) = \frac{|w^T \phi(x) + b|}{||w||^2} \tag{1.4}$$

the SVM algorithm goal is to find the hyper-plane that has the maximum margin from the closest points we wish to classify, in other words the goal is to maximise the minimum distance (figure 1.5), for n data points this optimisation problem can be written as:

$$w^* = \arg\max_w [\min_n d_H(\phi(x_n))] \tag{1.5}$$

For a binary (there are two classes of data points) classification problem, substituting a class 1 point into the DB equation 1.3 will output a > 0 value while a class 2 will give a < 0 value,

$$w^T \phi(x) + b > 0, x \in class1 w^T \phi(x) + b < 0, x \in class2 \tag{1.6}$$

The predicted class of a point x is correct or incorrect depending on the value of the product of the DB equation and the actual data point class y, for n data points,

$$y_n[w^T \phi(x) + b] = \begin{cases} \geq 0, correct \\ < 0, incorrect \end{cases} \tag{1.7}$$

In the case of a perfectly separable dataset, H separate all points correctly,

$$w^* = \arg\max_w \left[\min_n \frac{y_n[w^T \phi(x) + b]}{||w||^2} \right] = \arg\max_w \frac{1}{||w||^2} \left[\min_n [w^T \phi(x) + b] \right] \tag{1.8}$$

We can normalise (re-scale) the inner term which represent the distance of the closet point to the DB,

$$\left[\min_n [w^T \phi(x) + b] \right] = 1 \tag{1.9}$$

this can be done by multiplying the weights w and bias d by a constant c which keeps the direction of the vectors in H which won't change the sign of the output of the equation 1.3.

$$w^* = \arg\max_w \frac{1}{||w||^2} \tag{1.10}$$

We can convert the last step 1.10 into a minimisation problem and introduce the half for mathematical convenience, this gives the equation known as the Primal formulation of SVM for perfectly separable data.

$$\begin{aligned} w^* &= \arg\min_w \tfrac{1}{2} ||w||^2 \\ y_n[w^T \phi(x) + b] &\geq 1 \, \forall n \end{aligned} \tag{1.11}$$

In real-world scenarios, the assumption that every point in the dataset is separable is almost never met. Instead of aiming for perfect classification of each point, we allow the algorithm to make some incorrect classifications. This is done by introducing for each data point a slack variable ξ, for a correctly classify data point $\xi = 0$ while for a incorrectly classified data point $\xi > 1$.

$$\begin{aligned} &\underset{w,b,\xi_n}{\arg\min} \tfrac{1}{2}||w||^2 + C\sum_n \xi_n \\ &y_n[w^T\phi(x) + b] \geq 1 - \xi_n \; \forall n \\ &\xi_n \geq 0 \forall n \end{aligned} \tag{1.12}$$

To control the amount of slack permitted for miss classification, a regularisation parameter is employed (C). If C equals 0, the classifier faces no penalty for slack, allowing for considerable miss classification. In this instance, the decision boundary (DB) will be linear, and there is a risk of underfitting. On the contrary, an infinite C ($C = \infty$) signifies that even a slight slack is heavily penalised, resulting in a very intricate DB and potential over-fitting. 1.12 is a Convex Quadratic optimisation problem because the objective function is quadratic in w and the constraints are linear in w and ξ. To solve this problem we need to find ϕ, which is difficult to compute. The Dual Formulation of SVM eliminates the reliance on ϕ by introducing the concept of a Kernel. This formulation rephrases the same problem using a different set of variables, thanks to the method of Lagrange Multipliers. In a general problem where x is the variable, and we aim to minimise a function f under a set of constraints determined by a set of functions g_i:

$$\begin{aligned} &\min_x f(x) \\ &g_i(x) \leq 0 \end{aligned} \tag{1.13}$$

The problem can be rewritten as Lagrangian, where instead of having a set of variable x introduces a new set of variables also called Lagrange Multipliers λ_i:

$$\begin{aligned} &L(x, \lambda_i) = f(x) + \sum_{i=1}^{n} \lambda_i g_i(x,) \\ &\lambda_i \geq 0 \end{aligned} \tag{1.14}$$

When we apply Lagrange Method to the SVM primal equation1.12, the Lagrangian is:

$$L(w, b, \{\xi\}, \{\lambda_n\}, \{\alpha_n\}) = \left[\frac{1}{2}||w||^2 + C\sum_n \xi_n\right] + \sum_n [\alpha_n\{1 - \xi_n - y_n[w^T\phi(x_n) + b]\}] + \sum_n \lambda_n(-\xi_n) \tag{1.15}$$

by differentiating with respect with the primal variables w, b and ξ:

$$\begin{aligned} &\tfrac{\delta L}{\delta w} = w - \sum_n \alpha_n y_n \phi(x_n) = 0 \Rightarrow w = \sum_n \alpha_n y_n \phi(x_n) \\ &\tfrac{\delta L}{\delta b} = \sum_n \alpha_n y_n = 0 \Rightarrow \sum_n \alpha_n y_n = 0 \\ &\tfrac{\delta L}{\delta \xi_n} = C - \alpha_n - \lambda_n = 0 \Rightarrow C - \alpha_n - \lambda_n = 0 \end{aligned} \tag{1.16}$$

By substituting 1.16 in 1.15 we can eliminate the primal variables and express the Dual form of SVM:

$$
\begin{aligned}
&\max_{\alpha_i} g(\{\lambda_n\},\{\alpha\}) = \sum_n \alpha_n + \tfrac{1}{2}\sum_n \alpha_m \alpha_n y_m y_n \phi^T(x_m)\phi(x_n) \\
&\alpha_n, \lambda_n \geq 0 \; \forall n \\
&\sum_n \alpha_n y_n = 0 \\
&C - \alpha_n - \lambda_n = 0
\end{aligned} \tag{1.17}
$$

In this form the ϕ dependency can be replaced with a kernel, a kernel is a function of the base feature x which is symmetric and positive semi-definite, for a kernel k, k has two properties:

- $k(x_m, x_n) = k(x_n, x_m)$
- $\sum_n \sum_n v_m v_n k(x_m, x_n) \geq 0$

The inner product $\phi^T(x_m)\phi(x_n)$ can be represented with a kernel function such as $\phi^T(x_m)\phi(x_n) = k(x_m, x_n)$

1.17 hence becomes ϕ independent and much easier to compute for a convex quadratic equation solver which will output a set of weights and bias:

$$
\begin{aligned}
&\max_{\alpha_i} g(\{\lambda_n\},\{\alpha\}) = \sum_n \alpha_n + \tfrac{1}{2}\sum_n \alpha_m \alpha_n y_m y_n k(x_m, x_n) \\
&\alpha_n, \lambda_n \geq 0 \; \forall n \\
&\sum_n \alpha_n y_n = 0 \\
&C - \alpha_n - \lambda_n = 0
\end{aligned} \tag{1.18}
$$

To make predictions we can use,

$$
w^T \phi(x) = \sum_n \alpha_n y_n k(x_n, x) \tag{1.19}
$$

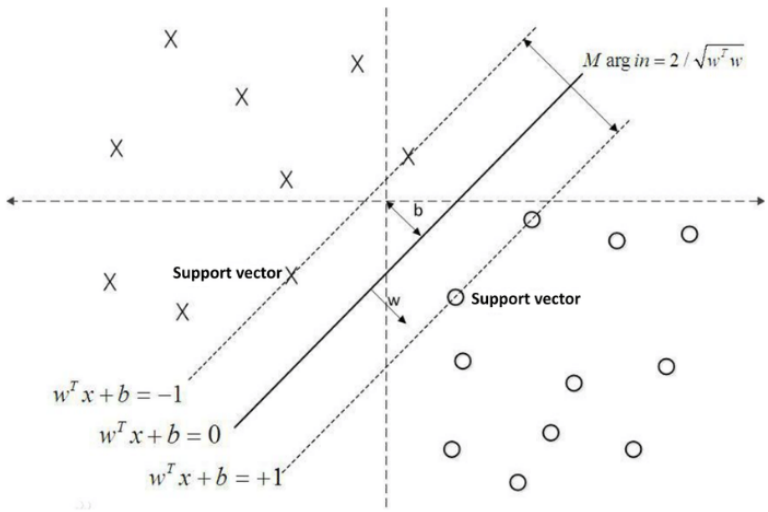


Figure 1.5: **Schematic of linear SVM.** The DB of SVM with the largest margin. × and o denote two-class training examples. $wTx + b = 0$ is the optimal hyperplane to do the separation, where w is a weight vector and b is a bias, and an SVM training model with the largest margin $2/\sqrt{wTw}$ is built. The support vectors are the samples on the dotted lines. The optimisation classification hyperplane is determined by the solid line. Image is taken from source [85].

When the data is not linearly separable, SVMs use a kernel function to transform the input data into a higher dimensional space where the data becomes linearly separable. One common kernel function used in SVMs is the Radial Basis Function (RBF) kernel [149]. The RBF kernel is a popular choice because it is capable of modelling complex, non-linear decision boundaries between classes. The RBF kernel maps the input data to an infinite-dimensional feature space, making it highly flexible and capable of capturing complex patterns in the data. The RBF SVM classification algorithm has several hyperparameters that can be tuned to improve its performance, including the regularisation parameter, the kernel coefficient, and the kernel width. The regularisation parameter controls the trade-off between maximising the margin and minimising the training error, while the kernel coefficient and kernel width control the shape and scale of the DB. The optimal hyperparameters are typically found through a process of cross-validation on a separate validation dataset.

$$K(x, y) = \exp(-\gamma ||x - y||^2) \tag{1.20}$$

where x and y are two data points, $||x - y||^2$ is the squared Euclidean distance between them, and γ is the kernel coefficient that controls the width of the kernel. The value of γ determines the width of the kernel function. A small gamma results in a wider kernel, which means that the DB is smoother, while a large gamma results in a narrower kernel, which means that the DB is more complex and can better fit the training data.

1.7.2 Deep learning

Deep learning refers to the use of neural networks with multiple layers to perform complex tasks such as image and speech recognition, natural language processing, time series forecasting and robotics. Deep learning models are trained using large amounts of labelled data, using a process known as backpropagation, which involves iteratively adjusting the weights of the network to minimise a loss function. One of the key advantages of deep learning is its ability to automatically learn relevant features from raw data, without the need for manual feature engineering. This makes it a powerful tool for tasks such as image recognition, where the relevant features may be complex and difficult to define explicitly.

Convolutional Neural Networks (CNNs) are a type of deep neural network commonly used for image classification, object detection, and other computer vision tasks. CNNs are composed of multiple layers of neurons, each of which performs a specific transformation on the input data. The first layer of a CNN typically consists of several convolutional filters, which extract local features from the input image. These filters slide across the input image, performing a dot product with each pixel and its surrounding neighbours. The output of this operation is a set of feature maps that capture different aspects of the image, such as edges, corners, and textures. The subsequent layers of a CNN typically consist of pooling layers, which reduce the spatial dimensionality of the feature maps, and activation layers, which introduce non-linearity into the model. The output of these layers is passed through several fully connected layers, which perform high-level abstraction and produce the final classification output. CNN can also be used for time series or digital signal classification, for example, Baek et Youngji [158] predicted the useful lifetime of bearings by measuring the vibration experienced by the bearings with accelerometers and used a CNN trained with the CWT of the measured data for prediction.

In addition to CNNs, other types of deep neural networks include recurrent neural networks (RNNs), which are used for sequence modelling tasks such as speech recognition, language translation and other time series data [86], and generative adversarial networks (GANs), which are used for generative tasks such as image synthesis and data augmentation [138]. Overall, deep learning has revolutionised the field of artificial intelligence in recent years, enabling breakthroughs in a wide range of applications such as self-driving cars, medical diagnosis, and recommender systems. Its ability to automatically learn complex patterns from data has made it a powerful tool for solving previously intractable problems, and its potential for further advances in the future is enormous. However, due to the large number of parameters in these models, deep learning is reliant on appropriately large amounts of training data.

1.8 Human activity monitoring

Accelerometers have been extensively used in human health monitoring with much success. Most of this research focuses on classifying human gait or distinct activities such as sitting or walking,

which can then used as surrogate measures of health. Less research has been performed to directly correlate a person's health with accelerometry data.

Chung et al [23] used a combination of electrocardiogram (ECG) and accelerometry monitory with on-body wireless sensors to detect abnormalities in the activity of elderly persons at home in real-time. The study uses an accelerometer to improve the analysis of the ECG data by detecting when the user is resting. In this study, the health assessment is derived from the analysis of the ECG data however the decision making is only done when the patient is detected to be resting. Hong et al [64] developed a mobile health monitoring system based on activity recognition using an accelerometer. In this study, the research team was able to estimate the calorific expenditure of 6 people of different weights by modelling the relation between energy expenditure and activity magnitude by simultaneously measuring the accelerometry data and the carbon dioxide output of a user. Crouter et al [25] performed a study in which 48 participants' energy expenditure was measured with a gas analyser during various activities where the participant was wearing an accelerometer. Although these studies were performed on a small sample size, it was found that accelerometry data alone can be used to estimate the calorie output of an individual which is a key metric to maintain a healthy weight but is often overestimated by other systems. Bourke et al [17] used a threshold-based analysis of accelerometer data to accurately detect human falls while Doughty et al [36] reduced the false positive rate of detection of falls by using accelerometry data and orientation data from an additional sensor. Zakaria et al [136] used consumer-grade smartphone accelerometers and ML to detect human activities (standing, sitting, laying, going up/down stairs, walking).

1.9 Livestock activity monitoring

In the case of livestock, developing a monitoring system based on accelerometers is a practical alternative to video-based tracking particularly where the animals can roam over wide areas. Welfare assessment of livestock is traditionally performed by direct visual observation which is prone to human error and time-consuming, cheap alternatives such as accelerometer-based sensors are an emerging option for precision livestock farming. Most of the research uses accelerometry data to detect behavioural traits that are linked to animal health, while little research has been done to directly link accelerometry data to animal health.

Accelerometer studies are most often conducted in ruminants, mainly dairy cows, goats and sheep. Krieger et al [79] used a tail-mounted accelerometer to predict calving in dairy cows with a real-time monitoring system, it has been shown that such a system can help reduce stillborn calves occurrences and other calving complications by detecting the onset of parturition minutes before the birth. The team used digital signal processing techniques to detect tail raising events and video monitoring to test their algorithm. Werner et al [156] tested the accuracy of the commercially available dairy cow accelerometer-based sensor MooMonitor+ by visually

comparing grazing behaviour with the sensor output. In their experiment, the sensor analysis highly correlated with grazing time. Grazing is a key metric for pasture management and general animal welfare, such a device can optimise animal management. Pastell et all [110] used multiple accelerometers attached to each limb of dairy cows to detect lameness with high accuracy. Wavelet data analysis of the accelerometry trace allowed the team to detect asymmetry of gait during walking. Shahriar et al [129] used unsupervised ML, namely K-mean clustering to accurately detect heat events in dairy cows by analysis of the intensity of the FFT [60] of the activity recorded. Moreau et al [97] showed that it is possible to classify goat's grazing behaviour, specifically eating, resting and walking with the commercially available HOBO bio-logger which uses a tri-axial accelerometer internally and relatively simple threshold-based analysis of the activity traces. Alvarenga et al [4] used a Decision Tree [72] and an accelerometer placed under the jaw to successfully discriminate between chewing and biting eating behaviours which is valuable for grassland management. Marais et al [90] used the ML algorithms, linear discriminant analysis and quadratic discriminant Analysis to accurately classify, lying, standing, walking, running and grazing sheep with accelerometers attached to a collar. Although less extensive the use of an accelerometer for pigs is possible in some scenarios. In a study by Halvorsen et al [56] epicardial accelerometers (accelerometers which monitor the motion of the outer surface of the heart) were used for real-time accurate detection of myocardial ischaemia which occurs when blood flow to the heart is obstructed in pigs. Ramonet et al [121] used commercially available accelerometers (Hobo Pendant G) attached to sow legs to detect standing and lying posture.

1.10 Companion animal activity monitoring

ML is increasingly being used for companion animal health to provide a better understanding of pet behaviour and activity levels. Accelerometers can be attached to a collar or harness and record the pet's movements. For example, Weiss et al [154] developed a dog collar for activity monitoring and recognition. Vision-based sensors have also been used, such as Pons et al [115] who developed a cat tracking system with a depth camera and was able to successfully classify cat's posture. Such technology is useful for pet owners and veterinarians to monitor the health and behaviour of companion animals, which can help detect potential health problems and improve the overall quality of life for pets.

For activity monitoring, the data collected can be used to track how much exercise a pet is getting and whether it is meeting recommended levels for its age and breed. This information can help identify potential health problems such as obesity, which can lead to a range of health issues for pets. Morrison et al [99] showed that obese dogs spent significantly less time in vigorous-intensity physical activity than ideal-weight dogs. Monitoring of pet behaviour is also a possible application. For example, accelerometers can be used to detect when a dog is barking excessively, which may be a sign of anxiety or other behavioural issues [16]. Uijil et al [30] were able to

accurately detect specific behaviours (walk, trot, canter/gallop, sleep, static/inactive, eat, drink, and headshake) in dogs by using collar-mounted accelerometers and video monitoring. Hussain et al [67] used wearable sensors (accelerometer and gyroscope) to infer dog well-being based on accurate detection of pet activities such as walking, sitting, down, staying, eating, sideway, jumping, running, shaking, and nose work. Other applications for using smart sensors on pets include location tracking, visual monitoring, feeding management and general health monitoring based on gait detection. The utility of different accelerometers for the detection of osteoarthritis in dogs has been reported [94]. In cats, it has been demonstrated that travelling distance and mobility are correlated with accelerometry data [81]).

The use of ML and wearable sensors for companion animal health has the potential to provide valuable insights into pet behaviour and activity levels. By monitoring certain aspects of a pet's life, veterinarians and pet owners can detect potential health problems early and take steps to improve the health and well-being of their pets. However, further research is needed to fully explore the potential of this technology and develop effective tools and strategies for using it in the field of veterinary medicine.

1.11 Discussion

Accelerometers have become an integrant part of the modern human lifestyle due to their simple and affordable nature, they are commonly used in everyday items such as smartphones and entertainment systems. In the industrial world, they are powerful tools to assess mechanical component health. In recent years, thanks to the advancement in miniaturisation and the rise in ML algorithm popularity they have successfully been used and researched as tools to monitor human health in various ways and are more and more used in animal health assessment for livestock management or companion animals. In the field of human activity monitoring, past studies have primarily utilised accelerometers to categorise activities such as gait or differentiate between sitting and walking, providing indirect indicators of health. However, a noticeable gap exists in directly correlating an individual's activity with accelerometry data. Decision-making processes in health assessments are often based on the occurrence of certain gait. In the domain of livestock, accelerometry data is frequently employed to identify behavioural traits linked to animal health. However, there is a scarcity of research establishing a direct link between accelerometry data and animal health. Studies predominantly concentrate on large ruminants, potentially limiting the generalisability of findings to other livestock species. Furthermore, the reliance on threshold-based analysis and limited sample sizes raises concerns about the robustness, scalability and applicability of the results. While companion animal activity monitoring shows promise, further exploration is needed to fully realise the potential of accelerometry data in the field of veterinary medicine.

1.12 Thesis structure

This thesis aim is to correlate animal health to accelerometry data. Chapter 1 provides a literature review of past and present use of accelerometry data for human, livestock and companion animal monitoring. The use of ML to analyse accelerometry data is also discussed. Chapter 2 describes in detail our small-ruminant study group and highlights with extensive visualisation its own unique challenges including the difficulty of data acquisition, and the drawbacks of the telemetry hardware used. It also explores solutions to combat the high amount of missing data in our dataset via the use of deep learning techniques which aim to understand typical activity patterns from the available data and predict missing points with minimal bias. Chapter 3 presents an interpretable ML pipeline for the classification of small-ruminant health status with accelerometry data. The effect of exogenous factors known to affect the animal's health status is also investigated. Chapter 4 shows that we can apply the pipeline devised in chapter 3 with minimal modifications to detect signs of degenerative joint disease in domestic cats. We conclude up by outlining the potential practical applications of this research and suggesting steps for further studies that could enable automated health monitoring for both livestock and pets.

1.13 Dissemination

Some parts of this thesis have been presented at academic conferences and published to preprint services with the aim of journal publication at future dates.

My work on small-ruminant health classification with ML and accelerometer is novel as it is based on a unique dataset which allows the matching of activity data to the level of parasitic infections of sheep and goats in South Africa. My findings are currently published on the bioRxiv preprint service and undergoing revision with up-to-date analysis. Submission to a peer-review journal is in preparation. In 2020 I also presented this novel work at the 74th Association for Veterinary Teaching and Research Work (AVTRW) Annual Conference.

The application of the pipeline I developed for the processing of livestock activity data to domestic cats has shown promising results in the classification of degenerative joint disease during peaks of high activity, and we have a manuscript in preparation for submission to The Veterinary Journal.

Ultimately, this research's true benefits lie in its translation to on-the-field systems that can accurately give insight into the health of farmers livestock and pet owners alike. I presented the proof of concept for an accelerometer-based wearable prototype for livestock, dogs, and cat and researched the business opportunities at the annual University of Bristol Careers Basecamp "Development Stage pitches" competition and was successful in my application and received funding to develop my own open source wearable sensor. I also intend to apply to the upcoming start-up competition to hopefully further close the gap between academic research and real-world applications.

CHAPTER 2

UNDERSTANDING AND IMPUTING TELEMETRIC LIVESTOCK ACTIVITY DATA

The research questions and aims of this chapter revolve around understanding and imputing telemetric livestock activity data gleaned from modest goat and sheep farms in South Africa. The principal dataset, curated by Dr Jan Aucamp van Wyk, initially underpinned an inquiry into the economic repercussions of nematode infection in small ruminants. The chapter aims to dissect the essence of this data, employing intricate visualisations and crafting imputation methodologies rooted in deep learning. Acknowledging the constraints posed by conventional machine learning techniques in dealing with limited datasets, the imputation process emerges as pivotal for optimal analysis. The spotlight is on two contemporary imputation techniques, Generative Adversarial Nets Imputation (GAIN) and Multi-Directional Recurrent Neural Networks (M-RNN), juxtaposed with the more straightforward approach of linear interpolation. The study strives to offer insights into the nuanced predicaments tied to livestock health management in South African rural communities, contributing to a broader comprehension of telemetric livestock activity data.

The data unpinning this chapter and the next was collected by Dr Jan Aucamp van Wyk in multiple South African small-scale goat and sheep farms. This data initially supported an investigation conducted by Dr Babayani, Nlingisisi Dombole, Prof Eric Morgan and Dr Jan Aucamp van Wyk [9], their research highlighting the economic impact of nematode infection in small ruminants, and the challenges associated with the health management of such livestock in South African rural communities. While the work of Babayani focused on the practical aspect of the data collection, such as testing the limitations of the hardware of the different versions of the accelerometer-based transponders used, limited statistical analysis of the data was performed and concluded that the activity level of the animals is a non-discriminatory risk indicator of clinical infection with *H. contortus*.

In this chapter we use detailed visualisations to understand the nature of this data, and develop deep learning based imputation methodology for the downstream supervised ML framework presented in chapter 3. As traditional ML techniques tend to generalise and perform better with larger datasets, it becomes necessary to impute the data for optimal analysis. The cat accelerometry study (chapter 4) did not suffer such issues as non-telemetric sensors were used, for that reason the imputation is only performed on the goat and sheep data.

Imputation is the process of estimating the value of missing data points, which can simply be calculated by simple interpolation approaches or with much more advanced approaches that harness the expected correlations with non-missing values. In this chapter we will focus on two different modern imputation techniques, Generative Adversarial Nets Imputation (GAIN) and Multi-Directional Recurrent Neural Networks (M-RNN), which both utilise deep learning of the conditional distribution of the data to intelligently impute the data. We compare the techniques to each other and also the simpler approach of linear interpolation. (Fig. 2.1).

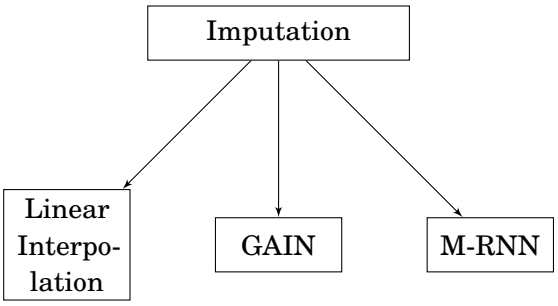


Figure 2.1: **Imputation approaches**.

2.1 Study description

Data were collected from a farm of 108 acres close to Delmas in Mpumalanga Province, South Africa, and at Cedara, a government research farm and agricultural college, the pastures of which comprise 25 acres adjacent to Hilton, KwaZulu-Natal. 31 female adult *Ile de France* sheep ewes at Delmas and 64 goats at Cedara were individually FAMACHA-evaluated [143] at weekly or

fortnightly intervals, respectively, and had associated longitudinal accelerometry data recorded within the study period. For both farms for the entirety of the study from January 2015 to April 2016 for the sheep farm and March 2012 to December 2013 for the goat farm, a number of the animals in the herds were equipped with transponders, on the goat farm in February 2013 the transponders were replaced with a newer version, this event is taken into account in order to not mix the data of different animals.

Discussion of FAMACHA evaluation is deferred to the next chapter (3) as the methods developed in this chapter deliberately ignore this information to avoid potential over-fitting as it is used as the response variable for prediction. On both farms, there were multiple improved pastures, which were irrigated, and utilisation occurred by alternation at intervals according to visual assessment of amounts of available water and forage. At both farms, young adult ewes/does were randomly selected for the trials, without attention to reproductive class, but remained with their flocks/herds of origin for the duration of each trial. The animals were kraaled at night and let out at a standard time in the mornings for herding to pasture, where they remained until collected and returned to the kraals in the late afternoon. Adjustments were made according to season and special management events such as vaccination, and hoof inspection, which were conducted first thing in the morning. As part of routine husbandry, each animal which scored ≥ 2 during a FAMACHA evaluation was immediately treated with Levamisole (Ripercol-L, Bayer Animal Health) at $7.5\,mg.kg^{-1}$ [9].

2.1.1 Telemetric monitoring system

Telemetric monitoring systems were supplied by Accitrack Ltd., Paarl, South Africa. Tagged animals on both farms were equipped with low-cost RFID transponders suspended by a sturdy ribbon around the neck (Fig. 2.2A). A single solar-powered base station (Fig. 2.2B). was installed on each farm, mounted at the top of a five-meter wooden pole. Each transponder contains an active RFID transceiver operating at 868Mhz as well as a battery and an A1 type accelerometer for activity level measurement. The accelerometers had a set acceleration threshold of 2g so that every time an acceleration $\geq$ 2g is sensed, a stored integer is incremented by +1. Version 2 transponders had an increased range (10km versus 1km) and a larger battery, at the expense of significantly increased weight. In addition, version 2 transponders also output minimum and maximum acceleration for the three orthogonal axes at each time interval, but this data was not utilised in this study. All tags were set to transmit the data every minute to the base station, at which time the accelerometer count was reset to zero. In order to extend battery life, the tag only transmits data to the base station once a minute, with data transmitted including the identifier of the transponder, the battery level, the signal strength, a timestamp, and the activity level. Data transmission is not performed if the signal-to-noise ratio drops below 10dB, which can occur when there are significant occlusions between the animal and base station. In these cases, the data for that time interval is lost and becomes missing data to impute. Through mobile connectivity with

General Package Radio Services (GPRS), the base station then regularly forwards the received data to the Accitrack cloud repository.

A – Sheep wearing a transponder B – Base station

Figure 2.2: **Telemetry hardware**. While (A) shows one of the collar mounted v2-accelerometer based transponders used in this research, (B) show a solar panel base station on the roof of a small building on a private South African farm.

2.2 Raw data pre-processing

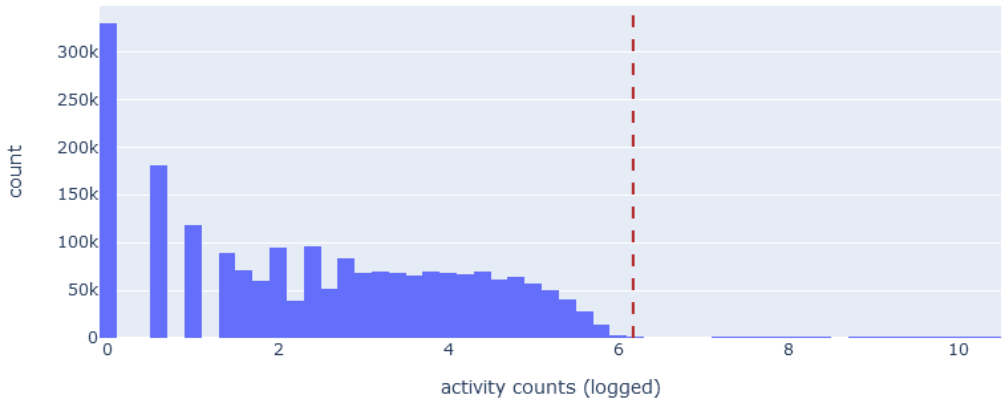
All raw data is stored on the Accitrack cloud repository for two weeks only due to storage limitations, hence it was manually downloaded regularly by a researcher at the University of Pretoria for archival [9]. The exported data took the form of CSV files containing the sensor outputs in the desired time frame. In cases where the data was not retrieved from the cloud infrastructure, datasets for that time period were lost. The table storing the raw data recorded at the original minute resolution contains 11, 594, 243 records for the sheep farm Delmas and 75, 449, 986 for the goat farm Cedara. The raw data in CSV (Fig. 2.4) format was parsed and then transferred to the Hierarchical Data Format 5 (HDF5) (Fig. 2.5). All of the valid activity records found in the raw spreadsheet of a farm are stored in one corresponding HDF5 file. A non-valid record is defined as a record which reports a negative activity count or an activity count whose value is superior to 480, this threshold was estimated by visualising the histogram of the raw activity counts on each farm (Fig. 2.3). We observed that a small number (relative to the total number of record) of the recorded activity count measured very high values with the maximum count recorded at 22k on the goat farm, it is unclear what caused such high values but it is unlikely linked to a normal or possible natural animal behaviour. A negative accelerometry value can be explained by sensor memory overflow which occurs when the sensor return value is too large to be stored in the available memory. Such an event could occur during animal manipulation, which also might be the cause of activity counts values above 480. Considering the

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1			FIELD 1	FIELD 2	FIELD 3	FIELD 4	FIELD 5	FIELD 6	FIELD 7	FIELD 8	FIELD 9	FIELD 10	FIELD 11			
2	Date	Time	Control station	Type 12 tag messages	Tag serial number	Signal strength	Battery voltage	First accelerometer counter	First sensor value	Second accelerometer counter	Second sensor values (X-Y-Z)	Correlation identifier	Correlation value			
3	31/03/2015 00:00	6:00:00 AM	70101200027	12	40101310107	-71@	BC	A1	0	A2	-4:0:0:14:0:30	I	0			
4	31/03/2015 00:00	6:00:01 AM	70101200027	12	40101310109	-78@	BB	A1	0	A2	-1:9:-1:8:26:35	I	31			
5	31/03/2015 00:00	6:00:03 AM	70101200027	12	40101310092	-95@	BD	A1	0	A2	-1:0:-34:-31:-1:0	I	26	C	1742	1.00E-12
6	31/03/2015 00:00	6:00:04 AM	70101200027	12	40101310115	-74@	BB	A1	0	A2	2:7:17:22:23:25	I	2			
7	31/03/2015 00:00	6:00:06 AM	70101200027	12	40101310106	-76@	BB	A1	0	A2	12:15:-23:-16:21:25	I	23			
8	31/03/2015 00:00	6:00:07 AM	70101200027	12	40101310117	-63@	BB	A1	0	A2	6:8:-14:-11:-30:-28	I	30	C	1742	1.00E-12

Figure 2.4: **Subset of a raw spreadsheet containing accelerometry data.**

sensor configuration (increment a new count from 0 every time acceleration experienced by the accelerometer is > 2g every minute).

A – Sheep raw data histogram



B – Goat raw data histogram

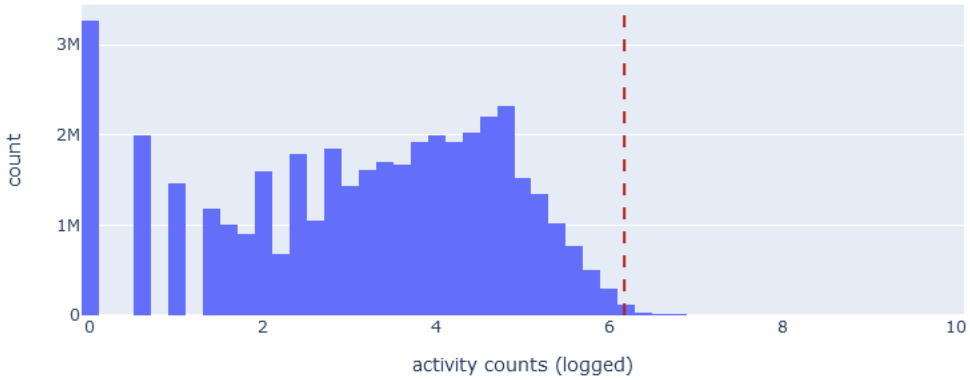


Figure 2.3: **Raw data histogram**. This figure shows the histograms of the log of the positive unfiltered activity counts. The vertical red dotted line illustrate the maximum activity count threshold, during normal activities we estimated that an activity count above 480 is unlikely to be linked to a natural or possible behaviour of a small ruminant. While (A) shows the histogram of the sheep farm (B) shows the goat farm. Note that negative and zero values were dismissed.

	0						
	index	timestamp	control_station	serial_number	signal_strength	battery_voltage	first_sensor_value
0	0	1427778...	70101200027	40101310107	-71	188	0
1	1	1427778...	70101200027	40101310109	-78	187	0
2	2	1427778...	70101200027	40101310092	-95	189	0
3	3	1427778...	70101200027	40101310115	-74	187	0
4	4	1427778...	70101200027	40101310106	-76	187	0
5	5	1427778...	70101200027	40101310117	-63	187	0
6	6	1427778...	70101200027	40101310026	-86	188	0
7	7	1427778...	70101200027	40101310239	-97	189	0
8	8	1427778...	70101200027	40101310347	-74	188	0
9	9	1427778...	70101200027	40101310052	-96	187	0
10	10	1427778...	70101200027	40101310389	-87	187	0

Figure 2.5: **Subset of a Hierarchical Data Format 5 (HDF5) file containing accelerometry data.**

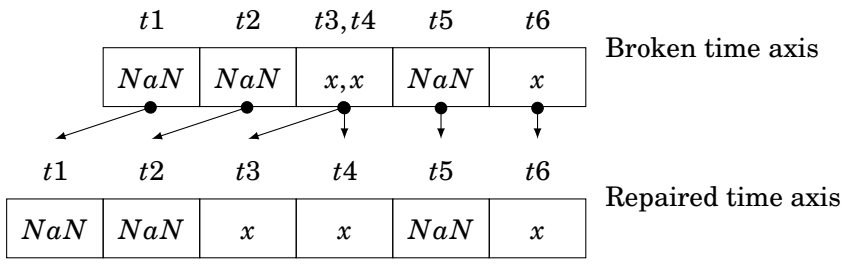


Figure 2.6: **Illustration of the back-fill repair process**. In this schematic x is an integer corresponding to a activity count value while 'NaN' (not a number) stands for a missing count.

2.2.1 Misalignment

Because of apparent misalignment in the accelerometry record timestamp of the Goat and Sheep transponders, the raw data needs to be "repaired" before imputation. The sampling resolution of the sensors used for this study was set to 1 record per minute. However, the transmission of the data does not always follow the same pattern (1 activity record sent every minute) because of communication errors the sensor transmits data only when possible. Furthermore, it is possible for the sensor to send multiple records for the same minute 'to catch up'. The reason why this occurs is unknown. Because of the irregularities of the sensor output, the first preprocessing step applied to the raw data is to re-align the sampling resolution to 1 record per minute, i.e allocate one record to a single minute bin. The repair process starts by fixing the time axis of every transponder on a common time axis that start at the time of the earliest record across all transponders and similarly ends at the time of the last record. The new time axis will holds a single activity count per bin (Fig. 2.6).

Figure 2.7: **Screen capture of Web visualisation tool** The web tool developed takes the form of a website, the top of the page contains a dashboard with access to multiple features, such as selection of the herd, and individual transponders and several and multiple post-processing options. The dashboard also displays details (Number of points in the signal, Minimum/Maximum activity values, start date, end date, time range of the data, presence/absence of FAMACHA report for the transponder) about the selected traces. Each trace can be zoomed in and will dynamically display a higher resolution of the data depending on the zoom level. This screen capture shows the data on the Delmas farm. The first row of figures shows on the Left the herd data as a heatmap of transponders (this figure shows the data aggregate at the day level, i.e each bin contains the sum of one day's activity count), on the right, two activity traces are also displayed at the day sampling resolution. The second row of figures shows on the left the CWT of the 2 traces selected and on the right the humidity, temperature, and signal strength. The last figure shows the histogram of the activity traces.

2.2.2 Transformations

In general, ML algorithms and statistical models assume that the data have Gaussian residuals. Because these algorithms are sensitive to the input data distribution, it is important to understand the distribution of the data before imputation or classification to optimise our preprocessing approach. Accelerometry data that has been captured as counts naturally follow a statistical Poisson distribution [74] which is a discrete distribution i.e. only takes a discrete set of values. It describes the number of events in a fixed time interval, for example, the number of sales a merchant makes every hour follows a Poisson distribution. This distribution is also characterised by the expected number of events per time interval parameter λ, where crucially the variance

rises directly with the expected number of counts. In addition, it is bounded by 0 and ∞. The Poisson distribution assumes that the rate at which each event occur is constant, in other words, the probability of an event occurring in a certain time interval should be the same for every time interval of equal length (assuming the underlying rate does not change due to covariates). And the occurrence of one event does not affect the occurrence of a subsequent event, i.e. the events are independent. The probability mass function P for X a discrete random variable with $\lambda > 0$ is defined as:

$$P(X = x) = \frac{e^{-\lambda}\lambda^x}{x!} \tag{2.1}$$

where,

- e is Euler's mathematical constant. $e \approx 2.71828\ldots$
- x is the number of occurrences. $x = 0, 1, 2 \ldots \infty$

As our ML methods assume Gaussian residuals with homoscedastic variance, it is important to pre-transform the data to achieve this.

2.2.2.1 Anscombe transformatiom

The Anscombe transform [7] is a statistical data variance stabilisation transformation, i.e. a transformation which aims to change the input data so that the variance of each data point or observation is not related to its magnitude. This transformation therefore transforms heteroscedastic Poisson [74] distributed data into approximately Gaussian distributed data with a homoscedastic standard deviation of one. This then allows us to appropriately apply ML algorithms that assume Gaussian variance, which is the vast majority of them. We consider a set of values x in a data set, the Anscombe transform A of x is defined as:

$$A(x) = 2 \times \sqrt{x + \frac{3}{8}} \tag{2.2}$$

2.2.2.2 Log transformation

We want to detect relevant bursts of activity as they may contain key information. In our use case, the percentage change in activity is more important than the absolute change, i.e. an increase in activity from 1 to 2 counts (100% change) is far more important than from 100 to 101 counts (a 1% change). The Logarithm function is a mathematical device to turn multiplicative change into additive change so that the ML can consider multiplicative effects symetrically. Note that applying the Log transform on the output of the Anscombe transform does not cause problems because it is akin to a simple division by two, the Log of x at the power 1/2 is half of the Log of x: $\log(x^{1/2}) = \frac{1}{2}\log(x)$.

2.2.3 Visualisation

A web-based online visualisation tool Fig. 2.7 was created in Python. For this work, we parsed the raw data into an SQL database. The data was re-binned into multiple time resolutions (Δt) for efficient interactive visualisation with zooming capability. The tool displays the histogram of the accelerometry data and the Frequency domain representation of the selected transponders (Continuous Wavelet Transform [53] or Short-time Fourier Transform [128]). Exogenous factors such as temperature and humidity can also be visualised with the accelerometry data. The tool also allows visualising the data from two transponders at the same time via a mirrored axis. Furthermore, visualisation of different pre-processing steps is also possible. *visualisation tool sources*.

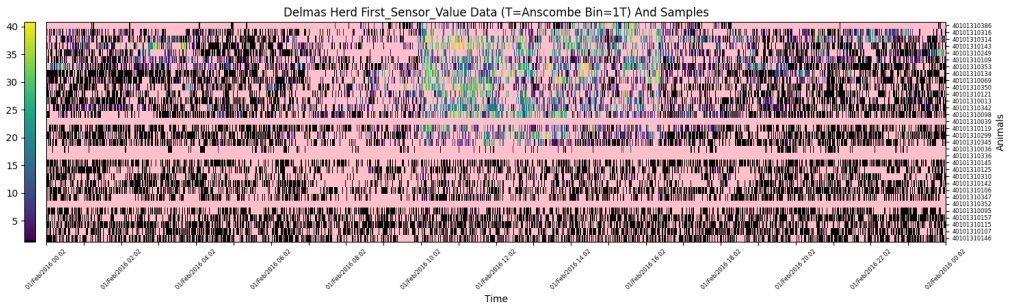
This tool was developed for exploratory analysis to allow us to determine whether transponders were faulty and failed to transmit, and provided visual verification of expected behaviours such as decreased activity levels at night. The visualisation also revealed the need for data pre-processing, as changes in average activity signal amplitude across the herd were observed due to varying calibration of the sensors. Additionally, the sensor's measurement of acceleration was influenced by the mounted position on each animal. For instance, a looser collar would permit more extensive sensor movement, resulting in higher activity values. On the contrary, longer wool significantly tended to decrease the frequency of registered activity occasions.

2.2.4 Need for data imputation

Visualisation of the raw unfiltered activity data in Fig 2.9 reveals the main problems. The heat maps show the herds where the x axis display the time and the y-axis the transponder unique identification number, each row contains the corresponding transponder activity count data, the pink color highlights missing activity data points. The animals are kraaled (i.e enclosed) every night and released to the pasture every morning, as part of normal husbandry practice and for health evaluation/monitoring the animals are periodically individually inspected after being guided to an inspection facility. From this visualisation we can observe multiple transponders with no or near to no data points likely due to hardware mis-use or failure but also large time frames where the data is missing for the entire herd likely due to the data administrator forgetting to download the data from the servers. In addition, by zooming into the time axis to a day, in Fig 2.8A we can see missing points likely due to drop of information packets during communication between the transponder and the base station (Fig. 2.2), Packet drops occur when the transponders are out of range. In Fig 2.8B, we can see that a decrease of transponder signal strength (i.e the amount of energy the radio pushes into the air over time) correlate with missing activity points, in other words, when the signal strength decrease too much no activity points are transmitted or recorded. Missing points can also be caused by the base station not being able to process large amounts of incoming data simultaneously.

The heat maps in Fig 2.9 show the raw data collected over the study period for the Sheep farm Delmas and the Goat farm Cedara. Each heat map contains activity data for 13 and 73 transponders for Delmas and Cedara, respectively. The heat map grids are configured so that each row of heat maps represents successive weeks of data from the start to the end of the study. The content of each individual heatmap shows the output (activity count) of each sensor aggregated at a 10 minutes bin, i.e., each bin contains the sum of 10 activity counts as the transponders were set up to a minute time interval.

A – Activity raw counts



B – Signal strength

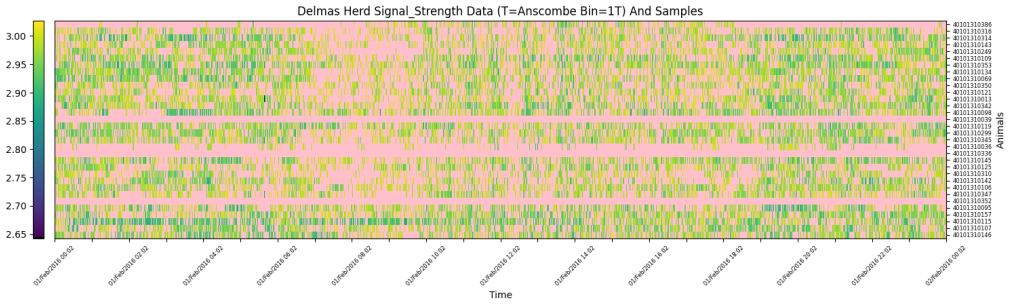


Figure 2.8: **Day of raw herd data**. Heat maps of a subsection of the unfiltered raw data, while (A) shows the activity count of multiple transponders (y-axis) after Log and Anscombe transformation, (B) shows the corresponding Logged transformed absolute value of the signal strength of the transponders. Note that the signal strength is measured in dBm (Decibel per milliwatts ranging from 0 to -100), because we are showing the absolute value of the signal strength which is a negative value, higher values indicate lower signal strength, while lower values correspond to high signal strength..

A **Cedara Goat herd activity**

Figure 2.9

B **Delmas Sheep herd activity**

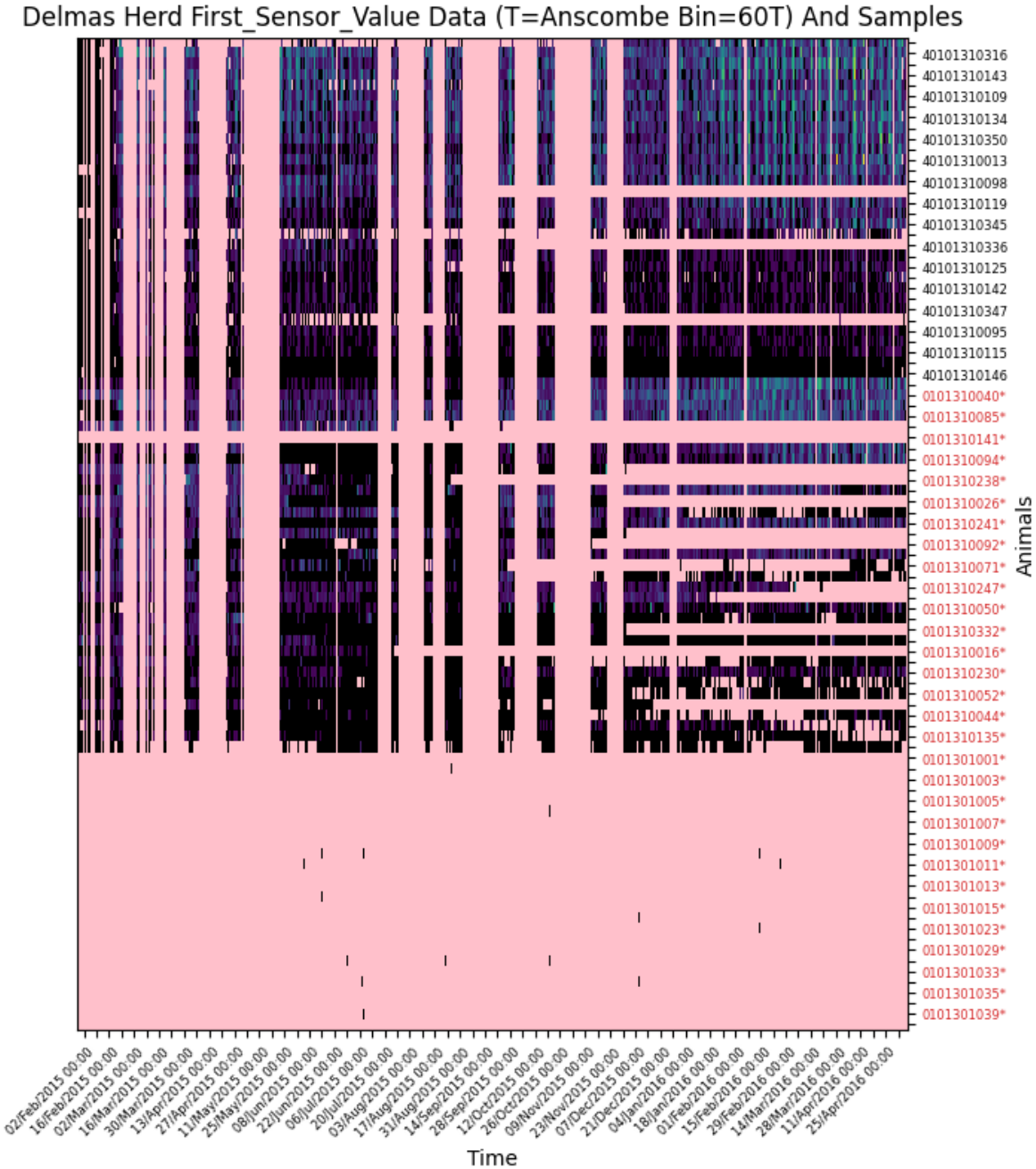


Figure 2.9: **Study group herd raw unfiltered activity**. Illustration of the raw activity counts of all of the transponders (y-axis) in the herd, the x-axis shows the time. The pink colour represents missing data points. While (A) shows the transponders data in the goat farm (B) shows the sheep farm. The vertical white dotted line in (A) marks the date of a transponder swap event on the farm. The transponder on the y-axis in red font are transponders with no ground truth data which can't be used in the later chapter 3 for our supervised ML pipeline.

In chapter 3 our ML pipeline will rely on modeling a day of activity data to predict animal health. The amount of days that do not contain any missing points in our dataset is limited as we highlighted in this section. It is therefore key to have the maximum number of examples of days for the ML pipeline to be able to work optimally, hence the need for data imputation.

2.3 Data imputation

2.3.1 Introduction

Missing values are a reality of modern datasets and a common occurrence in data science. They are caused by a variety of reasons ranging from human error to failure of measurement tools, for example in the medical field it is not uncommon for patients to miss appointments or stop their participation to experimental trials, a data set that logged such patient medical record would have missing record. Data imputation is the process in which we infer the missing values. There are many different ways to impute missing data points from very simple approaches such as padding, i.e replacing missing points with a fixed value or more complex techniques that use sophisticated algorithm to estimate missing points. Ultimately we want to replace the missing values with a value that could be true or close to the real value that was missed, for that reason more advance imputation are usually preferred depending on the use case. We will use simple and more advanced methods to impute our data; linear interpolation in a natural simple approach for time series data as it based on a first order approximation, the equation of a line $y = mx + b$ where m is the slope and b is the intercept, both can be calculated if we have at least 2 data points, thanks to the formula we can estimate any missing data points, the points will be said to be "linearly interpolated". This approach is fast but assumes that the missing points depend linearly on the nearest neighbour non-missing points from the same transponder, and are also de-correlated with all the other transponders.

Fang et al [40] surveyed current (2020) time series data imputation methods and found that deep learning approached based on the Recurrent Neural network algorithm [92] outperformed other approaches. For the more complex methods, we will use two different approaches both based on deep learning algorithms that aim to estimate the conditional distribution of the data across the herd.

2.3.1.1 Generative Adversarial Imputation Nets (GAIN)

GAIN is a missing data imputation technique based on Generative Adversarial Networks (GAN) [159], [24]. GANs are composed of two artificial intelligence models which work against each other. Most supervised ML models are used to generate a prediction. Classically we start by feeding a model input training data, the model, in turn, outputs a prediction, and we can compare the predicted output with the expected output based on the prediction results we can update the model to improve its predictive power. GAN is an unsupervised technique which consists of

two sub-models, a Generator and a Discriminator. The Generator's job is to create fake samples while the Discriminator's job is to take a given sample from the Generator and determine if the sample is fake or real from the original input domain set. As an example to illustrate how GAN works, we might want to generate pictures of human faces. We are aiming to train the Generator to create convincing face pictures. We start by first training the discriminator model to recognise what a picture of a face looks like, hence we need a large dataset of face pictures that we feed into the discriminator which will look at all the attributes that make up a face in order to characterise what a typical face should look like, once the discriminator is capable to recognise real faces then we can test it by feeding it pictures which are not pictures of faces to make sure it can make the difference. During this first phase, the Generator is frozen but once the discriminator gets good at recognising pictures from the original dataset i.e pictures of faces then we apply the generator to start creating fake pictures of faces, by selecting a random sample from the dataset and modifying it to create a fake sample, the next step will be to feed the discriminator the newly created fake sample and check if it is capable to identify the sample as fake, based on the answer either one or the other sub-models will be updated. If the discriminator successfully spots that the sample is fake then it remains unchanged while the generator will need to update its model to generate better fakes, and vice-versa if the generator creates a fake sample that fools the Discriminator, the discriminator model needs to update itself. We repeat the process until the generator gets so good that the discriminator can no longer pick out its fakes.

GAIN operates on tabular data, using a fully connected neural network (Fig. 2.11) for both the generator and the discriminator to model the conditional distribution between the columns, while assuming the rows are independent and identically distributed. In addition the generator is given a hint matrix containing the location of the missing data points along side the input data data that we wish to impute, this ensures that the model understand the true distribution of the data, ie the model has a concept of the distribution of the real data and the missing points, [159].

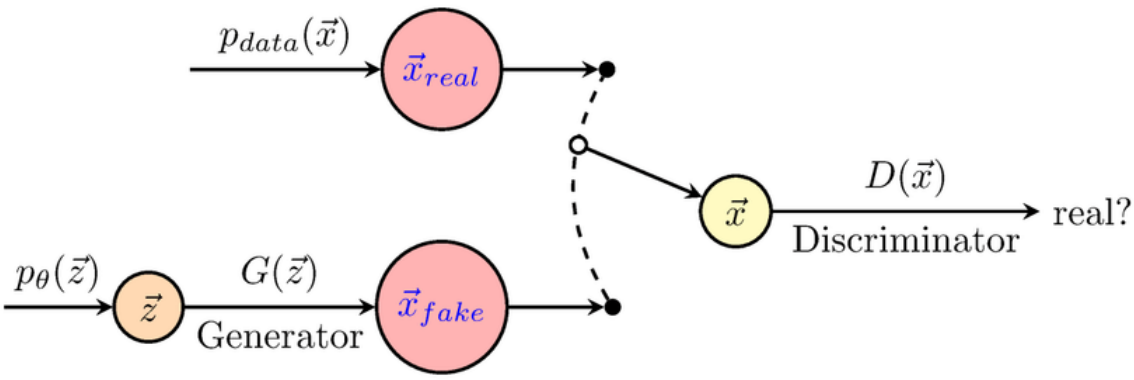


Figure 2.10: Schematic of generative adversarial networks (GANs). The generator (G), defines a probability distribution based on the information from the samples, whereas, the discriminator (D) distinguishes data produced by G from the real data.

2.3.1.2 Multi-directional Recurrent Neural Networks (M-RNN)

In our study, we can take advantage of imputation techniques such as the M-RNN approach devised by Yoon et al [160] which takes advantage of the temporal component across multiple data streams. In our use case, this approach can take advantage of the information contained in all of the transponders in the herd to impute the data because it can model both temporal correlations within each transponder and an orthogonal fully-connected neural network to model correlations across transponders.

Recurrent neural networks [92] RNN are a deep learning technique which predict sequences of data, which is commonly used in sequence modelling problems such as natural language processing and stock prediction where the sequence of the data is important. Traditional RNNs are also called feed-forward neural networks and use the concept of sequential memory by using previous information to affect later ones, this takes the form of a looping mechanism that allows information to flow from one step to the next, and previous information is saved in a hidden state.

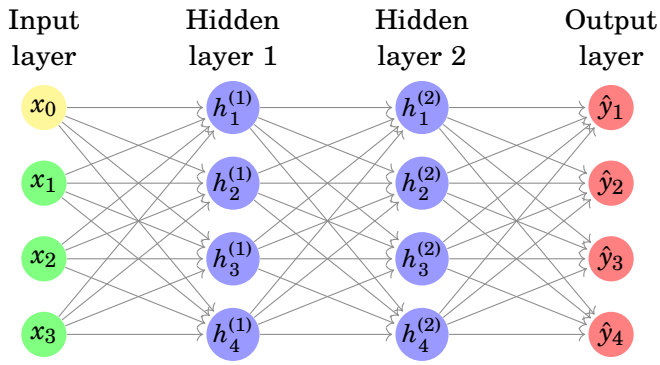


Figure 2.11: **Schematic of a Fully Connected Neural Network**.

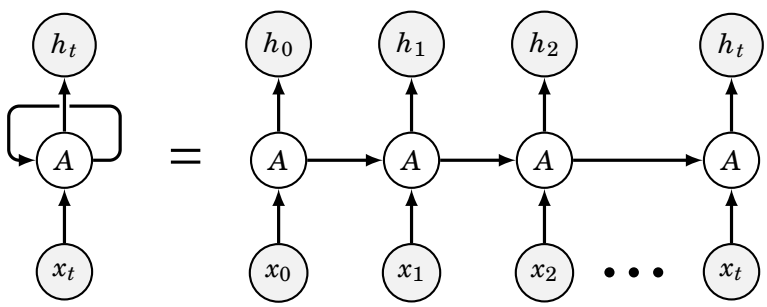


Figure 2.12: **Recurrent neural network diagram** x, A and h are the input, hidden and output layers respectively during the different learning steps, ie for each element in the input sequence x.

Short term memory is a known limitation of RNNs, it is caused by the vanishing gradient problem [61] which is due to the nature of back-propagation [59], training a neural network has 3 major steps, first, a forward pass that makes a prediction, second, a comparison of the

prediction to ground truth using a loss function which output an error value which estimates how "badly" the network is performing, last the error value is used to calculate the gradient in each node in the network through back propagation, the gradient being the value which is used to calculate the network internal weights which ultimately allows the network to learn, the scale of the adjustment to the internal weights of the network is directly linked to the gradient value, the larger the gradient the larger the weight adjustments will be, similarly small gradient values will cause small adjustments. When doing back-propagation each node calculates its gradients with respect to the gradients from the nodes in the previous layer which is a problem as the gradient in the next layer would be smaller after each consecutive layer preventing the earlier layer from "learning" has their weight are only adjusted by a very small amount due to small gradient values.

In an RNN each time step is a layer, to train an RNN back-propagation through time [155] is used. The gradient values will exponentially shrink as it propagates for each time step, which renders the model unable to learn from earlier time steps which are vital in most sequence modelling problems, for example, to predict the next word in a sentence knowing a large number of words is crucial for context, basing the prediction on just a few latter words can cause out of context predictions. To combat the short term memory problem of RNN, two main specialised RNN were created, Long Short Term Memory (LSTM) [62] and Gated Recurrent Units (GRU) [32] which are capable of learning long-term dependencies using mechanism called gates which are different tensor operations that can learn what information to add or remove to the hidden state.

M-RNN is composed of multiple independent bidirectional RNNs (Fig. 2.12), each model the data in an individual stream of data, in our case individual transponder data. In addition, a neural network with a fully connected (FC) layer (Fig. 2.11) is used to model the data across every transponder, where each layer in the FC layer contains the data of a single transponder. This last model can model important data from the herd which improves imputation results (section 2.3.3) at the animal level.

2.3.2 Methods

2.3.2.1 Selecting high-quality training data

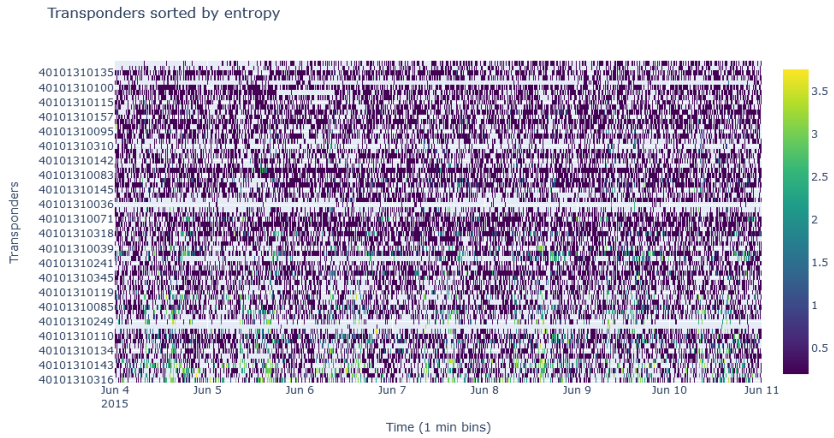
It is important to train our imputation models with high quality data that reflects best the real world activity of the animals. For this research the data collection not only suffered from missing data points but also invalid and weak signals due to the transponders not being used correctly or placed on the animals securely, for example some workers have reported to leave active transponders on a desk for a prolonged amount of time, such events were not logged, for that reason we will analyse the quality of the data to assure that the data that we used for this research is valid. For each transponder, we evaluate the complexity [37] of the accelerometry data with the Shannon entropy [130], in order to select the transponder that contains the densest

information and dismiss the lower quality (i.e, large amount of missingness and abnormal data points) traces (Fig. 2.13). The Shannon Entropy can help us measure the information content in the transponder which ultimately allow us to detect invalid transponders (Fig. 2.14). The two main problems the transponder traces suffer from are repetition of the same activity count value for an extensive period of time and excessive amount of missing points. This approach is effective at detecting both. The Shannon Entropy formula is given by:

$$H = -\sum_{i=1}^{M} P_i \, log_2 P_i. \tag{2.3}$$

Where P_i is the normalised probability distribution of the count in a transponder time series.

A – Sheep farm transponders sorted by increasing entropy



B –Goat farm transponders sorted by increasing entropy

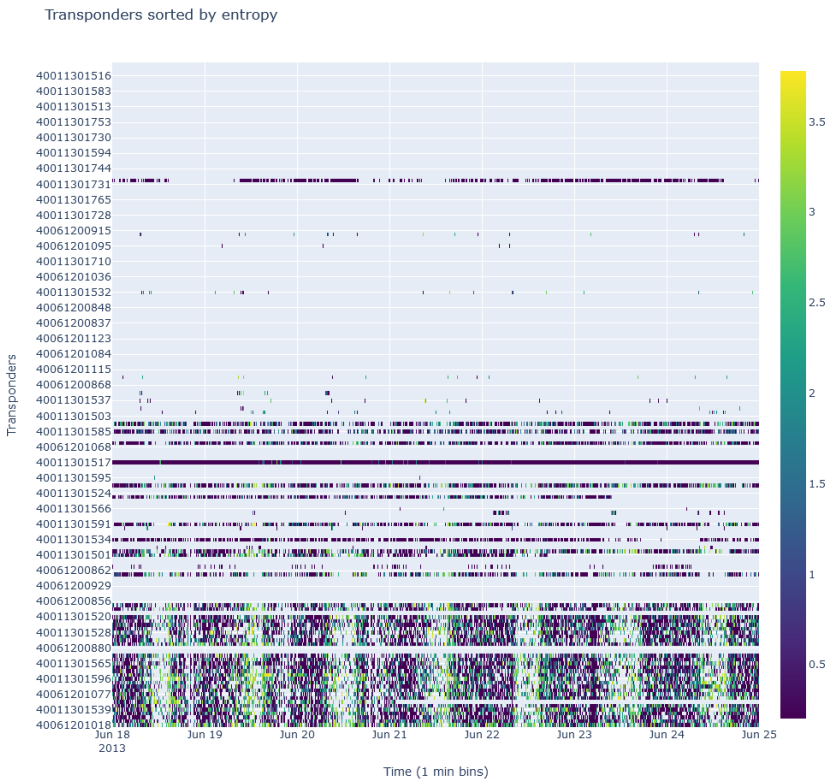


Figure 2.13: **Illustration of transponder data sorted by entropy**. This heatmap shows a 7 day section of all of the transponders sorted with their respective Shannon entropy in ascending order from the bottom to the top of the y axis. The raw activity counts are Anscombe and then Log transformed in this visualisation. We can observe the improvement of the "quality" of the transponders with lower entries. It is important to note that this illustration only shows a subsection of the time frame, transponders with no data in this subsection do not necessarily have a lot of missing points in other time frames. Missing points are displayed via the light turquoise color.

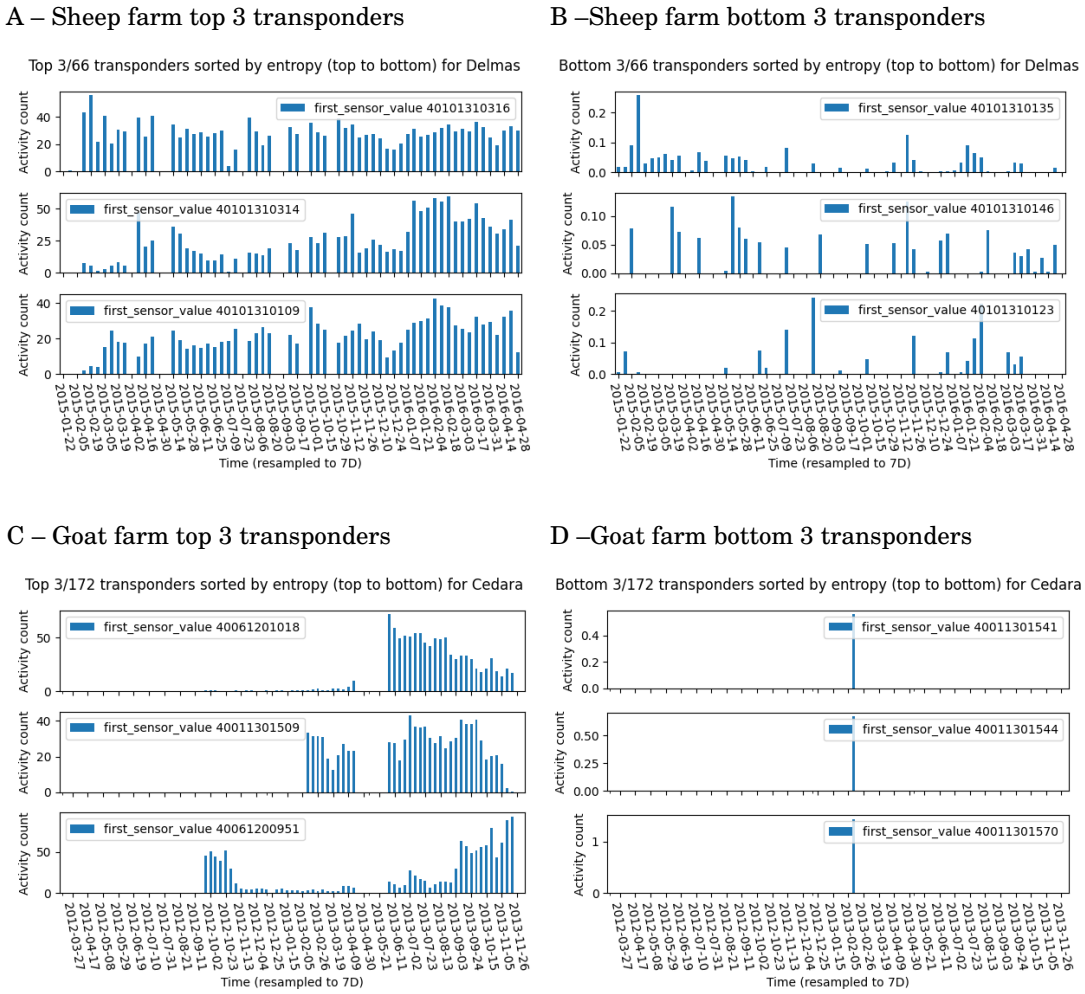


Figure 2.14: **Overall view of transponder data quality**. This figure shows the top 3 and bottom 3 transponders based on their respective data entropy. For visualisation convenience we re-sampled the data to a 7 days resolution by summing all data points within a week. While (A) and (B) show the Sheep transponders (C) and (D) show the Goat transponders.

2.3.2.2 Creating input samples

Here we will illustrate with a simple example how we will create the input samples for the GAIN and M-RNN techniques (Fig. 2.15). In this example the accelerometry data come from 2 different transponders, each collected twelve data points over time, transponder 1 could not measure accelerometry data 4 times at time $t4$, $t9$, $t11$ and $t12$ while transponder 2 only missed the data point at time $t5$. In both case we need to introduce the concept of window, i.e. a subset of the original input data of a variable length, in our example we used a window length of 4 points (in our study the data is sampled to a minute resolution, hence 4 points correspond to 4 minutes). For GAIN for each transponder we build samples by sliding the window along its time axis with

a stride equal to the window length, in that fashion we can build 3 samples per transponders for a total of 6 samples for the GAIN algorithm. In comparison for M-RNN we slide the widow along the time axis common to every transponders so that each samples contains the data of all transponders at all times, in our case we could build 3 samples. The key difference is that M-RNN accepts and models a matrix of data for each sample, wherease GAIN only models a vector.

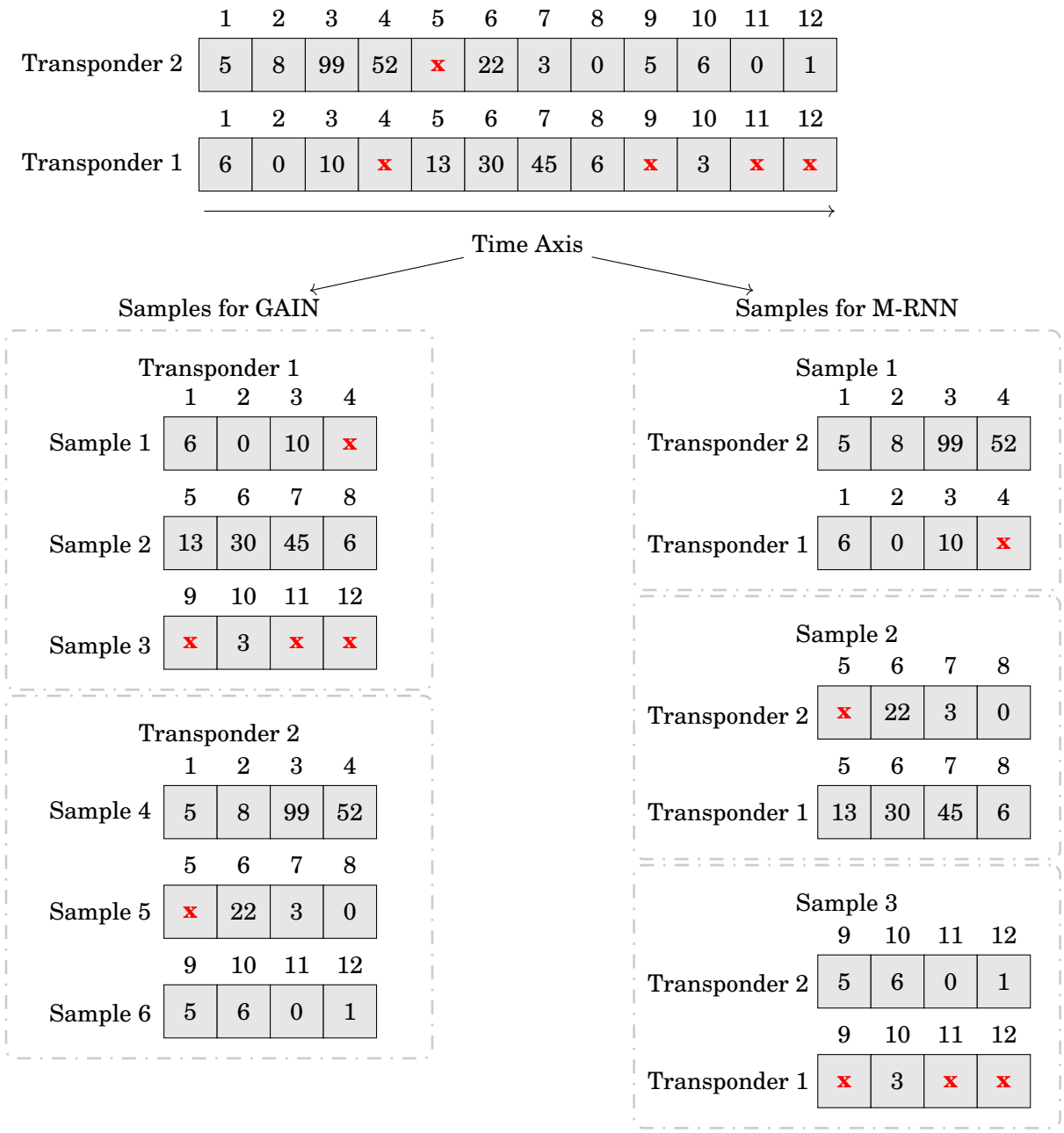


Figure 2.15: **Explanation of the data preparation**.This schematics shows how the samples are created for the two main imputation techniques we used. Generative Adversarial Nets Imputation (GAIN) [159] does not not inherently have a concept of the entire timeline of the data stream while Multi-Directional Recurrent Neural Network (M-RNN) [160] natively uses the timeline, the natural temporal order of the samples is taken into consideration.

The pipeline to implement the GAIN and M-RNN algorithms are illustrated in Fig. 2.16 and explained in detail in the following subsections.

2.3.2.3 GAIN implementation

For our application, the aim of GAIN is to create convincing fakes of our activity samples which can be used to impute our data set. We used the implementation developed by Yoon et al [159] and adapted to our use case. Below, we describe the different steps (Schematic. 2.16) to impute the missing points in the original accelerometry traces. We aim to train the GAIN model with the accelerometry data recorded during a single or multiple days, which will allow us to capture and model the activity data of the selected time frames, we will investigate the optimal length or time frame to use given our data set.

We consider a data set of N animals. For each animal a we have a data stream of length T where each data point is measured at the time t.

$$Input(N,T) = \begin{bmatrix} a_0(t_0),\ldots,a_0(t_T) \\ \ddots \\ a_N(t_0),\ldots,a_N(t_T) \end{bmatrix} \tag{2.4}$$

We want to train a GAIN with samples that contain accelerometry data and other features F that can help the algorithm to optimally model the data, in our case we want to model what a day of activity looks like. F is an array that holds the timestamp of the day and the unique transponder id associated with the accelerometry data. By stacking days of activity data from multiple transponders GAIN can learn across the entire herd what a typical day of activity looks like.

$$samples(N,T) = \begin{bmatrix} a_0(t_0),\ldots,a_0(t_{1440}),F_0 \\ a_0(t_{1440}),\ldots,a_0(t_{2880}),F_0 \\ \vdots \\ a_0(t_{T-1440}),\ldots,a_0(t_T),F_0 \\ \vdots \\ a_N(t_{T-1440}),\ldots,a_N(t_T),F_N \end{bmatrix} \tag{2.5}$$

- Training
  1. Raw data cleaning (Section. 2.2.1).
  2. Read each transponder file and filter out transponders that do not have at least 100 data points.
  3. Sort transponders by entropy.
  4. Calculate the log (2.2.2.2) of the Anscombe (2.2.2.1) of every data point.
  5. Chunk all transponder traces into n-day-length chunks and append the day timestamp and the transponder id for each chunk. (Eq. 2.5)

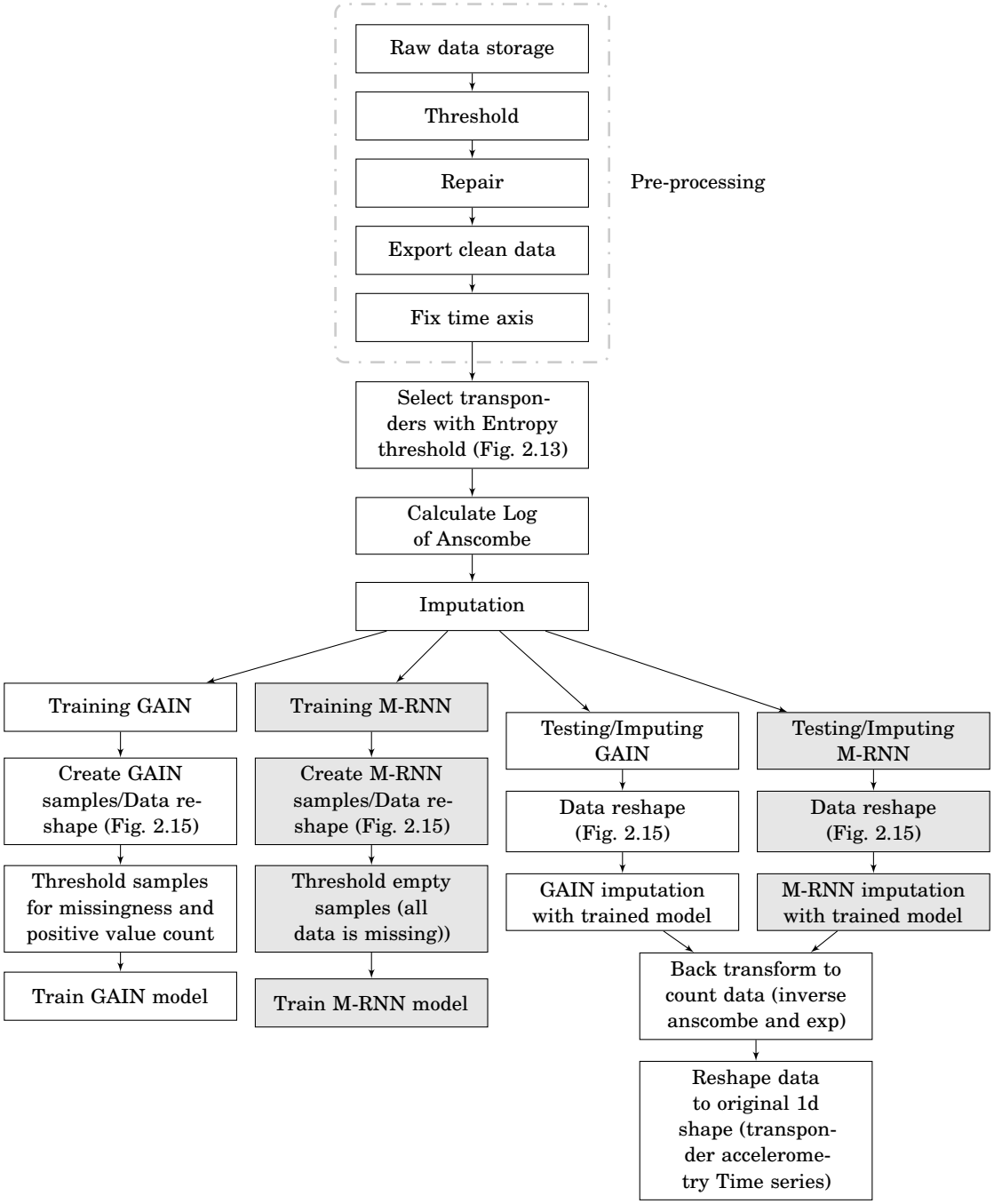


Figure 2.16: **Schematic of the GAIN and M-RNN imputation pipeline**. This flowchart illustrates the key steps for each imputation approach.

6. Filter out days where more than 80% of the data points are missing. There is 1 accelerometry data point per minute, therefore 1440 points in a day, if more than 1152 points are missing the chunk is dismissed.
7. The data is normalised in the range of 0 to 1.
8. The empty data points are masked and replaced with 0.
9. Train the GAIN model with high-quality n-day-length samples (Fig. 2.25).

- Testing

1. Reshape the accelerometry traces we wish to impute into n-day-length chunks.
2. Apply the same preprocessing approach used during training, namely calculating the Log(2.2.2.2) of the Anscombe (2.2.2.1) of all data points.
3. Feed the trained GAIN model with the n-day-length accelerometry trace to impute the missing point.
4. Back-transform the data point into accelerometry counts by calculating the inverse of the Anscombe and Log transform and reverse Normalisation from the range 0 to 1.
5. Restore the imputed data into its original shape of 1 accelerometry trace per transponder.

2.3.2.4 M-RNN implementation

In practice, M-RNN takes a list of 2D time series as input, in our case each time series correspond to the raw count activity data of the transponders in the herd. We consider a dataset of N animals. For each animal a we have a data stream of length T where each data point is measured at the time t (Fig. 2.17). Compared to GAIN where each transponder is treated as an independent data stream, M-RNN uses groups of sensors aligned on the same time axis, in other words, the herd activity needs to be preserved to benefit from this technique. Based on the amount of available data, we aim to model what a week of herd accelerometry data looks like, the fitted model would then be able to impute and predict what a week of accelerometry data should be based on the entire information contained in the herd data.

$$HerdData = \begin{bmatrix} a_0(t_0), \ldots, a_0(t_T) \\ \ddots \\ a_N(t_0), \ldots, a_N(t_T) \end{bmatrix} \tag{2.6}$$

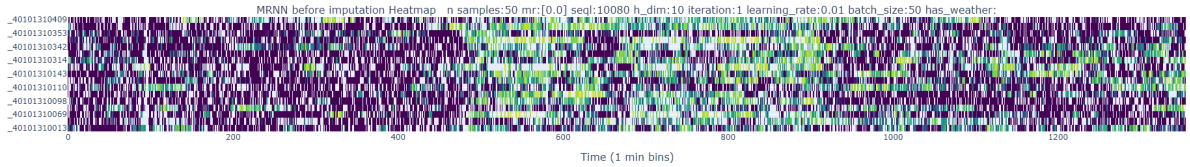


Figure 2.17: **This figure shows a subset of the accelerometry input data for M-RNN** each row shows a 1440-minute long stream of activity data after Anscombe and Log transformation where each data point is measured every minute.

Because M-RNN uses the mean square error as a cost function we first apply the Anscombe (2.2.2.1) and the Log (2.2.2.2) transformation to make the native Poisson distributed count follows a normal distribution.

To train the M-RNN model training sets are built by creating consecutive subsections of the input data, each i subsection $w(N,t)_i$ is of fixed length s where $t_s - t_n = s$ and n is incremented by s. Here w is the accelerometry data component of the sample where each row value comes from a different transponder, each sample is constructed by creating 3D matrix where the first matrix is w_i, a second mask matrix that will hold the location of the missing points in w_i, and a final matrix that for each row of w_i (Fig. 2.19).

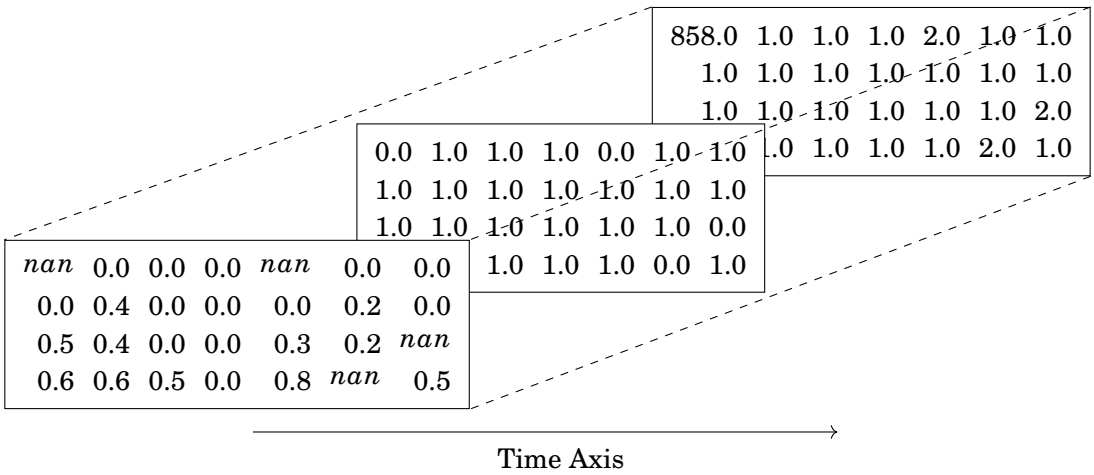


Figure 2.18: **Schematic of the M-RNN imputation 3D sample)**. In this illustration, for all matrices each row corresponds to the data stream from a single transponder, in this example, there are 4 transponders data streams. The columns represent time, there are 7 consecutive accelerometry records here. The first matrix (in the front) shows the accelerometry data, and the missing points are marked by the word 'nan'(i.e not a number). The other matrices are calculated based on the first. The matrix in the middle is a mask of the first one while the third one shows how many data points were missing before a new data point is recorded by the transponder, in this example the top transponder could not record 858 values before it could record a real data point (0).

$$w(N,t)_i = \begin{bmatrix} a_0(t_0),\ldots,a_0(t_s) \\ \ddots \\ a_N(t_0),\ldots,a_N(t_s) \end{bmatrix} \ldots \begin{bmatrix} a_0(t_s),\ldots,a_0(t_{s*2}) \\ \ddots \\ a_N(t_s),\ldots,a_N(t_{s*2}) \end{bmatrix} \ldots \begin{bmatrix} a_0(t_{s*i-1}),\ldots,a_0(t_{s*i}) \\ \ddots \\ a_N(t_{s*i-1}),\ldots,a_N(t_{s*i}) \end{bmatrix} \tag{2.7}$$

- Training

  1. Raw data cleaning (Section. 2.2.1).
  2. Calculate the Log (2.2.2.2) of the Anscombe (2.2.2.1) of every data point.
  3. The data is normalised in the range of 0 to 1.
  4. Build and reshape the entire data into multiple 3D matrices (activity, mask, time) (Fig. 2.18) of 7 days long with a sliding window along the time axis, we used a 7 days stride with no overlap.
  5. Filter out the windows that do not contain any activity data.
  6. Fitting all the RNNs (one per transponder)
  7. Fitting the Fully connected layer

- Testing

  1. Reshape data into 3D matrices (activity, mask, time) (Fig. 2.18).
  2. Feed M-RNN with the reshaped data.
  3. If data points are missing in all streams impute with the data from the mean window.
  4. Revert MinMax Scale normalisation and inverse Log (Apply natural exp) and Anscombe transform
  5. Restore original data traces shape

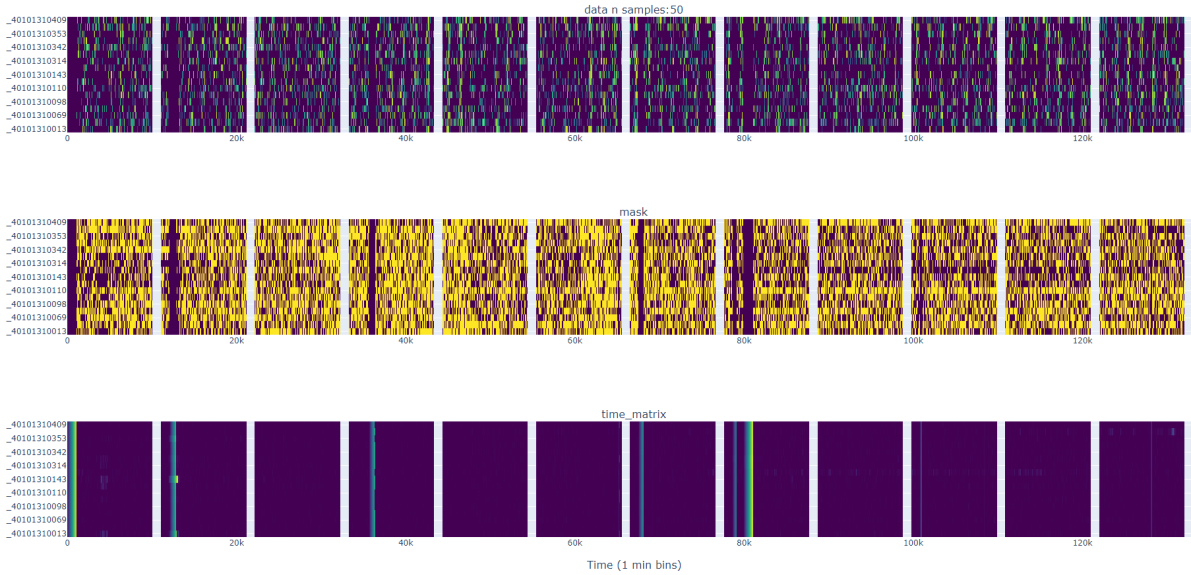


Figure 2.19: **Example of the training set for M-RNN** with real Sheep accelerometry data. In this example, we show 12 samples ready for training a M-RNN imputation model. Each column corresponds to the 3 elements of a 3D matrix (Fig. 2.18) from the top to the bottom, first the activity data, then the mask data, and finally the "time since missing point" data.

2.3.2.5 Evaluation

We can evaluate the results from the GAIN and M-RNN imputation by adding missing data points and comparing the imputed point with the original value of the same transponders. We will use the root mean square error (RMSE) [20] for that purpose and we will compare the imputation results with imputed values achieved via linear interpolation of the data on the time axis. Lower RMSE values are better as the indicate that the imputed value is closer to the real value. In this section we will present the results of different imputation experiments to establish the optimal methodology and parameters. For the GAIN imputation we will investigate the effects of the filtering threshold, the sample length (window size) and the effect of adding extra features in the training samples. For the M-RNN imputation we will experiment with the same parameters but also monitor the impact of using different number of data stream for training.

2.3.3 Results

The experiment shows that the M-RNN-imputed data points are closer to their original value compared to the GAIN and linearly imputed one, which suggests that M-RNN is more accurate at predicting missing values (Fig. 2.21).

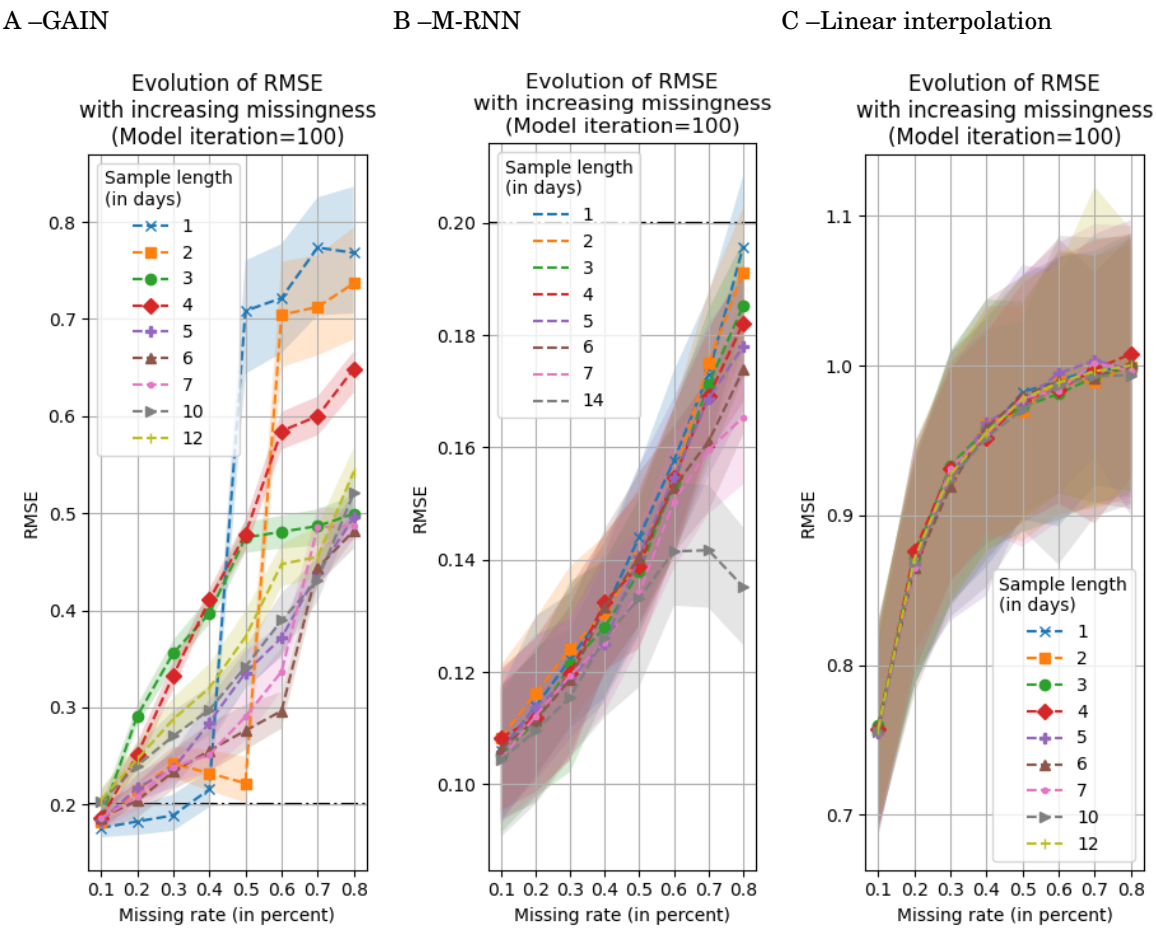


Figure 2.20: **Evolution of RMSE with increased missingness for GAIN and M-RNN**. The x-axis shows the missing rate, i.e the rate of artificially added missing points. The y-axis shows the root mean square error (RMSE). While (A) shows the results for GAIN (B) shows the results for M-RNN and (C) Linear interpolation.

2.3.3.1 Robustness toward missingness

By adding missing points to the original samples we can evaluate how resistant to missingness the imputation the imputation is. This can also give us a clear indication of which technique is most powerful for our use case. Figure 2.20 shows that M-RNN performs better for all sample length compared to GAIN.

2.3.3.2 Sample length and RMSE

Ultimately we want to find the optimal training configuration for each of the algorithms before comparing their outputs, we can use the RMSE to determine the best sample length to use. Here will add 40% of missingness, i.e 40% of the dataset points will be marked as missing (Fig. 2.21C).

A –GAIN B –M-RNN C –Linear interpolation

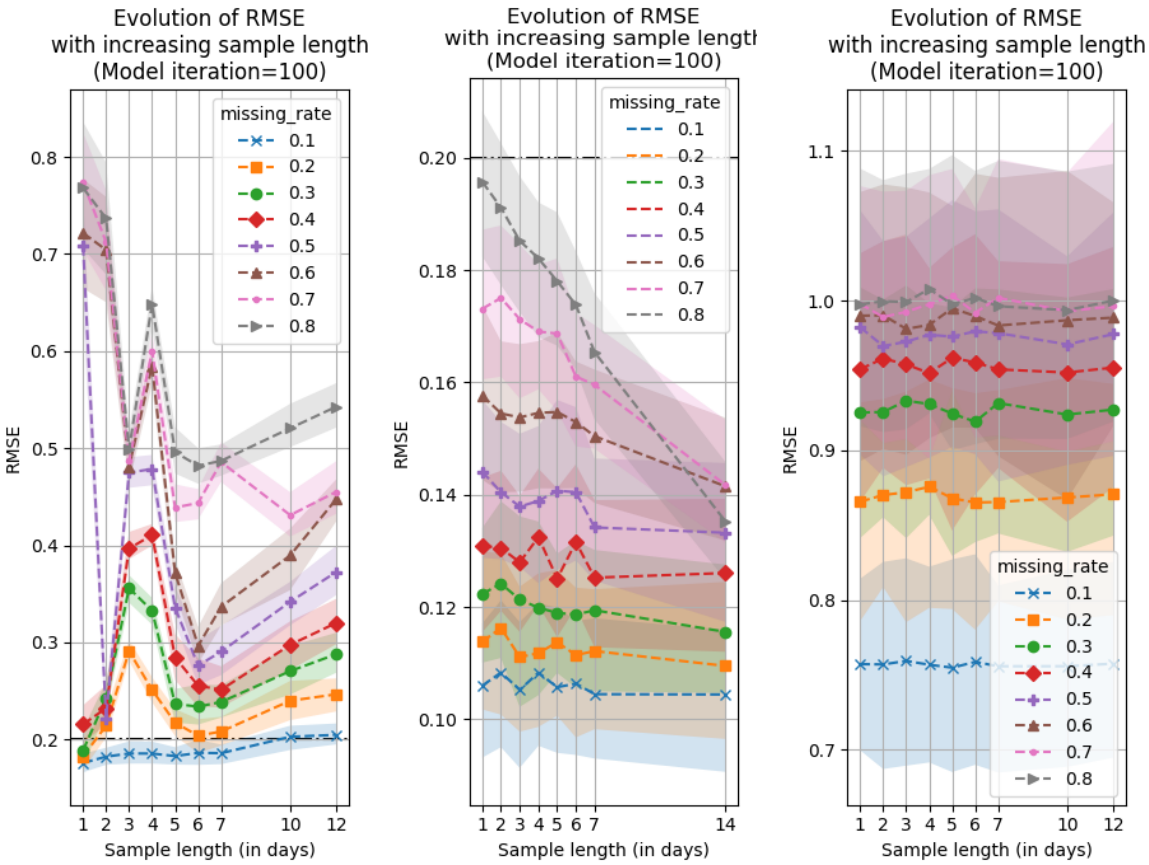


Figure 2.21: **Evolution of RMSE with increased sample length for GAIN and M-RNN**. The x-axis shows the sample length in days, i.e the number of activity data points in the sample, for example 1-day samples contain 1440 data points, 1 per minute. The y-axis shows the root mean square error (RMSE). While (A) shows the results for GAIN (B) shows the results for M-RNN and (C) Linear interpolation.

For the GAIN imputation sample length of 1 day gives us a RMSE slightly below 0.2 while other sample length give values > 0.2. Similarly 1 day give the lowest RMSE at < 0.12 for M-RNN. For both techniques increasing the sample length minimally increase the RMSE. Linear interpolation perform the worst with a RMSE around 0.60 for the lowest level of missingness.

2.3.3.3 M-RNN training and RMSE

We can train the M-RNN model with a subset of high entropy transponders to ensure we are using the best data for training our models, however we need to be cautious about over curating our data as some level of noise is realistic on the field scenario, for that reason we will train the models with different number of transponders and evaluate the effect on the RMSE. An

added benefit of training on a subset of the transponders significantly decrease processing time as M-RNN involves training as many Bi-RNN as transponders used for training.

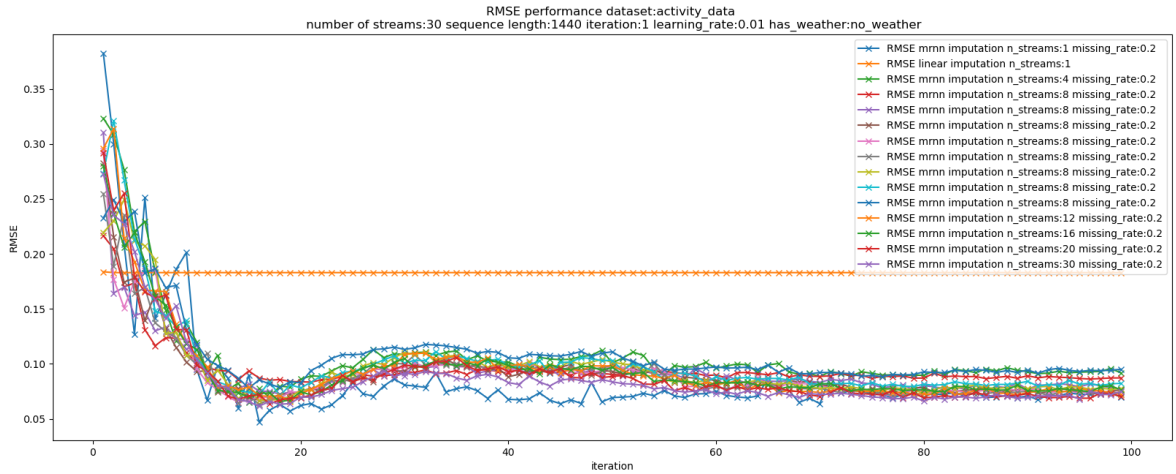


Figure 2.22: **Comparison of The root mean square error of M-RNN** with training on different number of transponders (data stream).

2.3.3.4 Model loss curves and sample length

Analysing the model loss function is a valuable approach to gauge the learning progress of the model. Regarding the GAIN models, one can anticipate the divergence of the two competing models as the generator enhances its ability to generate fakes, while the discriminator becomes more adept at detecting them (as detailed in Section 2.3.1.1). Similarly, for M-RNN, an expectation is set for the model's loss to decrease progressively as it undergoes the learning process. Notably, our experiment aligns with and validates both of these expectations.

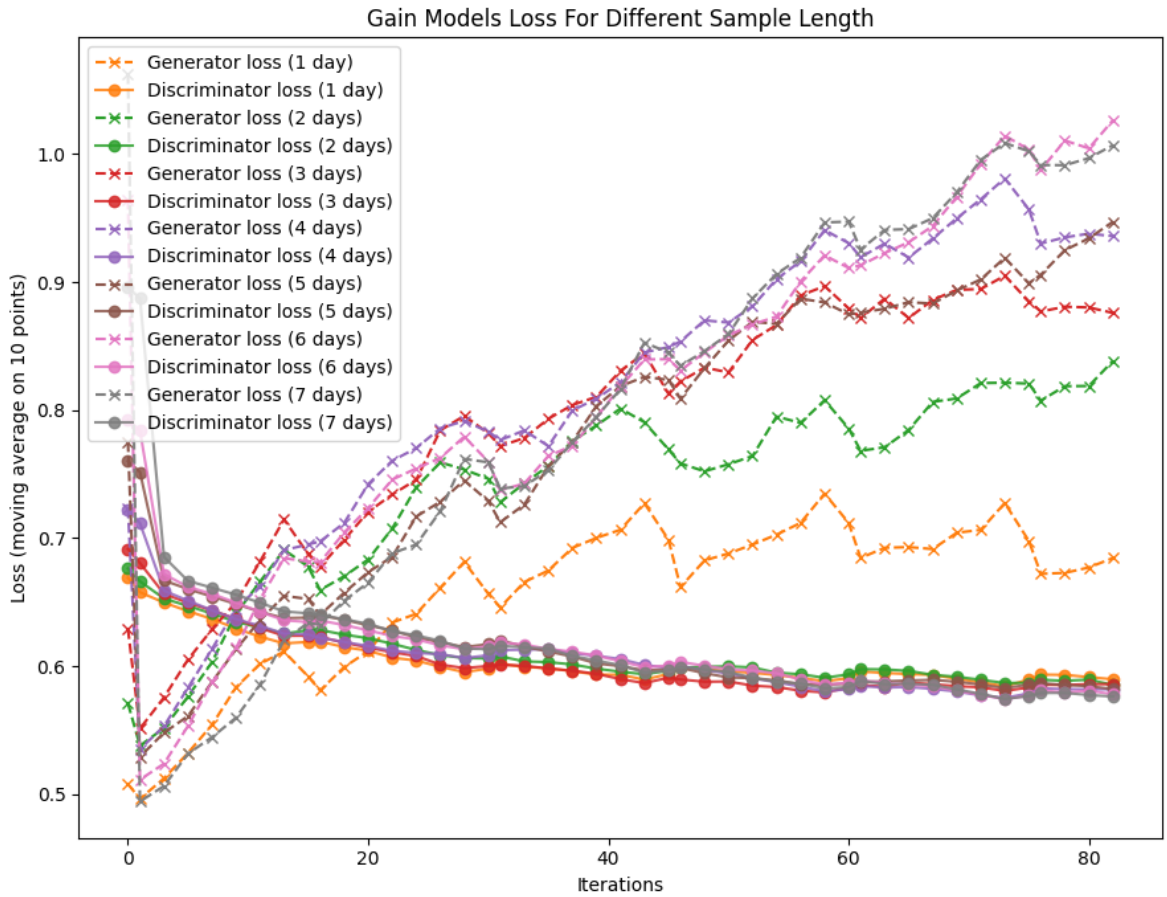
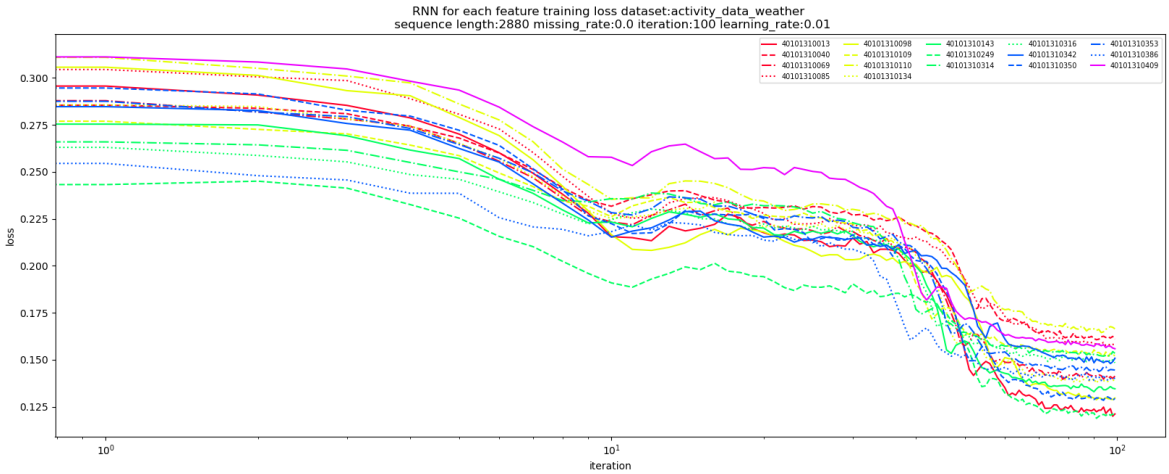


Figure 2.23: **Evolution of RMSE with changing sample length.** The x-axis shows the model iteration while the y-axis shows the moving average of the model loss. Each GAIN model (Generator and Discriminator) share the same color, the generator curves are marked with the "x" marker while the discriminators are marked with "•". Different colours represent different sample length.

A – Bi-RNN Models loss



B – FC layer model loss

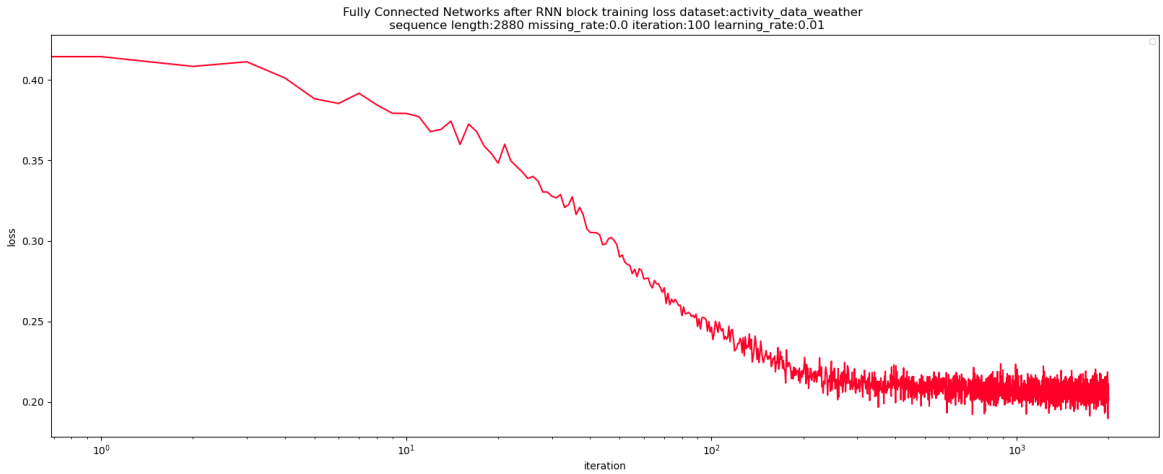
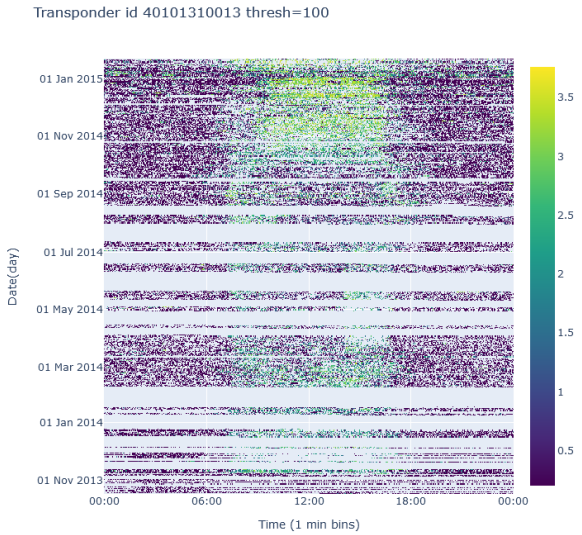


Figure 2.24: **M-RNN models loss**. (A) shows the loss of each Bi-RNN while (B) shows the loss of the fully connected layer. The x-axis shows the model iteration and the y-axis shows the loss. In (A) each curve correspond to a Bi-RNN model learning the data conditional distribution of a single transponder, we can observe that some model converge faster than others.

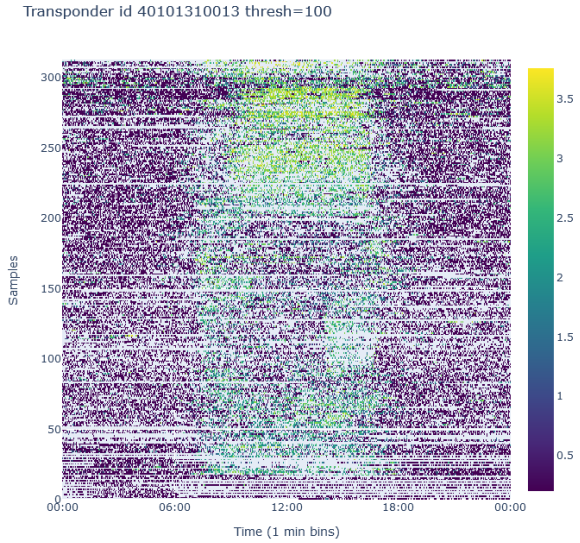
2.3.3.5 Imputation visualisation

In this section we will display and compare the output of the imputation with the help of heat maps. We will compare the output of GAIN and M-RNN with Linear interpolation. In Figure 2.25C we can observe that GAIN have successfully learned the structure of a day of activity, particularly for the samples range $[0,\ldots,20]$ which real values are low, in contrast the linear interpolation approach Fig 2.25D has no concept of a typical day structure. Similarly Figure 2.26 highlight how well M-RNN learned the data structure of each transponder as it can impute entire chunk of missing herd data, for example after 9.5k minutes of the x-axis.

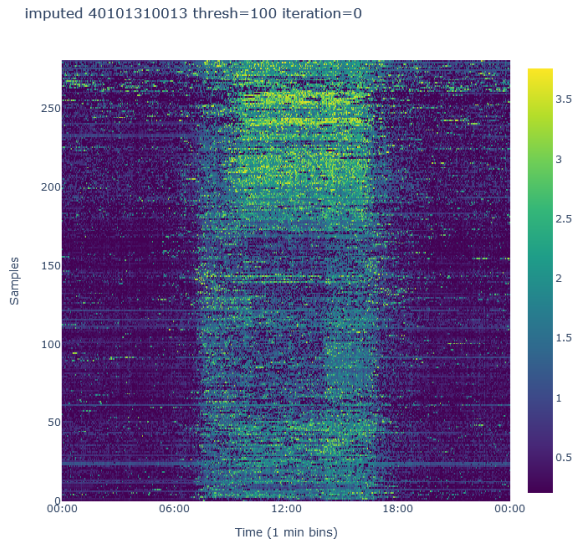
A – GAIN samples for a single transponder



B – Before GAIN imputation



C – After GAIN imputation



D – After Linear interpolation imputation

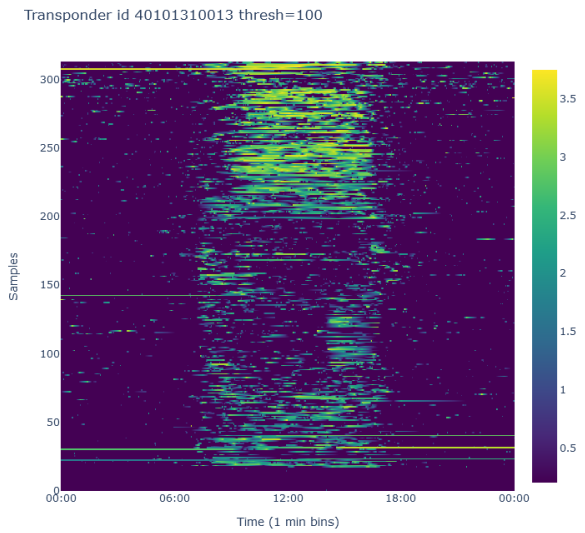


Figure 2.25: **Illustration of GAIN training samples from a single transponder for 1 day length samples.** For visualisation purposes, the samples are placed on the entire transponder data collection time frame which displays the period of emptiness in light turquoise (A). (B) shows ready-for-training samples for a single transponder while (C) displays the GAIN imputed output and (D) shows the linear interpolated samples.

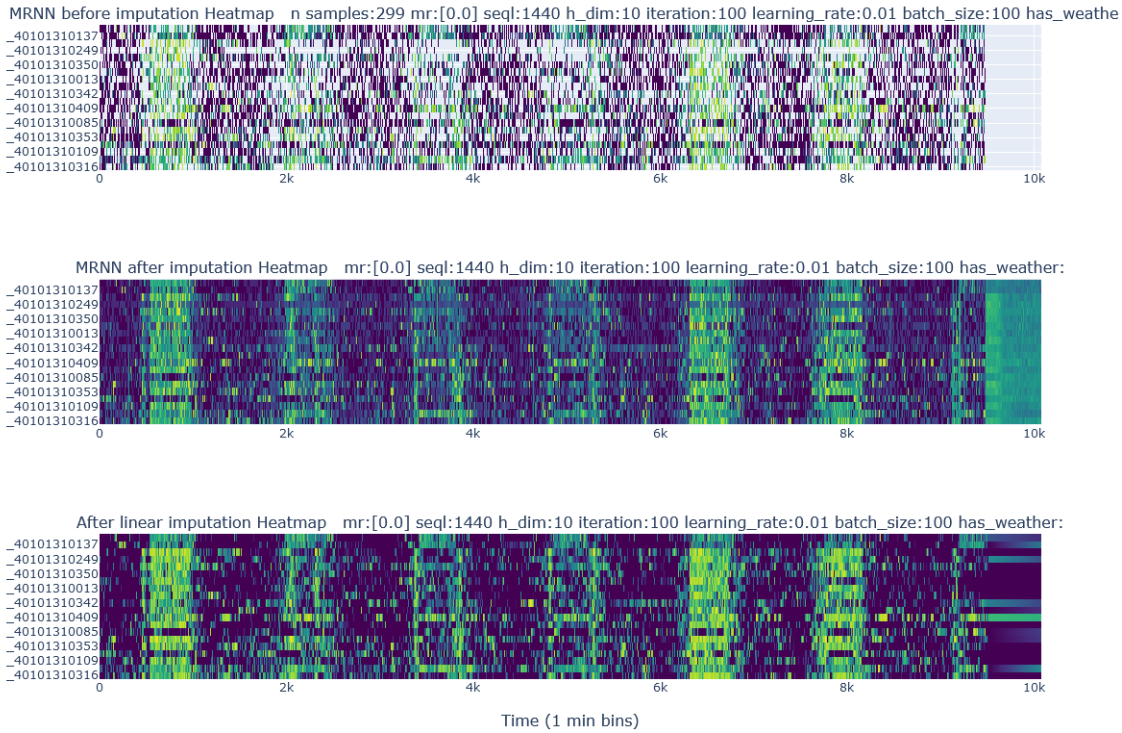


Figure 2.26: **Illustration of M-RNN imputation on a 7 days subset of herd data.** The top heatmap shows the herd data where each row is the data stream from a single transponder where the missing points are highlighted in light turquoise. The middle heatmap shows the M-RNN imputation results while the last heatmap shows the original data imputed with linear interpolation within each transponder trace.

2.3.4 Discussion

As expected linear interpolation performs the worst with RMSE values increasing faster with increased missingness compared to GAIN and M-RNN, it also has the highest RMSE of all approaches at every level of missingness.

Increasing the number of transponders used for training did not negatively affect the RMSE (Figure. 2.22), results were similar when training with 1 4, 8, 12 16, 20, or 30 transponders. This may indicate that a single transponder contains enough information to impute the herd, even though each transponder record the individual data of an animal, the herd behaviour is still contained in its activity data.

By adding more days in the samples we could model structures between consecutive days or even consecutive month if we used month sized samples, in our use case we increased the number of days up to 7 but the results in Figure 2.21 have shown that this advantage is negligible likely due to lower number of samples when using longer time frames.

2.4 Conclusion

In this Chapter we have presented the study group, the nature of our data and highlighted the need for imputation. We tried different methods to impute the missing activity data points in our data sets as it is vital to use as much data as possible without compromising the quality of the data for the chapter (3) in which we will create new samples composed on activity data labeled with the health status of the animal. We used two deep learning based imputation algorithms GAIN and M-RNN, for both the results have shown that using day-length samples for training allow the models to learn the real data conditional distribution best for accurate imputation. The M-RNN impuation gave the best results and was robust to high amount of missingness.

CHAPTER 3

Machine learning pipeline for predicting health status

In this chapter we develop and validate a supervised ML pipeline for the classification of developing ill-health in goats and sheep based on the preprocessing and imputation workflow established in chapter 2. We conducted temporal validation for the sheep farm, training models on different periods and testing their predictive capabilities. Cross-farm validation was explored to assess the classifier's ability to generalise across goat and sheep farms without retraining. The evolution of predictive power over time was studied, focusing on how well the model could forecast ill health as we moved back in time. Additionally, model explainability was addressed by training a linear SVM model and extracting coefficients to understand the importance of features over different time frames, specifically during daytime and nighttime. The chapter concludes with insights into the generalisability of the model, impact of exogenous factors like weather data, and the importance of fine-tuning hyperparameters for future improvements. The findings emphasise the potential of simple, low-cost telemetry systems for health monitoring in resource-poor farming environments.

As an exemplar for ill-health we use parasitic nematode *Haemonchus contortus* infection, and its manual assessment with the FAMACHA (FAffa MAlan CHArt) scoring system in particular. We develop and validate a supervised ML pipeline to classify the health status associated with a time series of activity counts or a combined time series containing activity count data and exogenous factors such as temperature and ambient humidity. The pipeline starts with the creation of the samples that will be used for the ML algorithms. Each sample is made out of an array of features and a target or ground-truth value corresponding to the FAMACHA transition of the animal for the feature set. Various preprocessing steps are tested to determine the optimal predictive model.

3.1 Introduction

The FAMACHA scoring system has been used successfully to detect anaemia caused by infection with the parasitic nematode *Haemonchus contortus* in small ruminants and is an effective way to identify individuals in need of treatment [12, 143].

FAMACHA scoring works by comparing sheep or goat conjunctivae (the mucous membrane that covers the front of the eye and lines the inside of the eyelids) against a colour calibrated chart as seen in Fig 3.1. FAMACHA scores range from 1 to 5, with score 1 being the most healthy, equating to haematocrit (measure of the proportion of red blood cells in your blood) $\geq 28\%$, and score 5 the most severely anaemic (haematocrit $\leq 12\%$). The scoring system requires minimal training, provides immediate results and does not rely on expensive equipment or laboratory analysis. However, the training is specific and the trainers are mostly in short supply, particularly in resources poor regions. However, assessing FAMACHA is labour-intensive and costly as individuals must be manually examined at frequent intervals, therefore a low-cost system that could automatically predict parasite burden may be of particular use.

A – FAMACHA card

B – Live FAMACHA testing

Figure 3.1: **FAMACHA testing on the field**. (A) shows the FAMACHA guide card while (B) shows a live test on the field on a sheep in South Africa while the guide card is displayed on a smartphone.

As described in Section 2.1, 31 female adult *Ile de France* sheep ewes at Delmas and 64 goats at Cedara were individually FAMACHA-evaluated [143] at weekly or fortnightly intervals, respectively, and had associated longitudinal accelerometry data recorded within the study period. This was conducted as part of routine farm management rather than within the defined study, which was focused on trialling the feasibility of the telemetric activity monitoring system hardware.

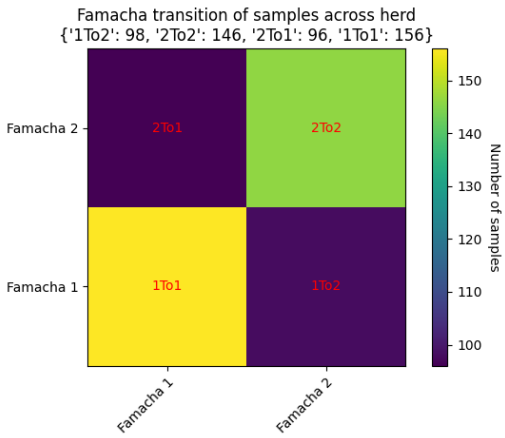
Our interest is to detect changes in FAMACHA for individuals before manual assessment. As FAMACHA scores are in the range $[0\ldots5] = \{1,2,3,4,5\}$ there are theoretically 25 possible

FAMACHA transitions. However, in practice samples with a FAMACHA score ≥ 3 are rare because animals need to be treated when they reach a score of 2, as illustrated by proportion of transitions (Fig. 3.2) and number of transitions (Table. 3.1) for both farms.

Table 3.1: **Proportion of FAMACHA samples per farm**

FAMACHA transition	Cedara	Delmas
FAMACHA 1→1	20%(79)	33.4%(414)
FAMACHA 1→2	11.6%(46)	19.2%(238)
FAMACHA 1→3	3.5%(14)	0%
FAMACHA 1→4	1.5%(6)	0%
FAMACHA 1→5	0%	0%
FAMACHA 2→1	13%(50)	18.4%(229)
FAMACHA 2→2	19.9%(79)	28.6%(355)
FAMACHA 2→3	7%(29)	0%
FAMACHA 2→4	2.3%(9)	0%
FAMACHA 2→5	1%(5)	0%
FAMACHA 3→1	5.3%(21)	0%
FAMACHA 3→2	5.5%(22)	0%
FAMACHA 3→3	2%(9)	0%
FAMACHA 3→4	0.5%(2)	0%
FAMACHA 3→5	0.2%(1)	0%
FAMACHA 4→1	0.5%(2)	0%
FAMACHA 4→2	3%(12)	0%
FAMACHA 4→3	0.5%(2)	0%
FAMACHA 4→4	0.2%(1)	0%
FAMACHA 4→5	0.2%(1)	0%
FAMACHA 5→1	0%	0%
FAMACHA 5→2	1%(4)	0%
FAMACHA 5→3	0.7%(3)	0%
FAMACHA 5→4	0%	0%
FAMACHA 5→5	0%	0%

A – Sheep



B – Goat

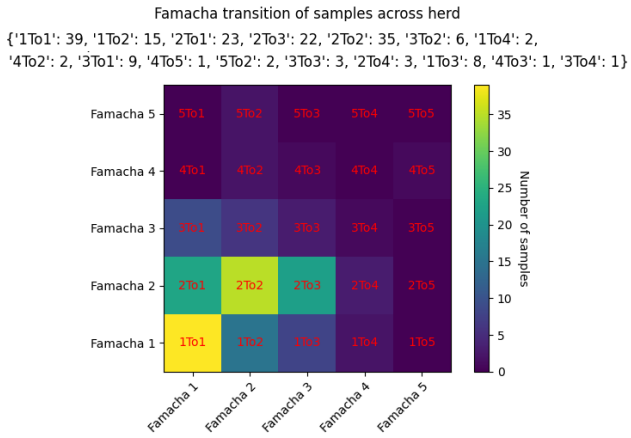


Figure 3.2: **FAMACHA transitions available.** Illustration of the amount of FAMACHA transitions in the dataset

3.2 Methods

We present our pipeline used for classifying the health status of the animals in Figure 3.3. The first step is to 'repair' the raw data. This step is necessary as the sensors used in this study have an irregular sample rate, i.e. the activity counts measured are not always associated with the appropriate time bin, for example, two different counts can be saved as marked as measured at the exact same time. This issue is believed to be caused by the communication protocol used by the transponders and the base station. The aim of the 'repair' step is to rearrange every measured count to its appropriate unique time bin. For this, the backfill approach described in chapter 2 is used (Figure 2.6). Imputation is the next step in the pipeline which allows us to use as much data as possible from the raw dataset. This outputs accelerometry imputed count data for multiple transponders which we subsequently use to build our dataset samples based on the evolution of the FAMACHA score on consecutive weeks (Figure 3.4). The figure also highlights the need for imputation as discussed in chapter 2 as we can clearly see multiple FAMACHA transition samples where activity data is missing.

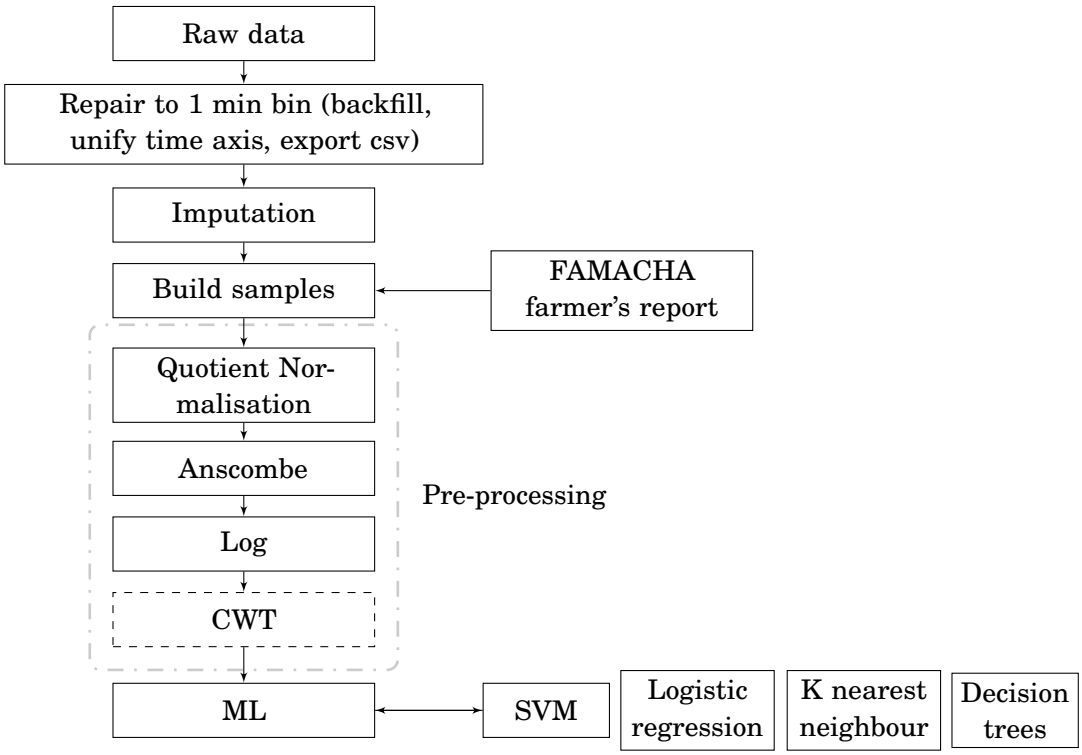
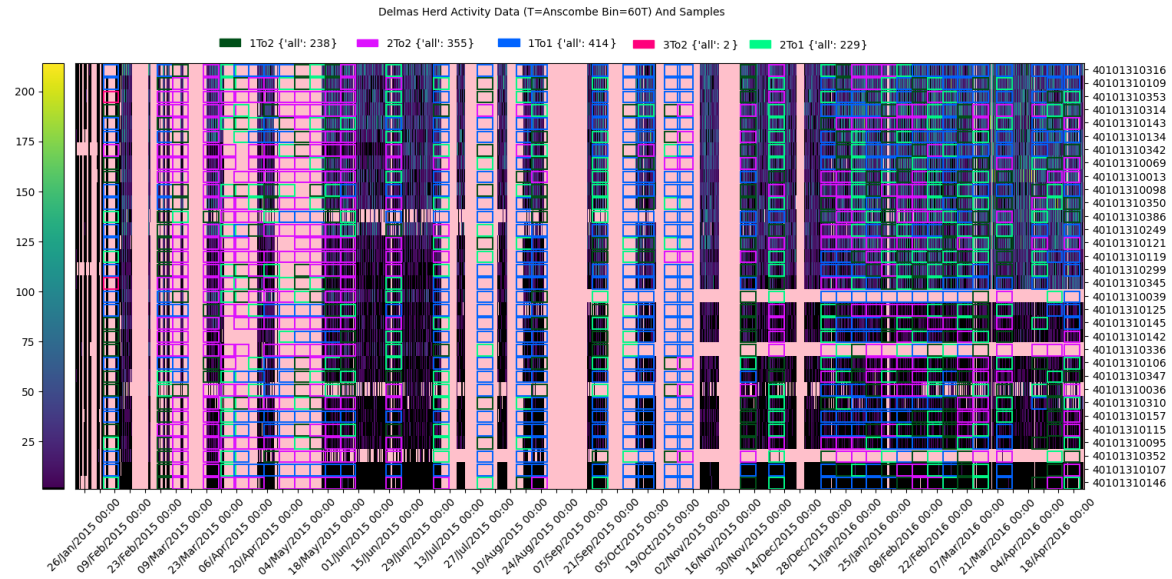


Figure 3.3: **Classification of health status pipeline**. Diagram showing the simplified pipeline used for the classifying of healthy from unhealthy samples.

After this stage, our samples each contain an array of activity points and a FAMACHA label. Samples are then pre-processed before training different ML classifiers through a cross-validation approach. We will feed our different models with the results directly and after Continuous Wavelet

Transform (CWT, see section 3.3) to bring out different frequencies of activity, evaluating and interpreting performance. We will also investigate the effect of the number of days used in the samples activity array, the amount of imputed data used, different pre-processing pipelines and classifiers.

A – Sheep



B – Goats

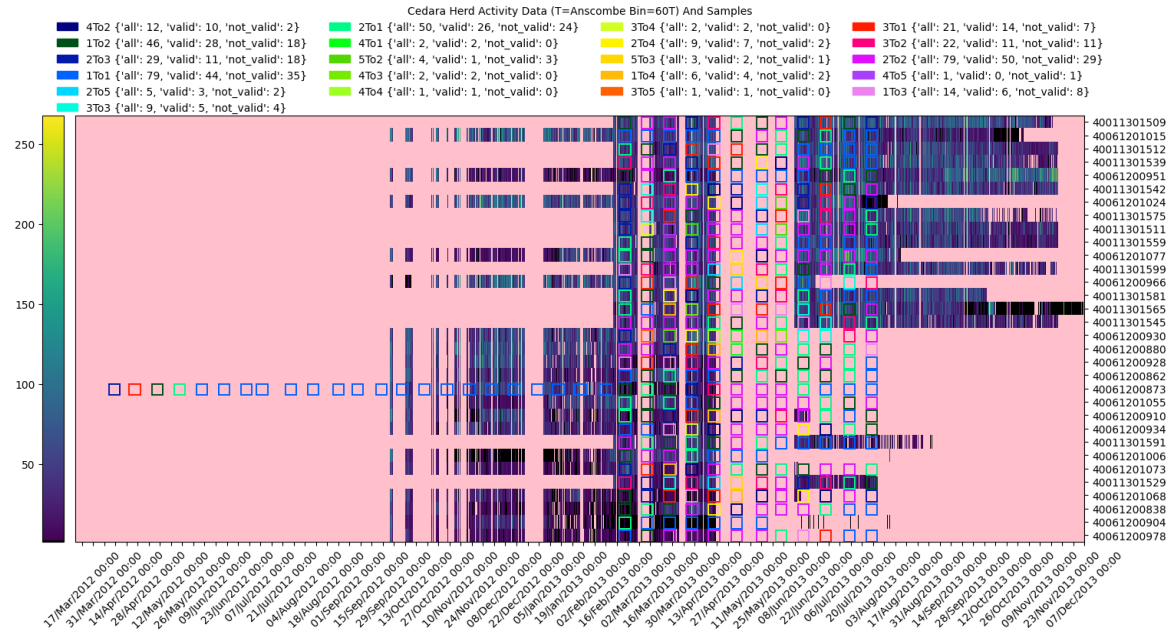


Figure 3.4: **Heatmaps of FAMACHA samples for ML**. For both heatmaps the x-axis shows the time from the start to the end of the study, the y-axis shows the different transponders in the herd after filtering out invalid ones (Section. 2.3.2.2). The activity count data displayed is transformed with the Log of the Anscombe (Sections. 2.2.2.2, 2.2.2.1). The data is re-sampled to an hour resolution (i.e. each bin contains the sum of 60 activity counts) for ease of visualisation of the entire time frame. While (A) show the sheep herd (B) shows the goat herd. The FAMACHA transitions are located and specified as per the legend with the overlay of coloured boxes. Note that in these heatmaps the activity data is not imputed, missing points are highlighted by the pink background.

3.2.1 Building samples

Because we use supervised ML, we need to build a dataset which contains sets of features and target/label or ground truth, i.e. the value we want to predict. In our case, the label is the change of FAMACHA score from two consecutive tests, the time distance between two tests being a week for the sheep farm Delmas and two weeks for the goat farm Cedara. The features in each sample are composed of the activity time series associated with a FAMACHA transition. The feature time series is selected based on the date of the first FAMACHA test, it contains the activity data measured before the test date at midnight, for example for a FAMACHA transition from 1 to 2 with the first FAMACHA score evaluated at 2 on Feb 12, a feature array contains 1 day of activity data will contain the data measured from Feb 11 at 00:00am to Feb 12 at 00:00pm (Fig 3.5). The date of the FAMACHA tests of all samples for the sheep herd and the goat herd can be visualised in Figure 3.4.

Class imbalance is a common problem in ML [58], where the distribution of instances across different classes is highly imbalanced, with one or more classes having significantly fewer instances than the others. This can lead to biased predictions, where the model tends to favour the majority class and performs poorly on the minority class. There are various techniques and algorithms that can be used to handle class imbalance, such as resampling, cost-sensitive learning, and ensemble methods [22]. For the FAMACHA transitions ≤ 2 our dataset did not have a significant class imbalance between the healthy and unhealthy (1To2 or 2To2) group as illustrated in figure 3.2 and Table 3.2. The biggest imbalance was in the case of the Sheep farm when comparing the healthy samples (1To1) with the unhealthy samples (1To2) where we had 156 and 98 possible examples respectively, in this case, the healthy group was approximately 1.6 times larger than the unhealthy. Slightly imbalanced datasets, such as in our use case, may not necessarily be a major concern for ML algorithms. In such cases, the class imbalance is relatively small, and many algorithms can still perform well without any specific handling of the imbalance [102].

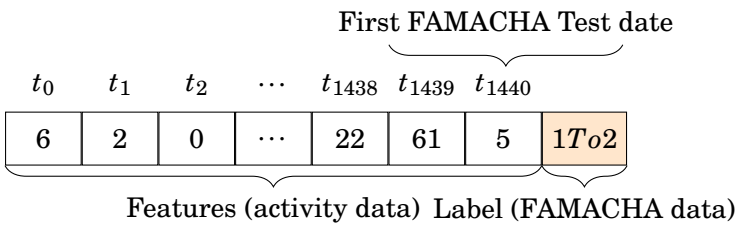


Figure 3.5: **Example of a single day-long ML sample**. Illustration of a ML sample composed of 1440 features (all activity counts) and a FAMACHA label indicating a transition of FAMACHA from 1 to 2. Note that we only use activity data sampled at the 1-minute resolution, in this example there is a day (1440 minutes) of activity data, 1 minute per feature.

As this research focuses on a binary classification approach, we decided to build 2 groups of

FAMACHA transitions, one being "healthy" and the other "unhealthy". Each group can contain single or multiple FAMACHA transitions. We tried groups of different compositions to find the optimal configuration for accurate classification. In section 3.1 we explained that FAMACHA scores ≥ 3 indicate poor health of the animal but our dataset is mostly composed of FAMACHA transitions containing score 1 and 2 (Figures. 3.2 and 3.1). We will define a sample as healthy if the FAMACHA score remained 1 for the two consecutive tests and we will refer to this transition with the notation FAMACHA $1 \rightarrow 1$ (1To1). Similarly we will refer to the unhealthy samples as FAMACHA $1 \rightarrow 2$ (1To2) or FAMACHA $2 \rightarrow 2$ (2To2). Although the amount of samples with other FAMACHA transitions is too low for training, we can test our models against those samples. Intuitively the unhealthy samples with FAMACHA $2 \rightarrow 2$ transition will be in a more "unhealthy" state that the FAMACHA $1 \rightarrow 2$ as the animal remained at a higher FAMACHA score (2) for longer, we will validate this hypothesis in the result section.

As part of routine husbandry, each animal which scored ≥ 2 during a FAMACHA evaluation was immediately treated with Levamisole (Ripercol-L, Bayer Animal Health) at $7.5 mg.kg^{-1}$ [9]. We created a dataset to examine our ability to predict which animals responded well to treatment, comparing FAMACHA $2 \rightarrow 1$ against FAMACHA $1 \rightarrow 1$. This was based on classifying the 5 days of accelerometer data immediately following treatment for the sheep. For the goat, FAMACHA $2 \rightarrow 1$ against FAMACHA $1 \rightarrow 1$ was compared and 3 days of accelerometer data were used.

	Cedara	**Delmas**
Healthy	1To1	1To1
Unhealthy	2To2, 1To2	2To2, 1To2

Table 3.2: **Definition of healthy/unhealthy status**

3.2.2 Exogenous factors

We have discussed how the parasitic burden of small ruminants is directly linked to the life cycle of $Helminths$ in section 1.3.1. We know that the worm population exponentially increase in wet weather, we can use the rainfall data on the location of each farm to asses how much predictive power it holds in comparison with the activity data, we will also combine the activity and rainfall data as well as other exogenous data sources such as the temperature and the wind speed. To include exogenous data in our samples we will append the data in the feature array (Fig 3.6). Similarly to the activity data (3.2.1), we select the data before the first FAMACHA test date up to the length of the sample. However, contrary to the activity data which is sampled per minute, the resolution (measurement per unit of time) of the weather data is hourly, it is not necessary to upscale the weather data to match the activity data resolution as this would only add redundant values to the feature array and the ML algorithms we used are not sensitive to the concept of different time resolutions in the feature space. We will only use Min Max scaling

(section 3.2.3) which preserves the relative differences in magnitude while bringing all values within a standardised interval. This approach preserve the original scale of the data while still providing a consistent range for machine learning algorithms that are sensitive to feature scales such as SVM.

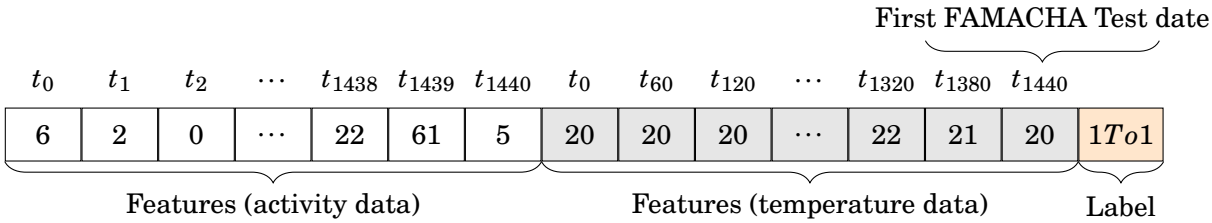


Figure 3.6: **Example of a single sample with exogenous data**. Illustration of a ML sample composed of activity and temperature data and a FAMACHA label indicating a transition of FAMACHA from 1 to 1. Note that the temperature data is sampled hourly while the activity data is sampled per minute.

3.2.3 Intensity normalisation

The magnitude or scale of the activity data greatly changes depending on the transponder used. The accelerometry sensor calibration might not be uniform across all of the transponders used. Furthermore, despite attaching the transponders with the same collar in the same position, it is challenging to regulate the tightness of the attachment. In addition, the attachment used does loosen up over time which will affect the recorded accelerometry values as the device will move with less inertia (Section 2.2). For these reasons intensity normalisation at the sample level is key for our ML approach as we aim to model the difference in patterns that might indicate ill-health within our data and avoid bias dues to the difference in intensities that we have no control over.

Feature scaling is an important step for most ML algorithms, some algorithm such as K-NN depends on computing distances between the data points while others rely on solving the gradient descent optimisation problem which is highly dependent on the scale of each feature. If a feature has an order of magnitude larger than others, it might have an overly high weight in the objective function of the algorithm and make it unable to learn from other features correctly. Features can have completely different magnitude ranges naturally, especially if they come from different sources, for example, weather data such as temperature and rainfall have different magnitude ranges. Feature normalisation aims to scale all the features to a comparable scale of the same range.

Some of the commonly used techniques for feature scaling include L2-normalisation, Min-Max normalisation and Standard Scaling. Min-Max normalisation scales the data array x between the range $[0, 1]$:

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{3.1}$$

While Standard Scaling subtracts the input array x by its mean value and divides it by its standard deviation σ, which centres the data around zero and scales it to unit variance ($\sigma = 1$).

$$x_{scaled} = \frac{x - \overline{x}}{\sigma} \tag{3.2}$$

L2 Normalisation simply divides the input sample array x by its Euclidean distance:

$$||x||_2 = \sqrt{\sum_{i=1}^{n} x_i^2} = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2} \tag{3.3}$$

These techniques are all linear transformations which do not change the probability distribution of the data, they only scale the magnitude of the data. Each of these techniques could be used for sample normalisation, however as all the data points are used to compute the scaling they can be biased by random bursts of high or low activity. Rather, we are interested in normalising each sample robustly so that only data points comparative to herd activity levels are taken into account. A viable solution is Quotient normalisation (QN), which scales the intensity of samples originating from each animal with a scaling factor that takes into consideration the herd behaviour.

3.2.3.1 Quotient normalisation

Quotient normalisation is a method often used for normalising gene expression [109], Dieterle et al successfully used Quotient Normalisation (QN) to normalise the concentration (intensities) of the spectra of complex biofluilds [34]. It is particulary useful when the variation of the data is not constant over time such as in our activity time series. Here we describe with examples the different steps of QN. The input must be activity counts stored as samples in a matrix M of size (nxm) such that each row contains a single activity sample for a total of n samples and each sample containing m activity counts.

$$M = \begin{bmatrix} s11 & \dots & s1m \\ \vdots & \ddots & \vdots \\ sn1 & \dots & snm \end{bmatrix}$$

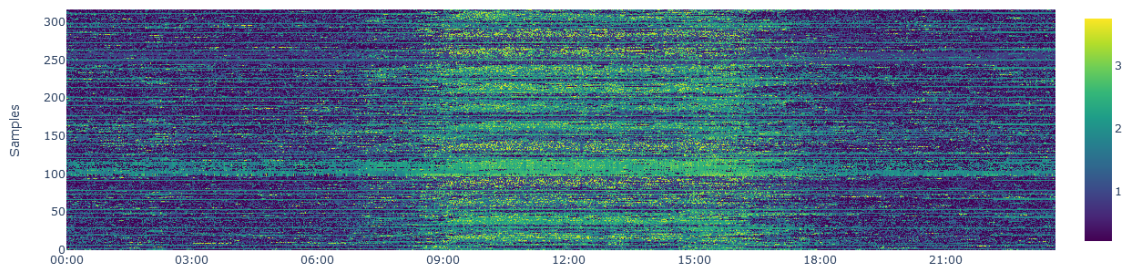


Figure 3.7: **Quotient Normalisation step 1**. Visualisation of the 1-day length activity samples after Anscombe transformation.

First, we define the herd-level activity across time as the median sample H of M. Here, the median value of each column of M is calculated and stored in a m sized array. Figure 3.7 shows all of the samples while 3.8 shows the median sample

$$H(m) = \left[median(\begin{bmatrix} s11 \\ \vdots \\ sn1 \end{bmatrix}), \ldots, median(\begin{bmatrix} s1m \\ \vdots \\ snm \end{bmatrix}) \right]$$

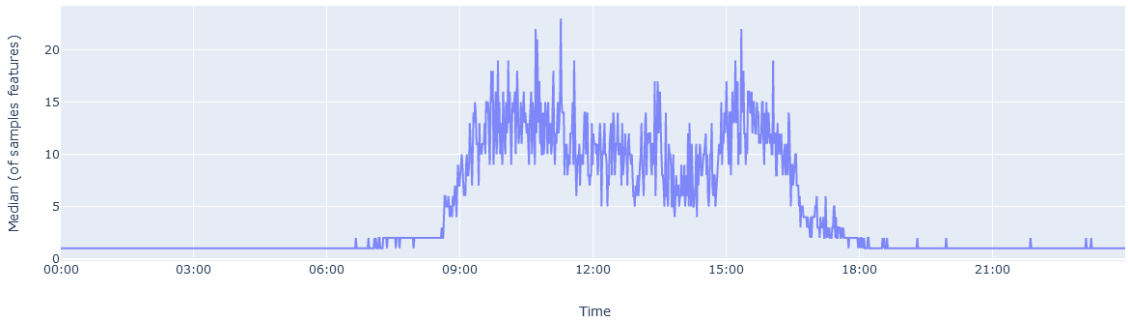


Figure 3.8: **Quotient Normalisation step 2**. The 'herd level activity' point-wise median sample [median of col1, ... median of col n]

each row in M is then divided by the "median sample" H element-wise to give D, the point-wise scaling matrix between the herd activity and each individual activity trace:

$$D = \begin{bmatrix} \frac{s11}{H(0)} & \cdots & \frac{s1m}{H(m)} \\ \vdots & \ddots & \vdots \\ \frac{sn1}{H(0)} & \cdots & \frac{snm}{H(m)} \end{bmatrix}$$

We then calculate the median of each row in D, which is our robust scaling factor for each sample:

$$A(n) = \begin{bmatrix} median(\begin{bmatrix} \frac{s11}{H(0)} & \cdots & \frac{s1m}{H(m)} \end{bmatrix}) \\ \vdots \\ median(\begin{bmatrix} \frac{sn1}{H(0)} & \cdots & \frac{snm}{H(m)} \end{bmatrix}) \end{bmatrix}$$

Now that we have calculated our scaling factors, we can just simply scale the input matrix M by dividing each row of M by the corresponding scaling value $A(n)$ obtained in step3. Figure 4.10 shows the fully normalised samples.

$$Norm = \begin{bmatrix} \frac{s11}{A[1]} & \cdots & \frac{s1m}{A[1]} \\ \vdots & \ddots & \vdots \\ \frac{sn1}{A[n]} & \cdots & \frac{snm}{A[n]} \end{bmatrix}$$

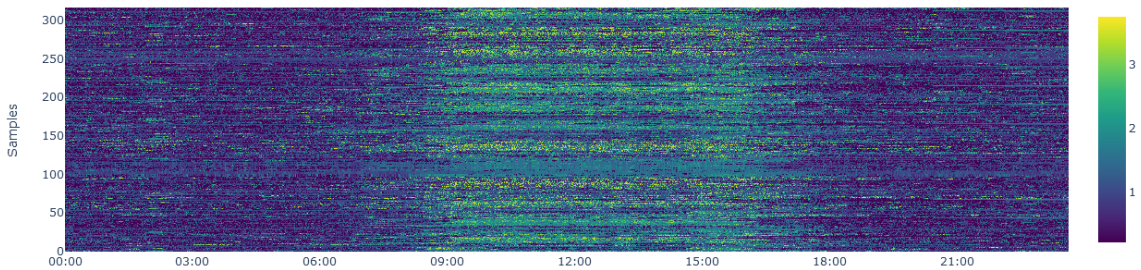


Figure 3.9: **Final results of Quotient Normalisation.** Visualisation of the activity samples (1 day length) after QN.

3.2.4 Machine learning

Time series classification is a common problem which has been tackled with various ML algorithms. We will classify our datasets with some of the most commonly used algorithms and compare their results, namely K-Nearest Neighbour (KNN) [157] [47], Support Vector Machine (SVM) [71], Logistic Regression (LREG) [2], and Decision tree (DTREE) [70]. SVM's are described in Chapter 2 section 1.7.1.

The K-Nearest Neighbour algorithm is one of the simplest supervised ML algorithm used for classification, it is based on feature similarity, aiming to group samples by comparing their relative distance in the feature space. The k in KNN is a parameter that refers to the number of nearest neighbours (samples) to include in a majority voting process, a new sample is classified by majority votes from its k nearest neighbours. In KNN choosing the optimal k value is the objective, the process of finding its optimal value is called parameter tuning (Fig. 3.10).

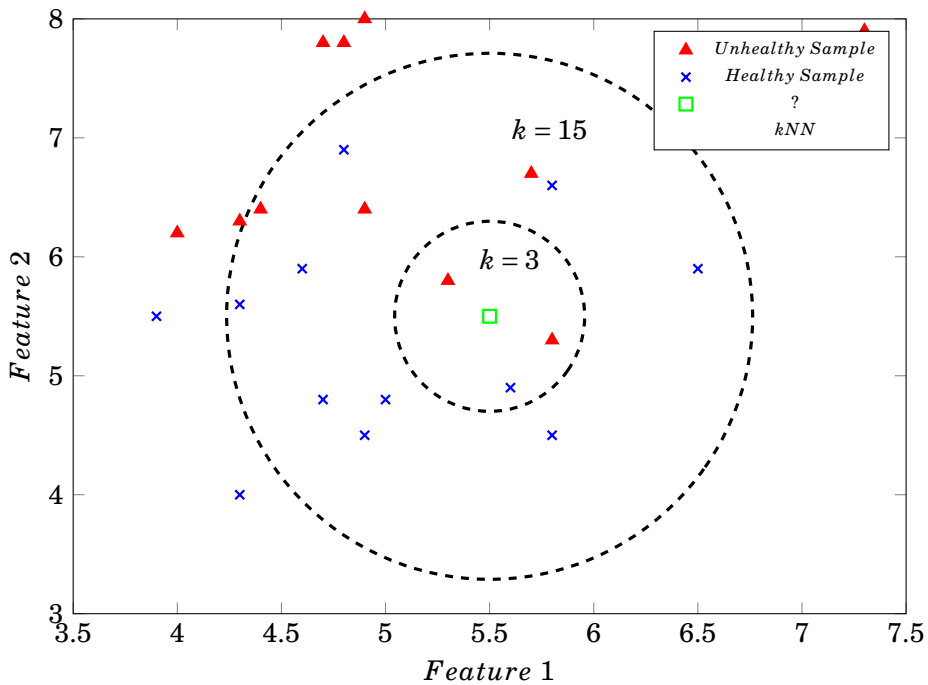


Figure 3.10: **K-Nearest Neighbour classification example**. Illustration of the KNN classification algorithm majority voting process, in this example we reduced the number of features in our samples to 2 for ease of visualisation. To classify a new unknown sample (represented by the green square), KNN aim to find the optimal number of nearest neighbour k for correct classification. In this example, if $k = 3$ the new sample will be classified as unhealthy while if $k = 15$ it will be healthy.

Because KNN does not learn a discriminative function from the training set, it is more appropriate for smaller datasets as computation expense is poor compared to other techniques on larger datasets. To find the nearest neighbours, KNN can use the Euclidean distance defined by

the Equation. 3.4 where d is the distance between two points of coordinate (x_1, y_1) and (x_2, y_2).

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{3.4}$$

By calculating the distance of an unknown data point from all the points in the dataset we can rank the k nearest neighbours. Note that for this example we used 2D points (with 2 features) for simplification. Our samples are of much higher dimensions that cannot be visualised on a 2D graph like in the example we presented in figure 3.10.

Regression analyses are methods for modelling relationships between variables, they allow the inference or prediction of a variable based on other variables. In a Logistic Regression (LREG), the variable we wish to predict is dichotomous i.e. can only have two values (for example healthy and unhealthy). It is called the dependent variable and in our case is the label/target of the sample we wish to classify. The goal of LREG is to estimate the probability of the occurrence of the prediction, the value range of the output of LREG is therefore between 0 and 1. The objective function of LREG is called the logistic function (Fig 3.11) and is defined by the Equation 3.5 where z is the equation of the linear regression defined as Eq 3.6 for k features x_k and b_k and a regression coefficients.

$$P(y = 1|x_1, \ldots, x_k) = \frac{1}{1 + e^{-z}} = \frac{1}{1 + e^{-(b_1 * x_1 + b_2 * x_2 + \cdots + b_k * usethe)}} \tag{3.5}$$

$$y = b_1 * x_1 + b_2 * x_2 + \cdots + b_k * x_k + a \tag{3.6}$$

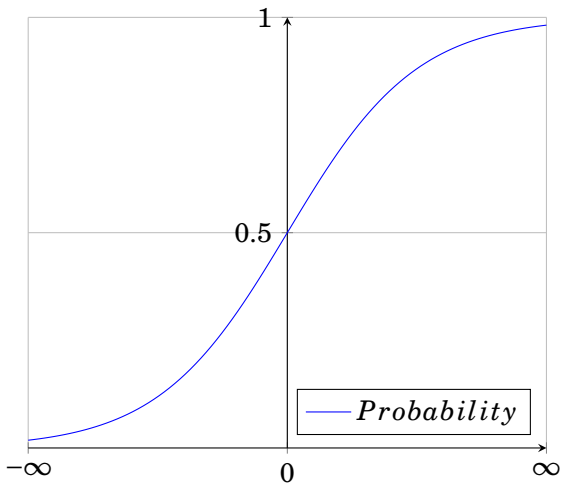


Figure 3.11: **Logistic function.** The x-axis is define between $-\infty$ and $+\infty$ while the y-axis is define between 0 and 1.

The probability that the dependent variable is 1 is given by Equation 3.5 for x_k features. For example, the probability of a 1-day activity sample 3.5 to be unhealthy is a function of the activity feature array defined by Eq 3.7:

$$P(is\ unhealthy) = \frac{1}{1 + exp^{-(b_1 * feature_1 + b_2 * feature_2 + \cdots + b_{1339} * feature_{1440} + a)}} \tag{3.7}$$

To find the coefficients b_k and a that best model the given data, the maximum likelihood optimisation method [124] is used.

Decision trees (DTREES) are binary trees that recursively splits a data set until their "leaf" nodes only contain samples of the same class. A DTREE is composed of two different types of nodes, decision and leaf nodes, the former applies a condition to split the data while the latter decides the class of a sample (Fig 3.12). The first node is called the root node where all of the data set samples will be initially split into child nodes. The optimal splits are decided by calculating the "Information Gain" from the previous or root node with the entropy (equation 3.8) difference between the child nodes' entropy and the previous node's entropy for every possible split and picking the split that maximises the Information Gain. The Information Gain is defined by the equation 3.9 where w is the relative size of the child node with respect to its parent node.

$$\begin{aligned} E &= \sum -p_i * log(p_i) \\ p_i &= probability\ of\ class\ i \end{aligned} \tag{3.8}$$

$$Information\ Gain = E(parent\ node) - \sum w_i * E(child_i) \tag{3.9}$$

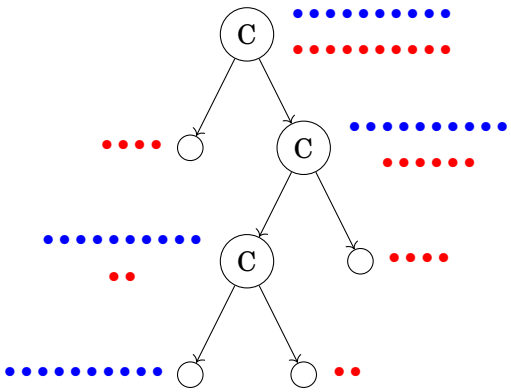


Figure 3.12: **Schematic of a trained Decision Tree.** In this example we used a 20 samples dataset with 10 healthy samples and 10 unhealthy samples, the healthy samples are represented by blue points outside the tree nodes while the red points show the unhealthy samples. The DTREE algorithm aim to find the best possible split of the samples in each node until the nodes contain only 1 type of sample based on a condition C.

3.2.5 Regularisation

We will use the Python Scikit-learn's (section A.1) implementation of the different ML classifiers we used [111] which offer numerous regularisation strategies. Regularisation is a technique used in ML to prevent overfitting of the model to the training data, which can result in poor generalisation to new, unseen data. In Scikit-learn, several popular classifiers implement regularisation techniques to improve model performance. Here's a summary of how regularisation works in Scikit-learn for SVM, DTREE, LREG, and KNN classifiers:

- SVM:
  Scikit-learn's SVM implementation includes the regularisation parameter "C," which controls the trade-off between achieving a low training error and a low testing error. The larger the value of C, the more emphasis is placed on correctly classifying the training examples, while smaller values of C allow the model to be more flexible and potentially better able to generalise to new data. For the RBF kernel, the "gamma" parameter is used which determines the width of the kernel function (Equation 1.20). A small gamma results in a wider kernel, which means that the DB is smoother, while a large gamma results in a narrower kernel, which means that the DB is more complex and can better fit the training data. We used the default value of $\gamma = \frac{1}{n\ features * variance\ of\ samples}$ (figure. 3.13). A grid search over a range of values for gamma and C is also a good option to find optimal parameters (Fig. A.1).

  SVM also uses L1 and L2 regularisation methods to penalise large weights for its linear version, we used the default L2 regularisation for our classifiers which works by adding a penalty term to the SVM objective function. The penalty term is proportional to the square of the L2 norm of the model coefficients, also known as the weight vector. The L2 norm is the square root of the sum of the squared values of the coefficients.

- Decision Tree:
  Scikit-learn's decision tree classifier includes regularisation techniques like pruning, which reduces the size of the tree by removing unnecessary nodes. Pruning can help prevent overfitting by simplifying the tree and reducing the risk of capturing noise in the training data. Scikit-learn's decision tree classifier also includes parameters like "$min\ samples\ split$" and "$max\ depth$" that can be used to control the complexity of the tree.

- Logistic Regression:
  Scikit-learn's logistic regression classifier includes regularisation techniques like L1 and L2 regularisation, which add a penalty term to the objective function to discourage the model from overfitting. The strength of regularisation can be controlled using the "C" parameter, similar to SVM. We will use the default L2 regularisation method.

- KNN:
  Scikit-learn's KNN classifier includes regularisation techniques like distance weighting, which weights the contribution of each neighbour to the classification decision based on their distance from the query point. Distance weighting can help prevent overfitting by reducing the influence of noisy or irrelevant neighbours in the classification decision we also used the commonly used k as the square root of the total number of samples as tuning k is a challenging and computationally expensive task in itself [57].

In summary, regularisation is a powerful technique for preventing overfitting in ML models. Scikit-learn provides several popular classifiers with built-in regularisation techniques, including SVM, DTREE, LREG, and KNN classifiers. By controlling the strength of regularisation and other hyperparameters, these classifiers can be optimised for better performance on unseen data. Gkerekos et al showed that default hyperparameters included in Scikit-learn are reasonably effective [49]. Kim et al used Scikit-learn's default regularisation parameters for their DTREE model with good results [77]. Wainer et al. concluded in their research paper on SVM regularization hyperparameters that it is likely not worth investing excessive computational time in the search for better hyperparameters. [150].

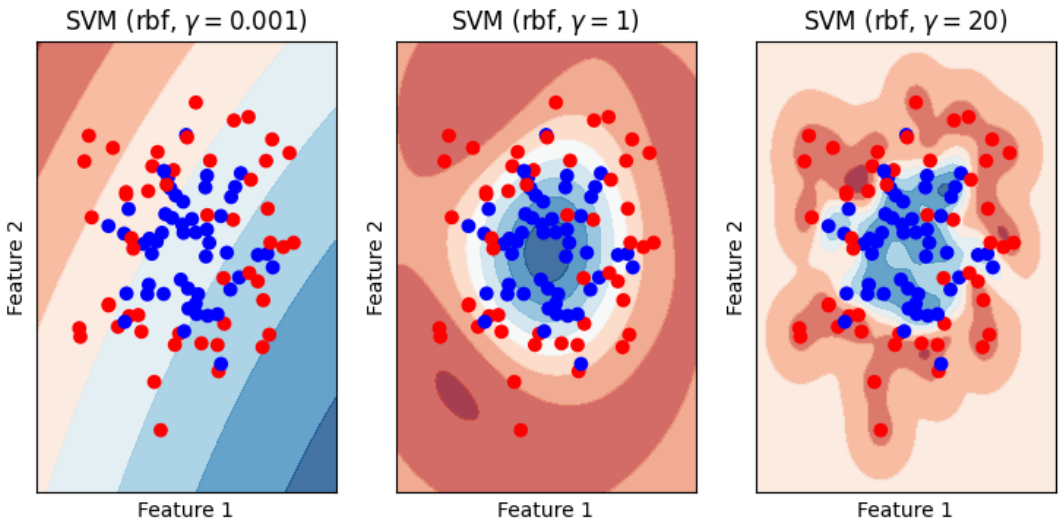


Figure 3.13: **Illustration of SVM model regularisation.** In this illustration we show the impact of the regularisation parameter of an SVM model with RBF kernel. In this example, the data set is composed of two classes, blue and red. The contours show the DB for our trained SVM model. On the left, we can see that this model decision function is not complex enough to model the data, this model is underfitting while on the right the DB is overly complex and tailored to the data including the noise of the problem, this is called overfitting. Regularisation aims to find a balanced DB like the one in the middle.

3.2.6 Evaluation

We will evaluate how well our models can classify our data via multiple experiments:

1. Classification of health status.
2. Classification of response to treatment.
3. Temporal validation.
4. Cross-farm validation.
5. Early prediction.
6. Model explainability.

In the 'Classification of health status' experiment we train our models with "healthy" and "unhealthy" samples and evaluate how well unseen activity data is classified. For the 'classification of response to treatment' we aim to evaluate the health evolution of the animal after being given treatment by training our models with examples of where the FAMACHA score did or did not successfully decrease after treatment, this way we can check treatment effectiveness.

We also explored the impact of concept drift which occurs when the data used for training a model changes over time in unforeseen ways that impacts the accuracy of the model (Fig. 3.14).

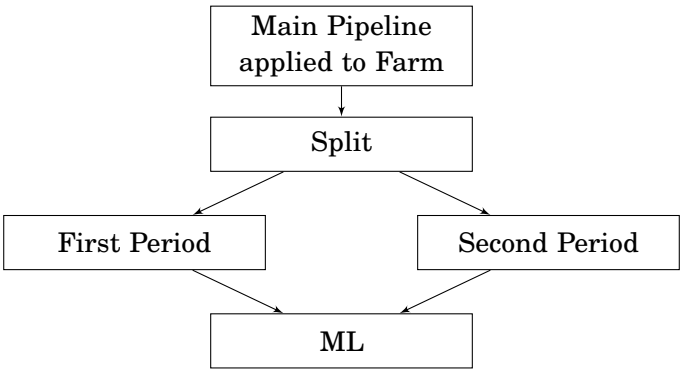


Figure 3.14: **Temporal Validation**. Schematic of the temporal validation

The notion of "concept drift" describes the decrease in performance of a given classifier due to changing environmental or sensing conditions over a period of time. For example, training data collected at the start of a given period becomes less representative of future data. This is a common issue in long-duration supervised classification problems that use real life data which is, in most scenarios, changing intrinsically [148].

With the aim to show how well a classifier trained on a farm performs on the data of another farm and other small ruminant species, the following experiment was devised. We trained

classifier with the data from the Delmas sheep farm and tested it on the data of the Cedara goat farm and vise versa (Fig. 3.15).

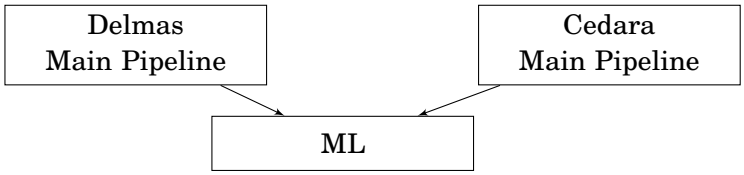


Figure 3.15: **Schematic of Cross-farm Validation** .

We also study how the predictive power of our model behaves as we move back in time, i.e. an earlier prediction of ill health.

Finally, we also looking at explaining the result of the trained models to understand what part of the activity data holds the most discriminative power for the classification of health status. We trained our models to classify healthy (FAMACHA 1To1) samples from unhealthy samples (FAMACHA 1To2 or FAMACHA 2To2) because of the relatively high number of samples available for these FAMACHA transitions (Fig 3.2). Other transitions can be used for testing/validation of our train model on unseen samples. We assume that a more advanced > 2 FAMACHA score in the second test of the transition should be classified as unhealthy while a transition ending with a FAMACHA score of 1 should be classified as healthy.

The rest of this section explains the methodology we use to evaluate performance of the above.

3.2.6.1 Cross-validation

Cross-validation is a widely used approach which allows us to split our samples into several 'folds', sequentially leaving out one fold to test the results of the mode trained on the rest of the data, as shown in Figure 3.16. Repeated nested k-fold cross-validation was used to optimise the hyperparameters and evaluate the model. Kim *et al.* [76] showed that the repeated cross-validation estimator outperforms the non-repeated version by reducing the variability of the estimator and providing lower bias. Hence we choose to use 10-times repeated 5-fold cross-validation to assess a realistic estimate of the performance of the model predictions while making the most of the data set available.

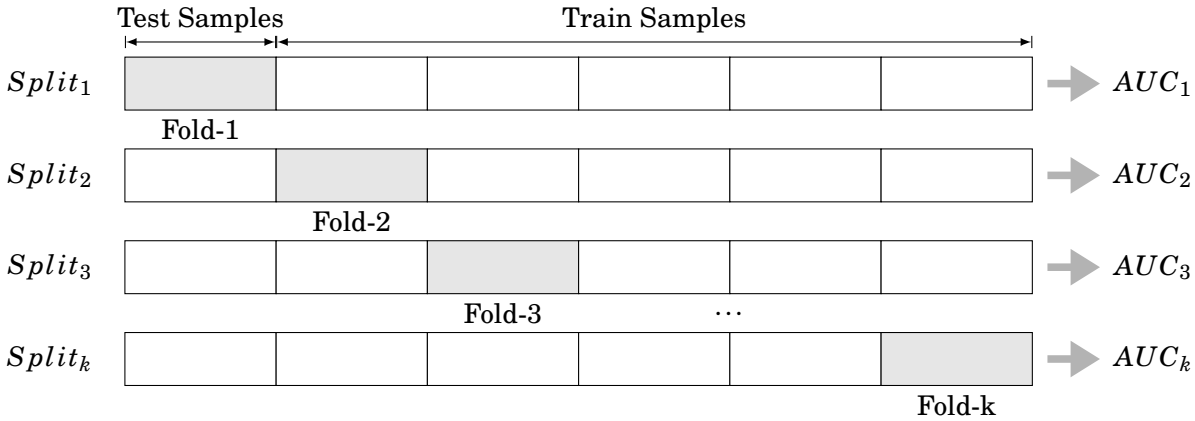


Figure 3.16: **Schematic of K-Fold cross-validation.** By splitting the ML samples we can train and test our model with different subsets of the entire data set. In this illustration, the test samples are represented in grey while the training samples are represented in white.

3.2.6.2 Area under the curve, Receiver operator characteristic curve and Precision

To explain Receiver Operator Characteristic curves (ROC) and the area under the curve (AUC) we must first introduce the concept of a confusion matrix. The confusion matrix summarises how a model performs on the test data, the rows in a confusion matrix correspond to what the ML algorithm predicted (for example, healthy and unhealthy) and the columns correspond to the known truth (Table. 3.3) The true positives (TP) are defined as samples that was unhealthy that were correctly identified by the prediction, the true negatives (TN) are the samples that were healthy that were correctly classified, while the false negatives (FN) are when the sample is unhealthy but the model predicted it was healthy, and false positives (FP) are samples that were healthy but the model predict them as unhealthy.

ML algorithms output score between -infinity and infinity of a given sample to be healthy or unhealthy, although many are able to convert this to a probability between 0 and 1. To make a

		Actual	
		Unhealthy	Healthy
Predicted	Unhealthy	**True Positives**	False Positives
	Healthy	False Negatives	**True Negatives**

Table 3.3: **Example of a confusion matrix**. This example is base on our use case for the binary classification of healthy and unhealthy samples.

decision a threshold must be set, for example if we pick a 0.5 threshold all samples returning a probability above 0.5 will be categorised as unhealthy while those below 0.5 will be classified as healthy. By changing the value of the threshold for deciding if a sample is unhealthy or not we change the calibration of the classifier and hence the values in the confusion matrix. For example, if we want to prioritise the detection of unhealthy samples we can lower the threshold to 0.1 which will increase the number of TPs but also increase the number of FPs and also reduce the number of FNs. Similarly we could set the threshold to 0.9 would decrease the number of FPs and increase the detection of TNs. By changing the threshold value we can find the optimal threshold value for our model for a particular task, ROC curves (Fig.3.17) provide a simple way to summarise the effect of different threshold values. The y-axis shows the True Positive Rate (TPR), also called the sensitivity:

$$TruePositiveRate = Sensitivity = \frac{TruePositives}{TruePositives + FalseNegatives} \tag{3.10}$$

In our case the True Positive rates tells us what proportion of the unhealthy samples were correctly classified. The x-axis shows the False Positive Rate (FPR), which is also $1 - Specificity$,

$$FalsePositiveRate = 1 - Sensitivity = \frac{FalsePositives}{FalsePositives + TrueNegatives} \tag{3.11}$$

The False positive rate tells us the proportion of healthy samples that were incorrectly classified.

We can calculate the TPR and the FPR for all of the confusion matrices generated by changing the value of the threshold from 0 to 1 and place the points on a ROC curve. A point at the top right corner ($TPR = 1$ and $FPR = 1$) means that even though we correctly classified all of the unhealthy samples, we incorrectly classified all of the healthy samples, while a point at the top left corner means that we correctly classified all unhealthy samples without mistake, the model makes "perfect" classifications. The line that goes through the points (0,0) and (1,1) is the "Chance line" (Gray line in Fig 3.17) where $TPR = FPR$ any points on this line means that the proportion of correctly classified unhealthy samples is the same as the proportion of incorrectly classified healthy samples. The ROC curve summarises all of the confusion matrices that each threshold produced.

The Area Under the Curve (AUC) is a value between 0 and 100 (in percent) is a metric that allow a simple comparison of different model performances, with higher AUC indicating better performances.

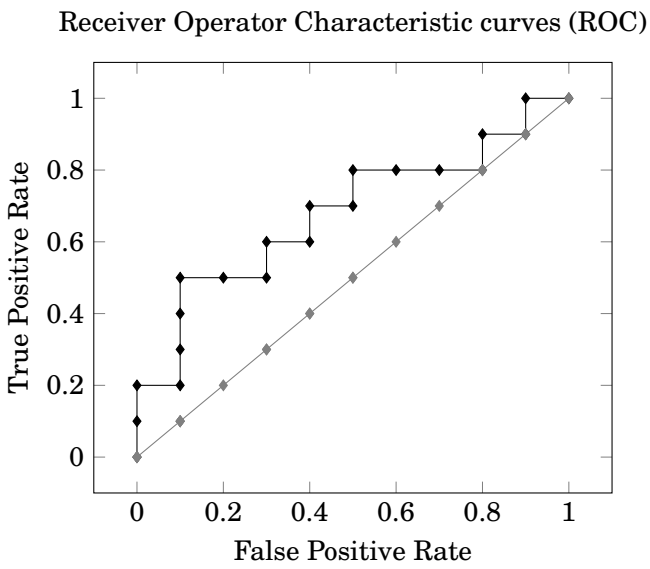


Figure 3.17: **ROC curve example**. Receiver Operator Characteristic (ROC) curves illustrating the performance of the classification model (black) compared to chance (gray), showcasing the trade-off between true positive and false positive rates.

Precision is another metric that gives insight into a model performance and is also based on the confusion matrices. It is defined as:

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives} \tag{3.12}$$

It is the proportion of positive results that were correctly classified. In the case that there are lots of samples that are healthy relative to the number of unhealthy samples, precision is more useful than the FPR because it does not include the number of TN in its calculation which makes it unaffected by class imbalances (large number of one category of samples compared to the other category), which is common in real world datasets.

3.2.6.3 Decision boundaries

The DB of a trained ML model is the functional boundary between the two possible predictions. Decision boundaries can give us insight into how a given model classifies samples. In this section we will focus on the DB of linear SVM models, in this case, the DB is a hyperplane as our samples exist in a high dimensional space of dimension equal to the number of features in the samples (for example 1-day samples of activity data have a dimension equal to 1440). We can use dimensionality reduction techniques such as Partial Least Square (PLS) [43] to visualise our samples and the DB hyperplane in a 2D space (Fig 3.18).

A – Sheep farm

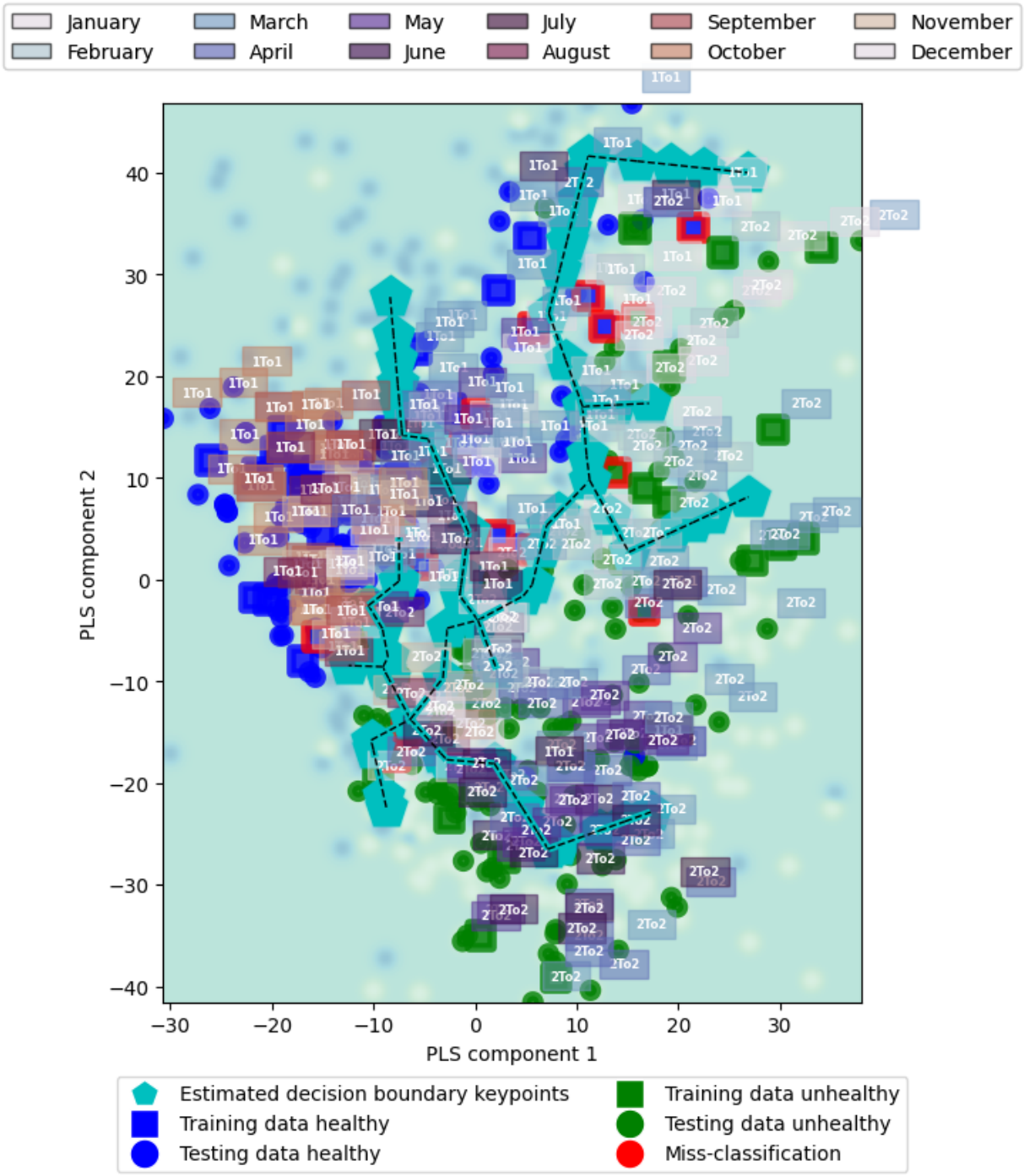


Figure 3.18

B – Goat farm

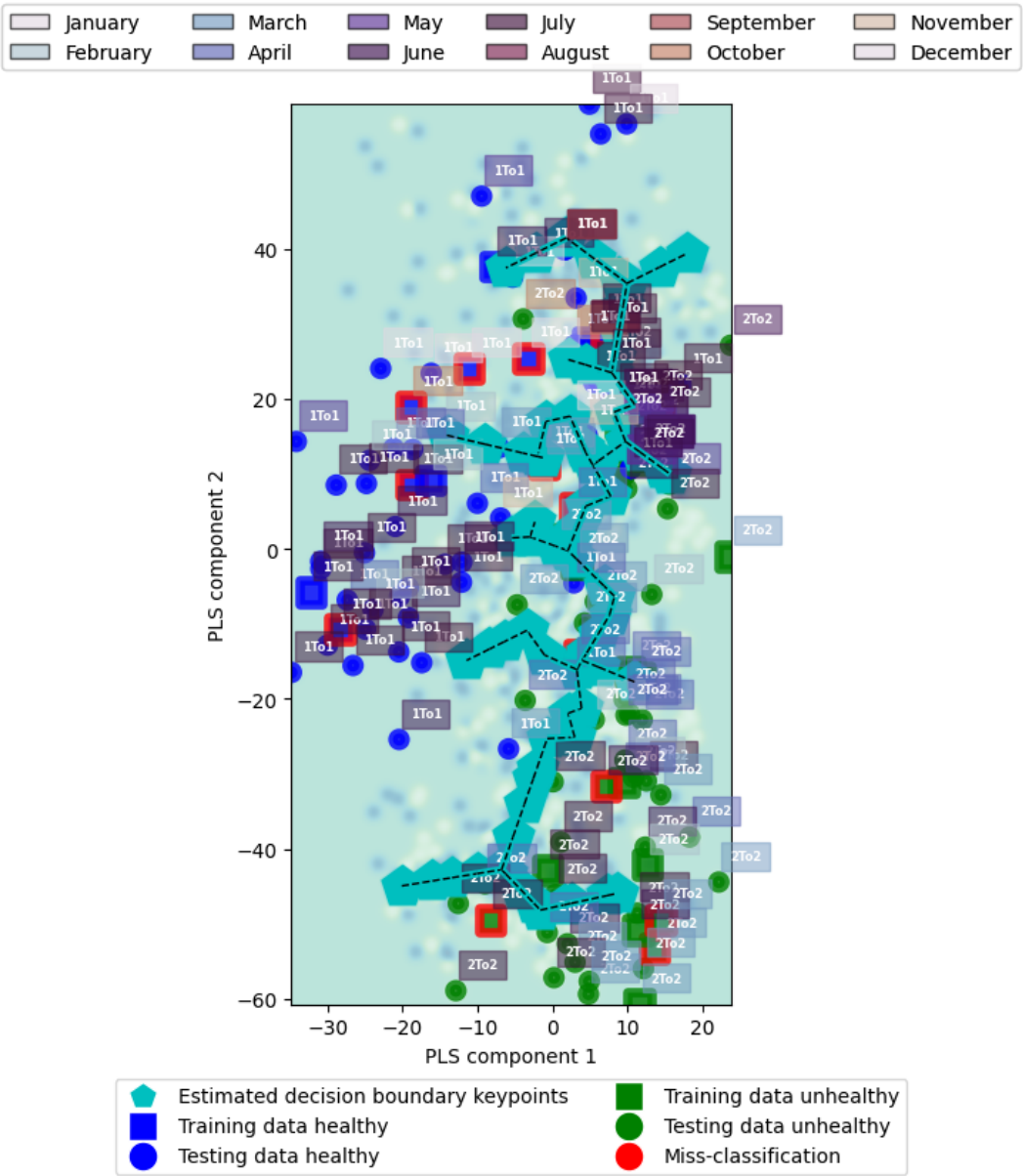


Figure 3.18: **Visualisation of SVM DB of sheep (A) and goat (B) samples.** Projection of our samples and the DB of a SVM classifier into a 2D space via dimensionality reduction with PLS. The black dotted line and the turquoise hexagone show the DB, while the green and blue circles show the testing samples, similarly, the blue and green squares show the training samples, and a red circle or square indicates a misclassification of the sample. This illustration also shows as an overlay the FAMACHA transition of each sample. The month of the data in the samples is also colour coded with a rectangular patch as per the legend.

3.2.6.4 Sample size and imputation/synthetic data

In Chapter 3 we discussed how we imputed the original activity data, the imputation results allow us to increase the number of ML samples (Fig. 3.5) we can create. Thanks to our approach we can create fully synthetic days of activity data if we chose, i.e. none of the data is from a true measurement but created by our deep learning imputation models which were trained on multiple examples of days. We can also simply fill in the gaps when the samples contain a smaller amount of missing points. We will use the number of synthetic days within a sample as a threshold to evaluate the impact of increasing the number of samples for training our ML algorithms while for our testing sets we will not use sample which contain uniquely synthetic data points. There are only a few days with no missing points which drastically reduces the number of samples we could build if we discarded samples with any level of missingness. Thanks to this approach we can use all of the real data available for classification, and potentially also high-quality synthetic data as a data augmentation approach. Figure 3.19 shows a schematic of a 7-day ML sample that contains a single synthetic day and a mixture of days with only real data and a mix of missing and real data.

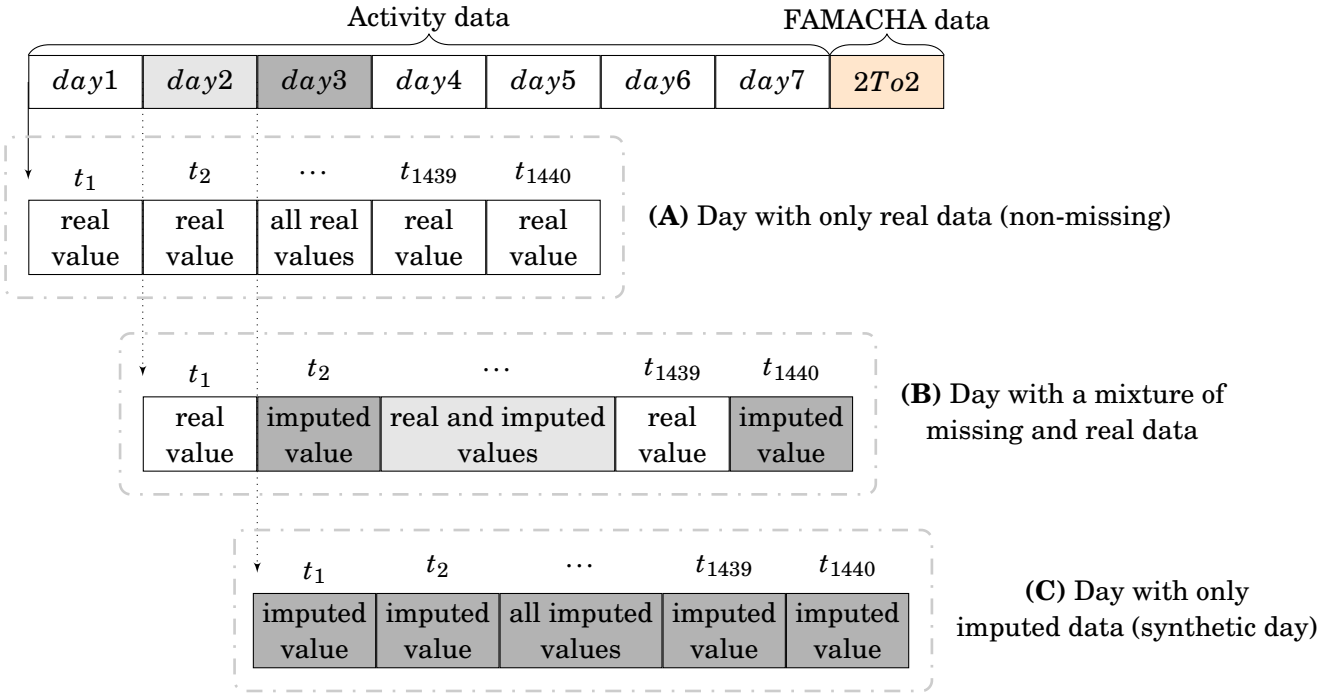


Figure 3.19: **Schematic of a 7-day-long ML sample with synthetic data**. A day of activity can more or less imputed points, in this example day 3 (A) is fully synthetic while day 1 (B) contains only real data and day 2 (C) has real and missing data. The samples in our datasets fit of these categories with the samples of type (B) being the most common

3.3 Results

3.3.1 Imputation

In chapter 2, we showed that the M-RNN imputation technique yields lower RMSE performances compared to the other method by evaluating the RMSE of the different transponder traces. In figure 3.20 we compare the RMSE of the ML samples (Fig. 3.5) we created for our ML pipeline.

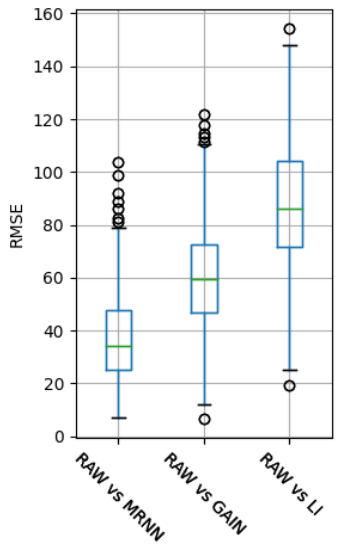


Figure 3.20: **Comparison of imputation performance of ML samples.** This box plot highlights the RMSE performance of the same ML samples for each imputation technique, we compared the different imputed versions of the samples with their real data counterpart.

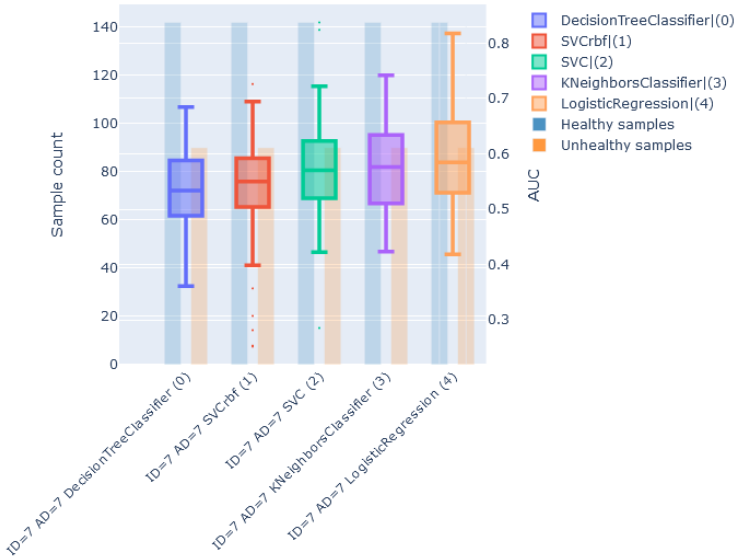
3.3.2 Classification of health status

For both farms, we will compare the results obtained by training different types of models, namely SVM (linear and rbf kernel), LREG, DTREE and KNN. We will also use different training labels for the health status of the samples, FAMACHA score of $1 \rightarrow 1$ will be the healthy class while FAMACHA score of $2 \rightarrow 2$ or FAMACHA score of $1 \rightarrow 2$ will be unhealthy. The number of activity days in a sample will be changed to 1, 4 or 7. Similarly, we will allow 1, 4 or 7 fully synthetic activity days in a sample (Section. 3.2.6.4). Furthermore, we will apply different pre-processing approaches to the activity time series of the samples. Initially we will show the top and bottom test results based on the AUC performance of the different models in tables 3.4 and 3.5.

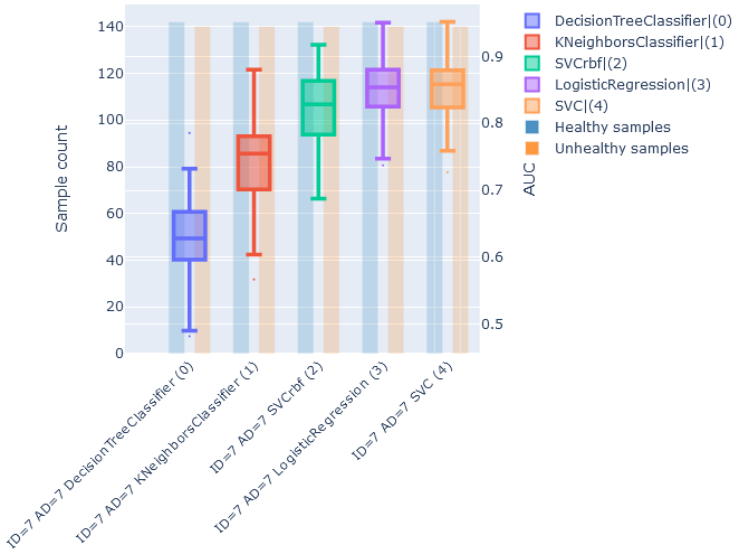
In figures 3.23 and 3.24 we show that the predictive power of our model is higher when we train with unhealthy samples with FAMACHA score of $2 \rightarrow 2$ compared to $1 \rightarrow 2$, this is likely due to the animal being in an unhealthy state (FAMACHA 2) for longer. Among the ML algorithms we tried SVM and LREG performed best with similar results while KNN and DTREE had lower performance. (Fig 3.21). In addition, although more computationally expensive we found

that using the frequency domain CWT transform did not improve classification performance (Table 3.6).

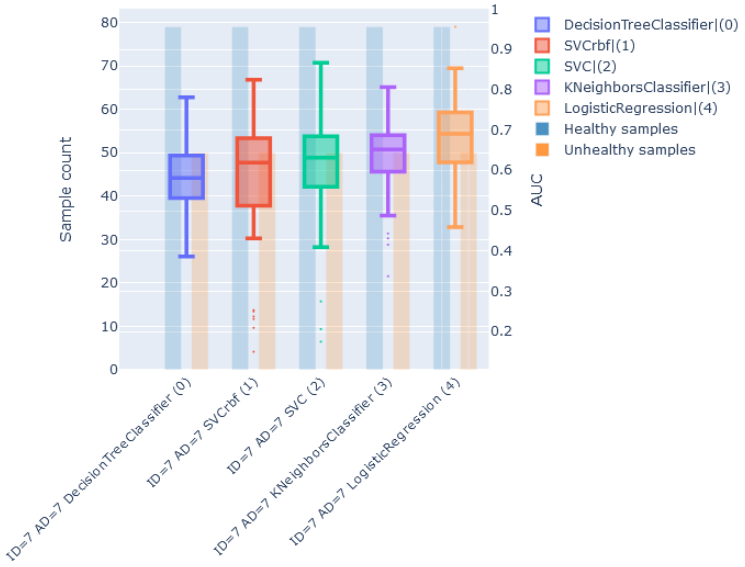
A – Sheep farm | Models trained with **unhealthy label 1To2**.



B – Sheep farm | Models trained with **unhealthy label 2To2**.



C – Goat farm | Models trained with **unhealthy label 1To2**.



D – Goat farm | Models trained with **unhealthy label 2To2**.

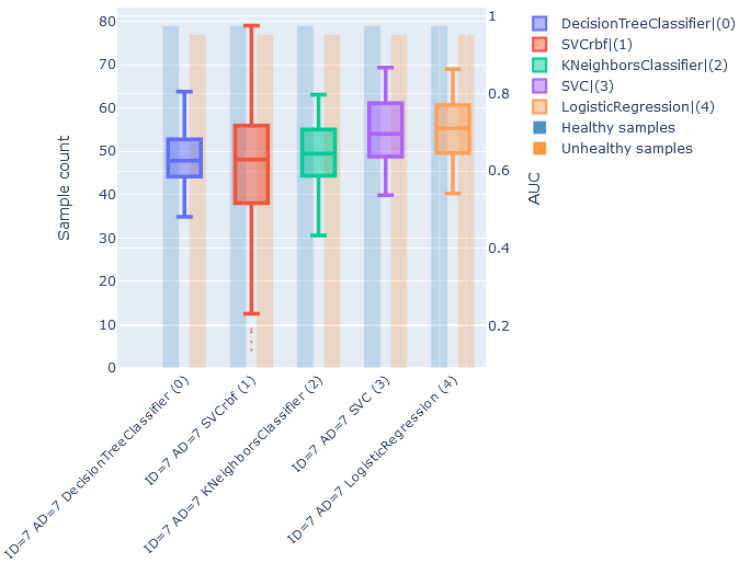


Figure 3.21: **Performance of different ML algorithms with M-RNN imputed samples for different unhealthy labels.** By fixing the number of activity days and the maximum number of synthetic days allowed in the samples to 7, we can compare the performances of different type of models. We also fixed the pre-processing pipeline of the activity time series to QN→ANSCOMBE→LOG→STDS. For each figure, the x-axis shows the ML algorithm while the left y-axis shows the number of samples per class and the right y-axis shows the AUC.

AUC test(95% CI)	Class0 P-test	Class1 P-test	N test	A-days	S-days	Imp	Clf	Pre-proc	Class1	Farm
0.80 (0.68-0.91)	0.70 (0.53-0.92)	0.72 (0.61-0.85)	42	7	4	M-RNN	LREG	QN->ANSCOMBE->LOG	2To2	delmas
0.80 (0.67-0.91)	0.70 (0.54-0.85)	0.73 (0.62-0.89)	42	7	4	M-RNN	SVM(linear)	QN->ANSCOMBE->LOG	2To2	delmas
0.80 (0.67-0.90)	0.70 (0.53-0.85)	0.73 (0.62-0.86)	42	7	4	M-RNN	SVM(linear)	QN->ANSCOMBE->LOG->MINMAX	2To2	delmas
0.80 (0.67-0.90)	0.70 (0.54-0.90)	0.72 (0.60-0.85)	42	7	4	M-RNN	LREG	QN->ANSCOMBE->LOG->MINMAX	2To2	delmas
0.80 (0.68-0.90)	0.71 (0.53-0.85)	0.74 (0.63-0.89)	42	7	4	M-RNN	SVM(linear)	QN->ANSCOMBE->LOG->STD	2To2	delmas
0.80 (0.68-0.90)	0.71 (0.53-0.85)	0.74 (0.63-0.89)	42	7	4	M-RNN	SVM(linear)		2To2	delmas
0.80 (0.67-0.90)	0.70 (0.53-0.84)	0.73 (0.63-0.86)	42	7	4	M-RNN	LREG		2To2	delmas
0.80 (0.67-0.90)	0.70 (0.53-0.84)	0.73 (0.63-0.86)	42	7	4	M-RNN	LREG	QN->ANSCOMBE->LOG->STD	2To2	delmas
0.77 (0.60-0.91)	0.71 (0.50-0.86)	0.69 (0.58-0.83)	42	7	4	M-RNN	LREG	QN	2To2	delmas
0.77 (0.62-0.89)	0.69 (0.53-0.85)	0.70 (0.58-0.82)	42	7	4	M-RNN	SVM(linear)	QN	2To2	delmas

(a) Top 10 models.

AUC test(95% CI)	Class0 P-test	Class1 P-test	N test	A-days	S-days	Imp	Clf	Pre-proc	Class1	Farm
0.46 (0.31-0.61)	0.61 (0.57-0.64)	0.19 (0.00-1.00)	41	7	4	LI	SVM(linear)	L2	1To2	delmas
0.46 (0.30-0.64)	0.60 (0.59-0.61)	0.00 (0.00-0.00)	33	4	1	LI	SVM(rbf)	L2	1To2	delmas
0.45 (0.26-0.66)	0.54 (0.42-0.63)	0.39 (0.14-0.73)	26	4	1	M-RNN	LREG	QN	1To2	delmas
0.45 (0.32-0.61)	0.60 (0.59-0.61)	0.00 (0.00-0.00)	33	7	1	LI	SVM(rbf)	L2	1To2	delmas
0.45 (0.32-0.56)	0.61 (0.59-0.62)	0.02 (0.00-0.31)	41	7	4	LI	SVM(rbf)	QN->ANSCOMBE->LOG->STD	1To2	delmas
0.45 (0.32-0.56)	0.61 (0.59-0.62)	0.02 (0.00-0.31)	41	7	4	LI	SVM(rbf)		1To2	delmas
0.45 (0.28-0.64)	0.57 (0.56-0.58)	0.00 (0.00-0.00)	26	4	1	M-RNN	LREG	L2	1To2	delmas
0.44 (0.32-0.59)	0.55 (0.52-0.57)	0.04 (0.00-0.77)	34	7	4	M-RNN	LREG	L2	1To2	delmas
0.43 (0.29-0.55)	0.61 (0.60-0.62)	0.00 (0.00-0.00)	41	7	4	LI	SVM(rbf)	QN	1To2	delmas
0.43 (0.24-0.62)	0.57 (0.56-0.58)	0.00 (0.00-0.00)	26	7	1	M-RNN	LREG	L2	1To2	delmas

(b) Bottom 10 models.

Table 3.4: **Comparison of model results for FAMACHA samples classification on the sheep farm.** "Class0" and "Class1" refer to the healthy FAMACHA transition ("1To1") and the unhealthy transition. "P" refer to the model precision. "A" and "S" indicate "Activity" and "Synthetic" respectively. "Imp" shows the imputation technique used. "Clf" shows the type of model used. "Pre-proc" shows the preprocessing used on the sample time series before training (QN: Quotient Normalisation, "STD": Standard Scaling).

AUC test(95% CI)	Class0 P-test	Class1 P-test	N test	A-days	S-days	Imp	Clf	Pre-proc	Class1	Farm
0.68 (0.39-0.94)	0.60 (0.50-0.78)	0.69 (0.27-1.00)	14	4	1	LI	LREG	L2	2To2	cedara
0.66 (0.45-0.89)	0.62 (0.41-0.84)	0.61 (0.34-0.97)	14	4	1	M-RNN	LREG	QN->ANSCOMBE->LOG	2To2	cedara
0.66 (0.45-0.88)	0.61 (0.43-0.83)	0.60 (0.34-0.96)	14	4	1	M-RNN	LREG		2To2	cedara
0.66 (0.45-0.88)	0.61 (0.43-0.83)	0.60 (0.34-0.96)	14	4	1	M-RNN	LREG	QN->ANSCOMBE->LOG->STD	2To2	cedara
0.66 (0.45-0.88)	0.62 (0.41-0.84)	0.61 (0.38-0.96)	14	4	1	M-RNN	LREG	QN->ANSCOMBE->LOG->MINMAX	2To2	cedara
0.64 (0.41-0.87)	0.56 (0.47-0.67)	0.69 (0.00-1.00)	14	4	1	M-RNN	LREG	L2	2To2	cedara
0.64 (0.31-0.88)	0.57 (0.43-0.70)	0.65 (0.00-1.00)	14	7	1	LI	LREG	L2	2To2	cedara
0.64 (0.31-0.89)	0.66 (0.00-1.00)	0.57 (0.39-0.81)	14	4	1	LI	SVM(linear)	QN	2To2	cedara
0.64 (0.39-0.91)	0.68 (0.08-1.00)	0.57 (0.40-0.82)	14	4	1	LI	LREG	QN	2To2	cedara
0.64 (0.17-0.87)	0.62 (0.41-0.84)	0.61 (0.35-0.97)	14	4	1	M-RNN	SVM(linear)	QN->ANSCOMBE->LOG	2To2	cedara

(a) Top 10 models.

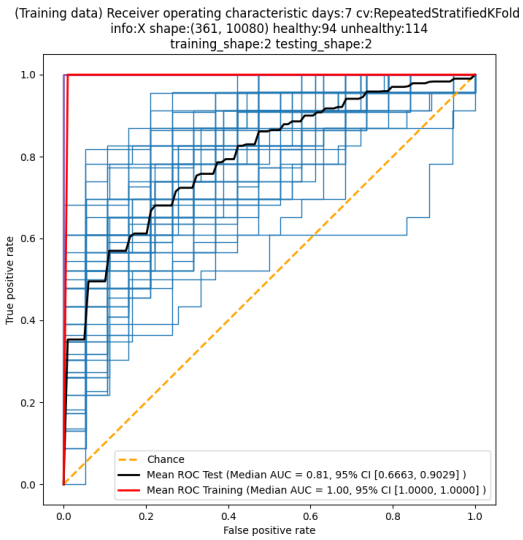
AUC test(95% CI)	Class0 P-test	Class1 P-test	N test	A-days	S-days	Imp	Clf	Pre-proc	Class1	Farm
0.42 (0.10-0.77)	0.54 (0.42-0.62)	0.51 (0.00-1.00)	14	4	1	M-RNN	SVM(rbf)	QN->ANSCOMBE->LOG->STD	2To2	cedara
0.41 (0.20-0.64)	0.68 (0.55-0.82)	0.44 (0.00-1.00)	14	7	4	LI	SVM(linear)	QN->ANSCOMBE->LOG->MINMAX	1To2	cedara
0.41 (0.10-0.75)	0.55 (0.44-0.62)	0.66 (0.00-1.00)	14	4	1	M-RNN	SVM(rbf)	QN->ANSCOMBE->LOG	2To2	cedara
0.41 (0.15-0.77)	0.56 (0.47-0.63)	0.68 (0.00-1.00)	14	7	1	LI	SVM(rbf)	L2	2To2	cedara
0.41 (0.18-0.66)	0.56 (0.46-0.67)	0.62 (0.00-1.00)	14	7	1	M-RNN	SVM(linear)	L2	2To2	cedara
0.41 (0.13-0.76)	0.53 (0.50-0.57)	0.22 (0.00-1.00)	14	4	1	M-RNN	SVM(rbf)	L2	2To2	cedara
0.41 (0.10-0.74)	0.54 (0.42-0.62)	0.56 (0.00-1.00)	14	4	1	M-RNN	SVM(rbf)	QN->ANSCOMBE->LOG->MINMAX	2To2	cedara
0.40 (0.08-0.71)	0.56 (0.45-0.67)	0.67 (0.08-1.00)	14	7	1	LI	SVM(rbf)		2To2	cedara
0.40 (0.08-0.71)	0.56 (0.45-0.67)	0.67 (0.08-1.00)	14	7	1	LI	SVM(rbf)	QN->ANSCOMBE->LOG->STD	2To2	cedara
0.40 (0.12-0.74)	0.56 (0.46-0.62)	0.72 (0.08-1.00)	14	7	1	LI	SVM(rbf)	QN->ANSCOMBE->LOG	2To2	cedara

(b) Bottom 10 models.

Table 3.5: **Comparison of model results for FAMACHA samples classification on the goat farm.** "Class0" and "Class1" refer to the healthy FAMACHA transition ("1To1") and the unhealthy transition. "P" refer to the model precision. "A" and "S" indicate "Activity" and "Synthetic" respectively. "Imp" shows the imputation technique used. "Clf" shows the type of model used. "Pre-proc" shows the preprocessing used on the sample time series before training (QN: Quotient Normalisation, "STD": Standard Scaling).

The training AUC (Area Under the Curve) is a measure of how well a ML model can distinguish between positive and negative classes in the training dataset. It represents the performance of the model on data that it has already seen during the training process. The testing AUC, on the other hand, measures the model's performance on a completely independent dataset that it has not seen during training. It gives an estimate of how well the model will perform on new, unseen data. If the training AUC is higher than the testing AUC, it indicates that the model is overfitting to the training data. Overfitting occurs when a model learns to capture noise and random fluctuations in the training data, rather than the underlying patterns that generalise to new data. To address overfitting, we can use techniques such as regularisation, in practice, we would have to tune the parameter of our models with a parameter grid search which is timely and computationally expensive. In figure 3.22A we show the training and testing AUCs of one of our LREG model and in figure 3.22B one of our SVM models, we can observe that although both models overfit on the training data with the default regularisation parameters the still performed well on the testing datasets, fine-tune parameter tuning may increase the testing performances even further.

A – Sheep farm | Models trained with **unhealthy label 2To2** while healthy is 1To1.



B – Goat farm | Models trained with **unhealthy label 2To2** while healthy is 1To1.

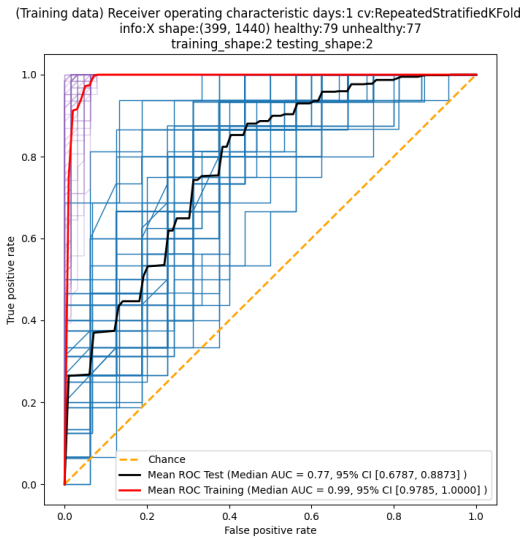
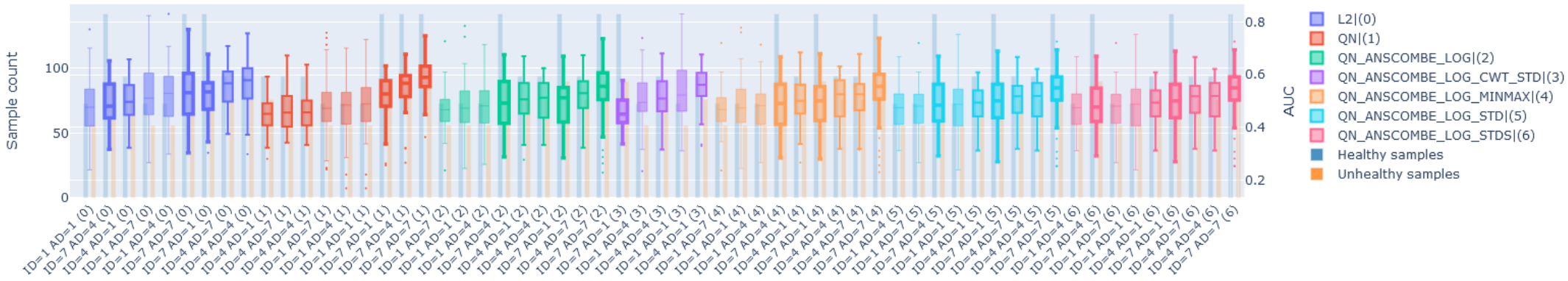


Figure 3.22: **ROC curves of models for classification of health status.** For each figure, the black curve shows the median test auc from the cross-validation while the blue curves show the AUC of each individual test split (section. 3.2.6.1). Similarly, the red curve shows the median AUC for the training data while the purple curves show the AUC of each training split.

A – Sheep farm | SVM(rbf) trained with **unhealthy label 1To2** while healthy is 1To1



B – Sheep farm | SVM(rbf) trained with **unhealthy label 2To2** while healthy is 1To1

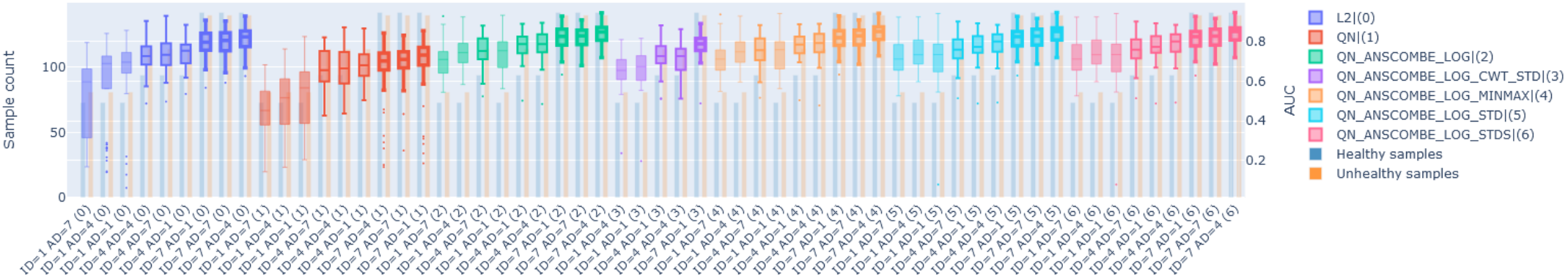
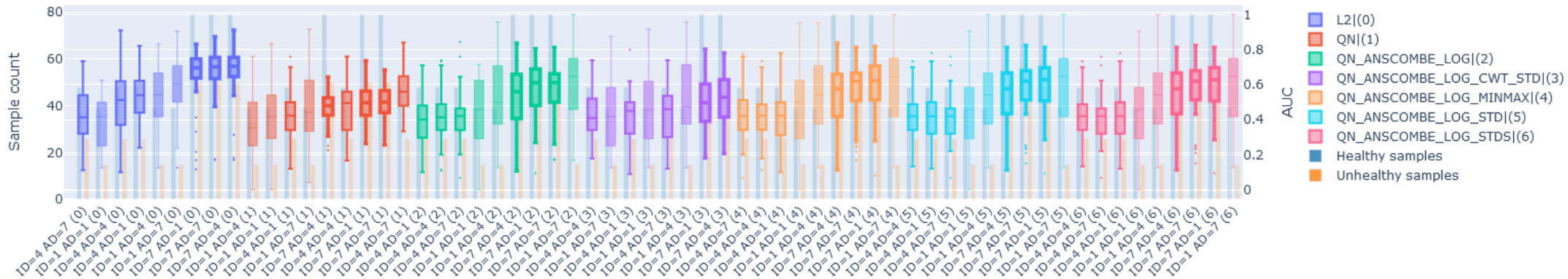


Figure 3.23: **Performance of SVM(rbf) with M-RNN imputed samples for different unhealthy labels on the sheep farm.** Boxplot illustration of different SVM models trained with different versions of the samples. The x-axis shows the number of activity days used in the samples with the notation "AD=", for example, "AD=1" means that the samples contained 1 day of activity data, similarly "ID=" indicates the number of fully synthetic(imputed) days allowed in the sample, the boxplot line thickness also shows the amount of imputed data allowed in the samples. The preprocessing pipeline is indicated by the integer in parenthesis, for example (0) refers to applying L2 normalisation on the samples' time series. The y-axis on the left shows the number of samples in the healthy and unhealthy classes while the right axis shows the AUC.

A – Goat farm | SVM(rbf) trained with **unhealthy label 1To2** while healthy is 1To1



B – Goat farm | SVM(rbf) trained with **unhealthy label 2To2** while healthy is 1To1

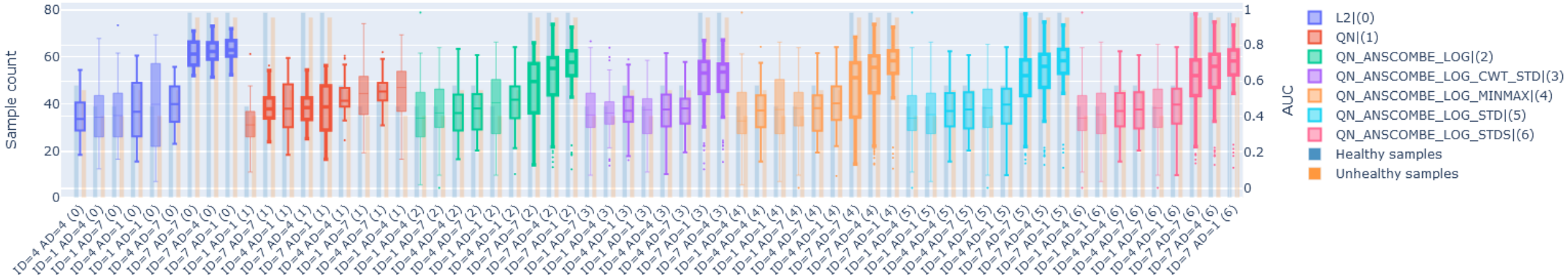


Figure 3.24: **Performance of SVM(rbf) with M-RNN imputed samples for different unhealthy labels on the goat farm.** Boxplot illustration of different SVM models trained with different versions of the samples. The x-axis shows the number of activity days used in the samples with the notation "AD=", for example, "AD=1" means that the samples contained 1 day of activity data, similarly "ID=" indicates the number of fully synthetic(imputed) days allowed in the sample, the boxplot line thickness also shows the amount of imputed data allowed in the samples. The preprocessing pipeline is indicated by the integer in parenthesis, for example (0) refers to applying L2 normalisation on the samples' time series. The y-axis on the left shows the number of samples in the healthy and unhealthy classes while the right axis shows the AUC.

AUC test(95% CI)	Class0 P-test	Class1 P-test	N test	A-days	S-days	Imp	Clf	Pre-proc	Class1	Farm
0.80 (0.80-0.80)	0.83 (0.83-0.83)	0.74 (0.74-0.74)	31	7	1	M-RNN	SVM(linear)	QN->ANSCOMBE->LOG->CWT->STD	2To2	delmas
0.69 (0.58-0.81)	0.63 (0.44-0.86)	0.63 (0.50-0.78)	31	7	1	M-RNN	SVM(rbf)	QN->ANSCOMBE->LOG->CWT->STD	2To2	delmas
0.65 (0.54-0.81)	0.61 (0.43-0.82)	0.63 (0.52-0.78)	31	7	1	M-RNN	LREG	QN->ANSCOMBE->LOG->CWT->STD	2To2	delmas
0.65 (0.53-0.78)	0.63 (0.46-0.87)	0.62 (0.54-0.69)	31	4	1	M-RNN	SVM(rbf)	QN->ANSCOMBE->LOG->CWT->STD	2To2	delmas
0.63 (0.44-0.80)	0.67 (0.46-0.92)	0.54 (0.44-0.66)	37	7	1	LI	KNN	QN->ANSCOMBE->LOG->CWT->STD	2To2	delmas
0.61 (0.48-0.80)	0.70 (0.41-1.00)	0.57 (0.50-0.64)	31	4	1	M-RNN	KNN	QN->ANSCOMBE->LOG->CWT->STD	2To2	delmas
0.61 (0.45-0.75)	0.64 (0.43-0.87)	0.53 (0.40-0.63)	37	4	1	LI	KNN	QN->ANSCOMBE->LOG->CWT->STD	2To2	delmas
0.58 (0.44-0.71)	0.55 (0.36-0.72)	0.57 (0.47-0.69)	31	4	1	M-RNN	LREG	QN->ANSCOMBE->LOG->CWT->STD	2To2	delmas
0.58 (0.42-0.76)	0.61 (0.46-0.76)	0.55 (0.39-0.79)	37	7	1	LI	DTREE	QN->ANSCOMBE->LOG->CWT->STD	2To2	delmas
0.57 (0.39-0.71)	0.60 (0.42-0.73)	0.55 (0.36-0.75)	37	4	1	LI	DTREE	QN->ANSCOMBE->LOG->CWT->STD	2To2	delmas

(a) Sheep farm | Top 10 models with CWT.

AUC test(95% CI)	Class0 P-test	Class1 P-test	N test	A-days	S-days	Imp	Clf	Pre-proc	Class1	Farm
0.63 (0.29-0.89)	0.73 (0.60-0.87)	0.63 (0.00-1.00)	14	7	4	M-RNN	LREG	QN->ANSCOMBE->LOG->CWT->STD	1To2	cedara
0.59 (0.33-0.86)	0.55 (0.37-0.77)	0.53 (0.33-0.75)	18	7	4	LI	LREG	QN->ANSCOMBE->LOG->CWT->STD	2To2	cedara
0.59 (0.38-0.85)	0.55 (0.42-0.70)	0.54 (0.33-0.74)	18	7	4	M-RNN	LREG	QN->ANSCOMBE->LOG->CWT->STD	2To2	cedara
0.59 (0.27-0.82)	0.57 (0.40-0.82)	0.51 (0.25-0.71)	14	7	1	M-RNN	LREG	QN->ANSCOMBE->LOG->CWT->STD	2To2	cedara
0.58 (0.32-0.78)	0.59 (0.33-0.78)	0.58 (0.26-0.84)	18	7	4	M-RNN	DTREE	QN->ANSCOMBE->LOG->CWT->STD	2To2	cedara
0.58 (0.25-0.83)	0.69 (0.43-0.98)	0.38 (0.03-0.61)	14	7	4	LI	KNN	QN->ANSCOMBE->LOG->CWT->STD	1To2	cedara
0.57 (0.29-0.82)	0.58 (0.27-0.81)	0.52 (0.17-0.70)	14	4	1	LI	LREG	QN->ANSCOMBE->LOG->CWT->STD	2To2	cedara
0.57 (0.27-0.86)	0.59 (0.28-0.83)	0.51 (0.22-0.71)	14	7	1	LI	LREG	QN->ANSCOMBE->LOG->CWT->STD	2To2	cedara
0.56 (0.14-0.86)	0.73 (0.57-0.88)	0.63 (0.00-1.00)	14	7	4	M-RNN	SVM(linear)	QN->ANSCOMBE->LOG->CWT->STD	1To2	cedara
0.55 (0.32-0.74)	0.58 (0.31-0.83)	0.52 (0.21-0.74)	14	7	1	LI	DTREE	QN->ANSCOMBE->LOG->CWT->STD	2To2	cedara

(b) Goat farm | Top 10 models with CWT.

Table 3.6: **Comparison of model results for health classification with frequency domain data (CWT).** "Class0" and "Class1" refer to the healthy FAMACHA transition ("1To1") and the unhealthy transition. "P" refer to the model precision. "A" and "S" indicate "Activity" and "Synthetic" respectively. "Imp" shows the imputation technique used. "Clf" shows the type of model used. "Pre-proc" shows the preprocessing used on the sample time series before training (QN: Quotient Normalisation, "STD": Standard Scaling).

3.3.2.1 Model evaluation on unseen labels

We plot density histograms of the prediction probability of our cross-validated best model (Tables. 3.5a and 3.4a) to evaluate how well the model classifies unseen samples. The model will classify a sample as unhealthy if the probability output is ≥ 0.5 (marked as a vertical grey dotted line) (Fig 3.26).

We show the result for the sheep farm in figure 3.25A. An SVM model was trained on healthy famacha (1To1, 142 examples) and unhealthy FAMACHA (2To2, 140 examples) and tested with repeated k-fold cross-validation on all of the samples available in the dataset including the ones used for training. When the density histogram values are skewed to the left it means that the model classifies the data as healthy while on the opposite side, it will classify as unhealthy. As expected the model perform well by properly classifying the label used for training "1To1" and "2To2" while the "1To2" and "2To1" transitions were mostly classified as healthy, arguably both are still in the healthy range of FAMACHA.

The goat farm figure 3.26A shows the results. Similarly, An SVM model was trained on healthy famacha (1To1, 79 examples) and unhealthy FAMACHA (2To2, 79 examples). Although in small numbers the goat farm had more variation of FAMCHA transitions that our model could be tested on, as expected the density histogram on the trained label shows mostly correct classifications, for the advanced stage in FAMCHA such as "2To5", "4To5", "5To3", "5To2", "4To3" and "4To1" our model perfectly classified all samples as unhealthy, even for the "1To2" transition the model seems to start "understanding" health decline as a majority of samples were classified as unhealthy instead of healthy with the demarcation getting clearer with "1To3", "1To4", "2To3", "2To4", "3To4", "3To1", "3To2" and "4To2". Only "3To5" showed mitigated results where the model performance was poor. For reference, the number of available samples in each group is stated in figure 3.2.

A – Sheep farm | Model trained on healthy 1To1 and unhealthy 2To2

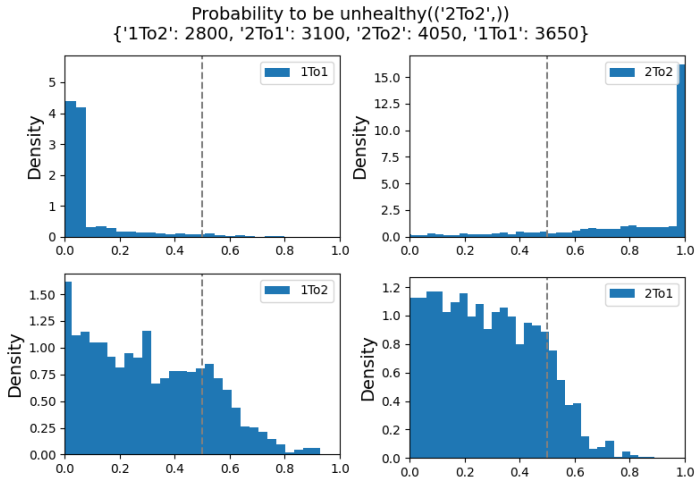


Figure 3.25: **Density histograms of prediction results for transitions used in training.** Density histograms are a way to represent the distribution of a set of continuous data. They are similar to regular histograms, which shows the frequency of values within certain intervals (bins) of a continuous variable. However, in a density histogram, the y-axis represents the density of data points within each bin, rather than their frequency. The density is calculated as the number of data points in each bin divided by the total number of data points multiplied by the width of the bin. This ensures that the area under the histogram equals 1, which makes it possible to compare the relative proportions of data in different bins, even if they have different widths.

A – Goat farm | Model trained on healthy 1To1 and unhealthy 2To2

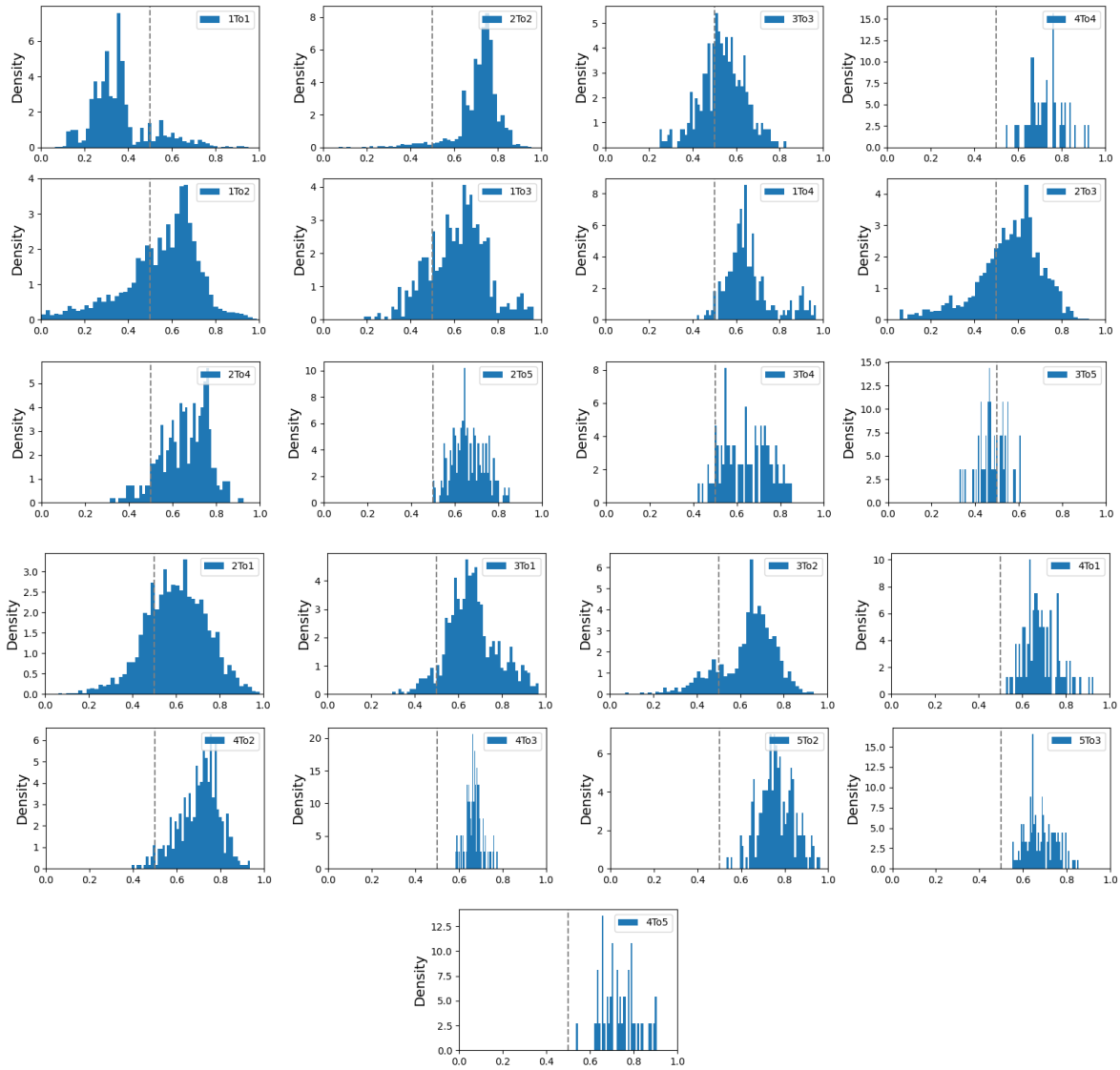


Figure 3.26: **SVM predictions on sheep unseen samples.** Density histograms of the model predictions on unseen labels. Each histogram shows the probability output on the x-axis and the density on the y-axis. The dotted grey vertical line marks $Probability = 0.5$. While (A) shows the results for the sheep farm (B) shows the goat. For (B) the FAMACHA transitions tested from top to bottom and right to left are: 1To1, 2To2, 3To3, 4To4, 1To2, 1To3, 1To4, 2To3, 2To4, 2To5, 3To4, 3To5, 2To1, 3To1, 3To2, 4To1, 4To2, 4To3, 5To2, 5To3 and 4To5.

3.3.2.2 Adding Exogenous features

In figure 3.27 we added exogenous features in our samples or only used exogenous features without activity data (Fig 3.6). It has already been discussed in Chapter 2 that the parasitic burden of the animals is directly correlated to the rainfall on the farm. As expected when using

the rainfall data (black box plot) the predictive power of our SVM model was high with a 76% median AUC with the 84 days of rainfall data before the FAMCHA test, the predictive power reduced significantly to 65% AUC when using a short 7 days window of the rainfall before the test which can be explained by the fact that the worm hatching time onset. Using a mix of Activity data and rainfall (orange box plot) gave an AUC above 80%.

3.3.3 Classification of response to treatment

We had 50 examples of FAMACHA $2 \rightarrow 1$, 67 for FAMACHA $1 \rightarrow 1$ for the sheep, and 53 examples of FAMACHA $2 \rightarrow 1$ and 79 examples of FAMACHA $1 \rightarrow 1$ for the goats. This resulted in a modest training set size which may diminish the performance of the ML (Fig. 3.2). Nevertheless, the classifier was able to predict a drop in FAMACHA score indicating a response to treatment with mean precision of 64% and 55% for the sheep and goats, respectively, and no change in FAMACHA score with mean precision of 61% and 66%. As shown in Fig. 3.28, the median AUC was 65% for the sheep and 70% for the goats.

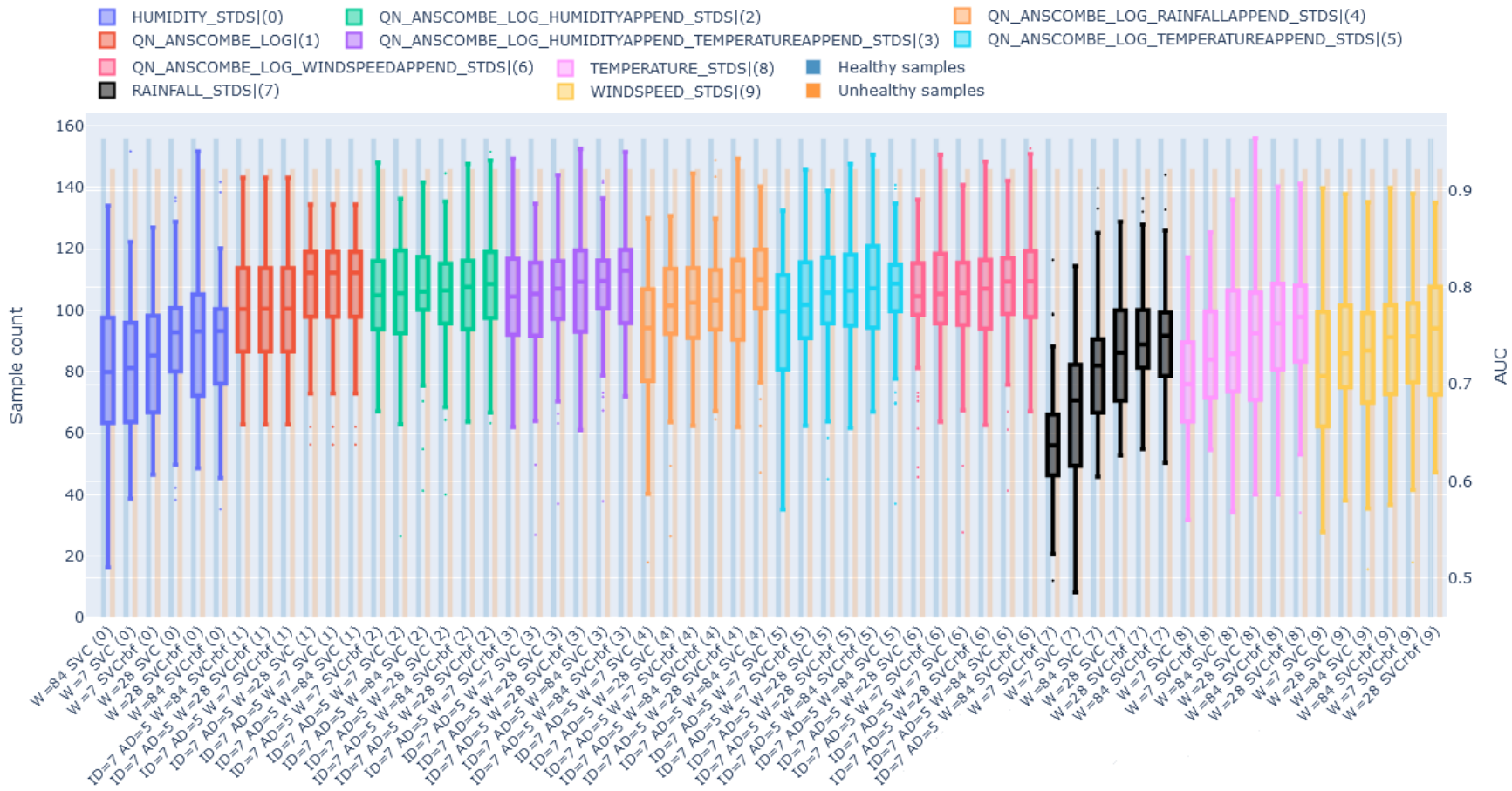


Figure 3.27: **Performance of SVM model with exogenous features for the sheep farm.** The x-axis shows the number of activity days via the notation "AD=", the number of synthetic days with "ID=" and "W=" for the number of days for the weather. For example, if the x-axis shows "ID=7 AD=5 W=84 SVC(0)" this means that if the samples contain activity data, a sample can contain up to 7 synthetic days, the activity time series contains 5 days from the first famacha test and the sample contains 84 days of weather (rainfall, temperature or wind speed) data. The models that were trained with a combination of activity and weather features are annotated with the word "APPEND" in the legend. The left y-axis shows the number of samples while the right axis shows the AUC.

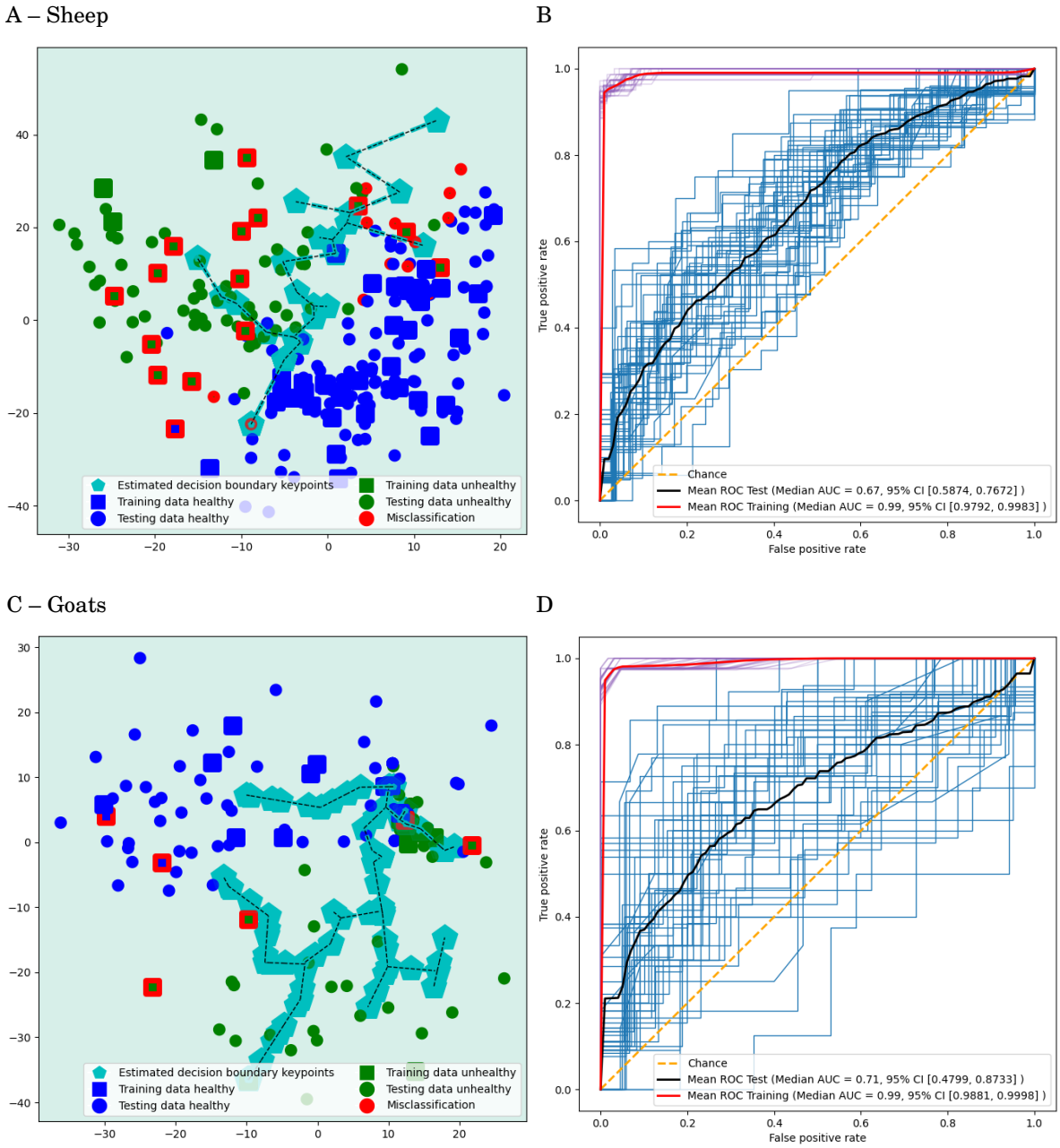


Figure 3.28: **Classifying the response to treatment.** For both farms the ML was trained to discriminate FAMACHA 2→1 against FAMACHA 1→1 using 7 days of accelerometry data directly proceeding anthelmintic treatment for the Sheep farm and 7 days for the Goat farm. (A) DB plot and (B) Receiver Operating Characteristic (ROC) curve for training and testing on sheep. (C) DB plot and (D) ROC curve for training and testing on goats.

3.3.4 Temporal validation

Two equally-sized periods of data were extracted from the sheep and goat datasets to maximise seasonal differences. For the sheep farm, the first period was from March 2015 to October 2015, and the subsequent period from October 2015 to April 2016. The first period contained 136 (54 healthy and 82 unhealthy) samples and 151 (96 healthy and 55 unhealthy) in the second period. It is clear from observing the resulting scatter plots of the trained classifiers (Fig. 3.29) that while there is little observable drift in the activity of the healthy animals, the less healthy animals cluster differently depending on the time period. Nevertheless, these clusters do not interfere with the DB and hence the overall precision of prediction did not noticeably decrease, yielding a mean AUC of 70% for the sheep when training on the first period and testing on the second and 74% when doing the reverse.

For the goat farm, only 46 (26 healthy and 20 unhealthy) samples could be built for the first period and 51 (14 healthy and 37 unhealthy) samples for the second one, which yielded a mean AUC of 58% when training on the first period and testing on the second, while we obtained a 50% mean AUC when doing the reverse. Because of the limited number of samples that was available on the goat farm, the latter results are inconclusive.

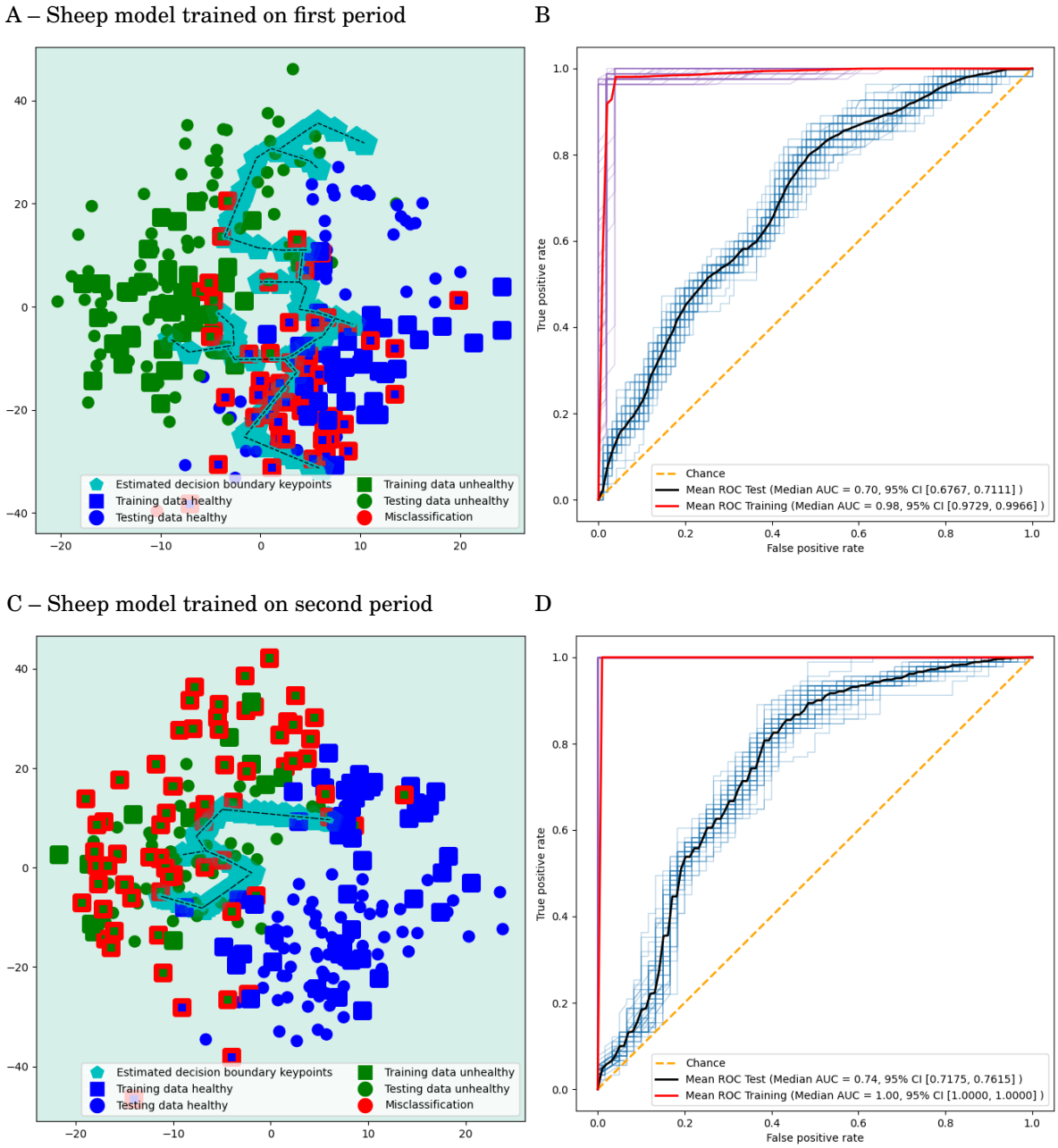


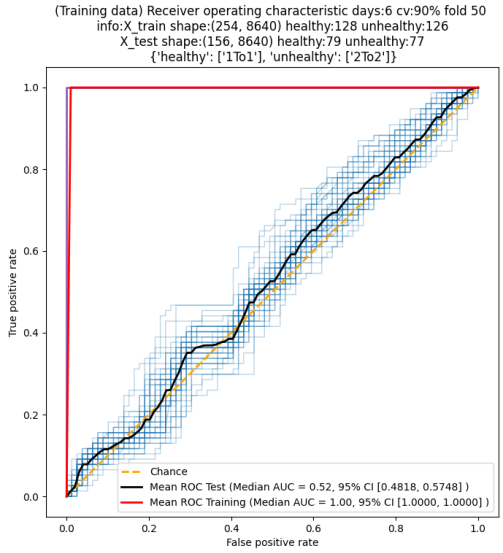
Figure 3.29: **Temporal validation for the sheep farm.** (A) Scatter plot and (B) ROC curve for training on the first period and testing on the second. (C) Scatter plot and (D) ROC curve for training on the second period and testing on the first.

3.3.5 Cross-farm validation

While a high degree of predictive power was achieved by training the model on each farm independently, a useful requirement for the practical application of our technique would be the ability of the classifier to generalise across farms without retraining. We evaluated the general-

isability of our approach from the goats to the sheep and vice-versa. We trained a SVM(linear) model with 90% (randomly selected) of the healthy (1To1) and unhealthy(2To2) samples from a farm and test on the other, by repeating the process 50 times we obtained the ROC curves in figure 3.30. For both farms, the samples were pre-processed with the same pre-processing steps: QN→ANSCOMBE→LOG→STDS. Furthermore, we use the 7-day version of the samples with a maximum of 7 days of fully synthetic data. The results demonstrate that cross-farm generalisability was not achieved, most likely due to the size of the datasets and too big of a difference between sheep and goat accelerometry data and behaviour.

A



B

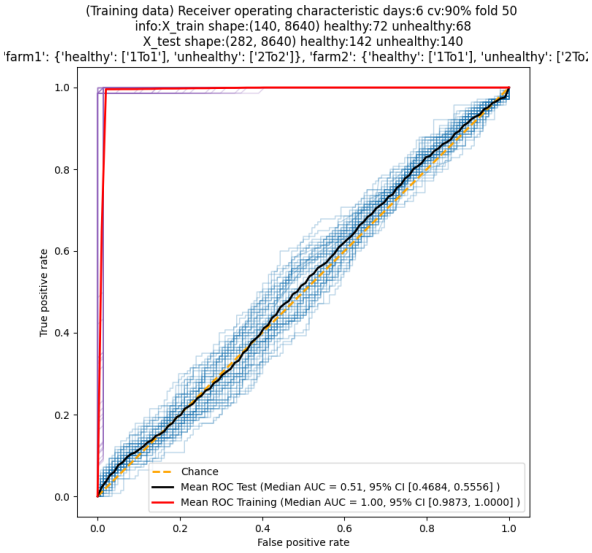


Figure 3.30: **Cross farm validation.** Figure (A) show the ROC curve for training on the sheep farm data and testing on the goat farm data while (B) shows the opposite. Testing curves appear in blue while training curves are in red.

3.3.6 Evolution of predictive power over time

We wish to establish how the predictive power of our model behaves as we move back in time, i.e. an earlier prediction of ill health (Figure 3.31). In figure 3.32 we trained and tested a SVM classifier to classify samples with FAMACHA 1→1 against 2→2. We used day-long samples and shifted their time frame to an increasingly earlier time from the first famacha test as described in the figure 3.32.

The AUC was 70% for the first 4 days and then decreased to approximately 60% and the sheep followed while the decrease started after the third day for the goats. This result confirms the hypothesis that the activity level of the animal shortly before (up to 7 days) FAMACHA evaluation is most representative of the animal's health.

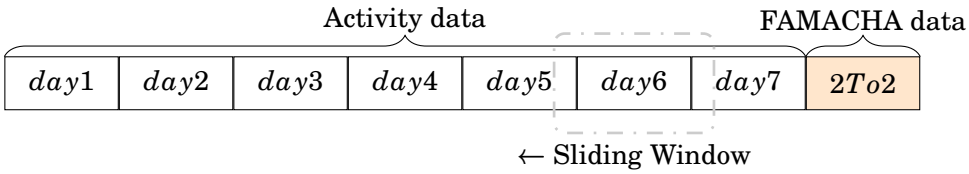


Figure 3.31: **Schematic of sliding window ML sample**. In this illustration we used a day-long sliding window, to capture the activity component of the sample, in this example, the sample label is 2To2, i.e the FAMACHA score remained 2 across two successive weeks.

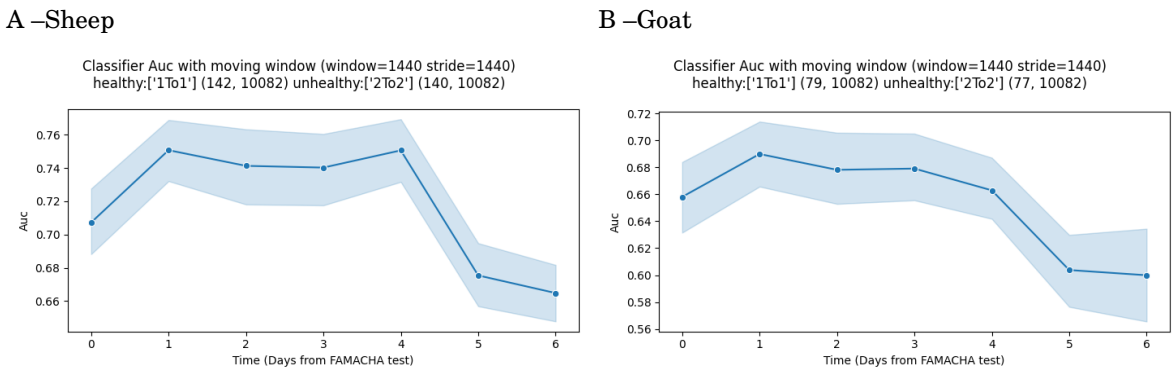


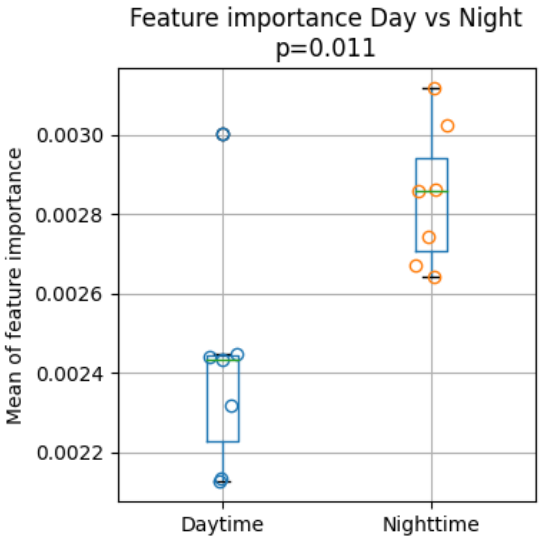
Figure 3.32: **Evolution of predictive power over time.** The x-axis shows the number of days before the first FAMACHA test while the y-axis shows the AUC.

3.3.7 Model explainability

We train a linear SVM model with all of the samples available and extract the coefficients of its DB 1.3 function which are linearly proportional to the importance of the features (each minute in the time series) [55] [21] [161]. For the different sample lengths, we train a SVM model and visualise the feature importance in the time domain, we also segmented the activity data into different time frames namely the daytime and the night time to evaluate which is most important. We centred our daytime segment around the median midday time and the median midnight time for the night time, the exact times were obtained with the use of a public historic weather data API (API, or Application Programming Interface, is a standardised set of protocols and tools that facilitate seamless communication and data exchange between different software applications) for each farm (Fig. 3.33). In figure 3.34 and 3.35) we can observe the daytime and nighttime importance of the sheep and goats by plotting the mean of the feature importance of each time frame for 7 consecutive days, we can use the p value of the two groups (array of daytime mean importance and array of night time mean importance) to evaluate if there is statistically significant differences between the daytime and nigh time activity [66]. We obtained a p value of 0.011 which is lower than the typical 0.05 threshold [33] that indicates that the observed

differences are unlikely to be due to chance while for the goat we obtain a higher p value of 0.535 which doesn't allow us to draw a similar conclusion for the goats.

A –Sheep 7 day



B –Goat 7 days

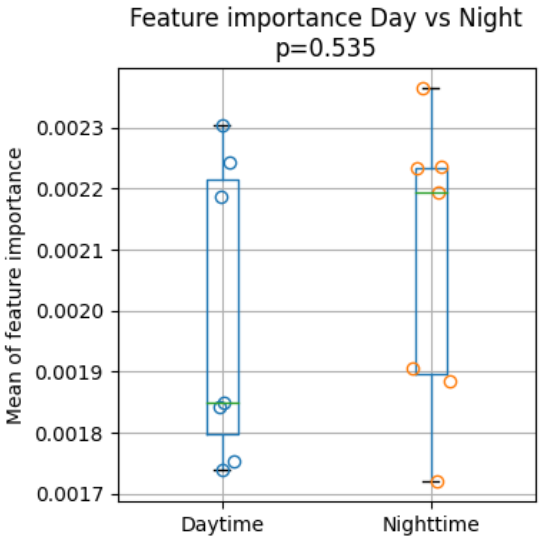


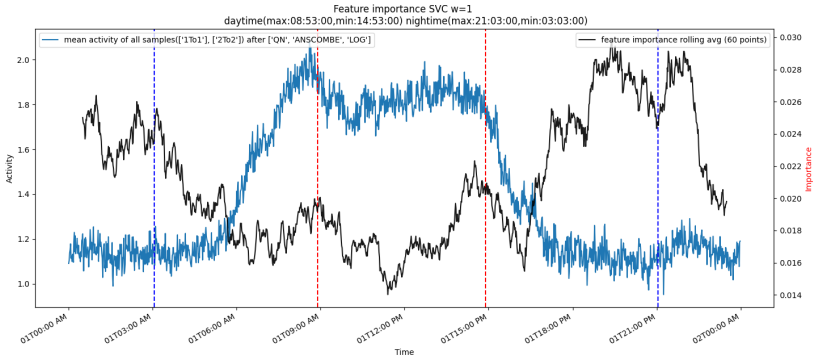
Figure 3.33: **Feature importance during the day and night for the sheep farm.** For each boxplot we plotted the mean feature importance of the daytime and night times for 7 days of activity preceding the first FAMACHA test.

3.4 Conclusion

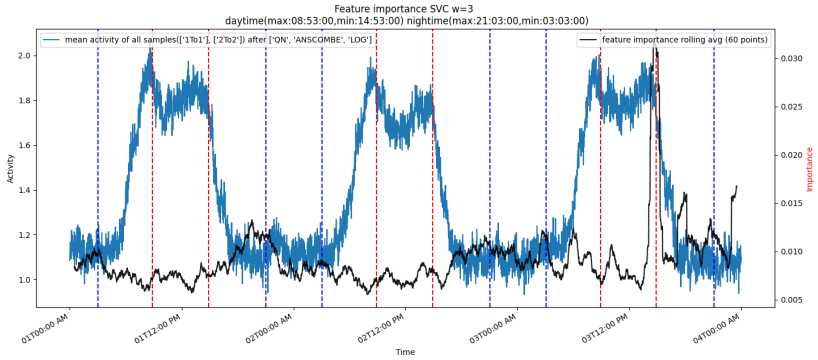
In this chapter, we have devised a ML pipeline that can classify FAMACHA score transitions based on animal activity data alone. We discovered that it is possible to classify activity data of goats and sheep who had a FAMACHA score of 2 for two consecutive tests with high accuracy, 87%(0.77-0.94) AUC for the sheep and 0.78%(0.68-0.89) AUC for the goats (Figure 3.22). We were also able to classify the animals' responses to treatment by training our models with activity samples presenting FAMACHA transitions of $2 \rightarrow 1$ and $1 \rightarrow 1$. We also tested against concept drift and found that although predictive power decreased when training and testing on two distinct time periods it remains high (AUC $\geq$ 70%) given enough samples. Furthermore, we tested our model generalisability by training them on the data of the goat farm and testing them with the data of the sheep farm and vice-versa, however this was not successful. The low performance can be explained by multiple factors, firstly the activity of goats and sheep although similar are inherently different, secondly the topography of the farms are different, thirdly the hardware of the transponders used varied from farm to farm and finally even after imputation the number of samples available is fairly low. In addition, we tested our model on the data we decided to not train with, namely the samples with higher FAMACHA scores ($\geq$ 3) and discovered that

despite being trained with "healthier" samples (FAMACHA $2 \rightarrow 2$) it was possible to accurately classified unseen samples with FAMACHA score increasing to values above 2. In an attempt to add interoperability to our results pipeline, we analysed our models to reveal what features are most discriminative for classification, we found that for the sheep the night time activity is most important, while the goats the results are inconclusive. Lastly, we also investigated the impact of exogenous factors on the classification, we focus on weather data including rainfall, temperature and wind speed. It has been established in Chapter 2 that FAMACHA is highly dependent on rainfall as wetter weather drastically increases *(H. contortus)* populations and infection of small-ruminant. We found that using the rainfall alone as features for the prediction of FAMACHA (FAMACHA $2 \rightarrow 2$) yields predictive power up to $75\% AUC$. However, using a combination of activity data and rainfall data offered better results. Furthermore, fine-tuning our ML algorithm hyperparameters for regularisation may increase performance in future work. While efforts in developed countries have been focused on building high-precision, securely-mounted and precisely fitted sensors, we demonstrate that robust results can be gained from much simpler, low-cost systems with rudimentary maintenance requirements suitable for both commercial and ressources-poor farmers in developing countries. It is important to note that due to significant calibration and mounting variation between transponders, including loosening of the transponder over time, it was necessary to perform normalisation of each activity trace to the herd/flock mean. This meant that uniform reductions in activity level from week to week are likely to be normalised out of our data. Nevertheless, a biological reason for a completely uniform reduction in activity level is implausible; instead, intensities of some daily activities are likely to be impacted more than others. We have shown that changes in the variation of activity levels constitute a strong predictor of early changes in health status, as regards haemonchosis and are robust to technical variation.

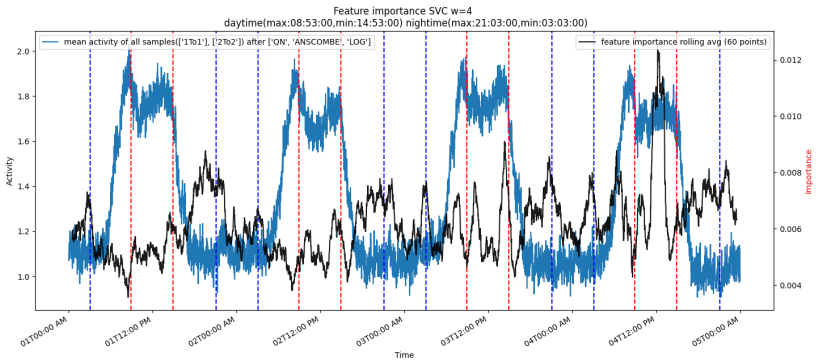
A – 1 day



B – 3 day



C – 5 day



D – 7 day

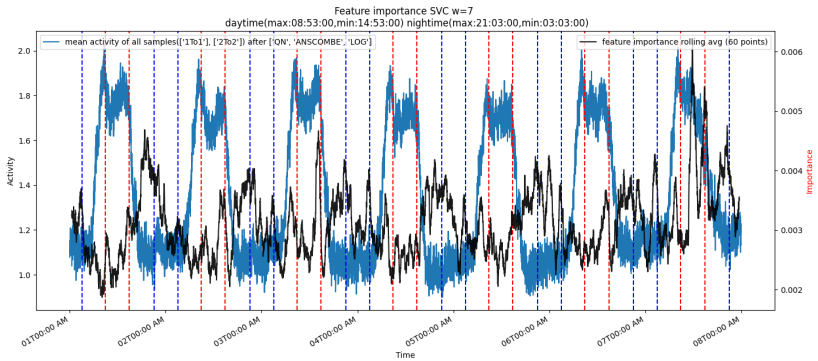
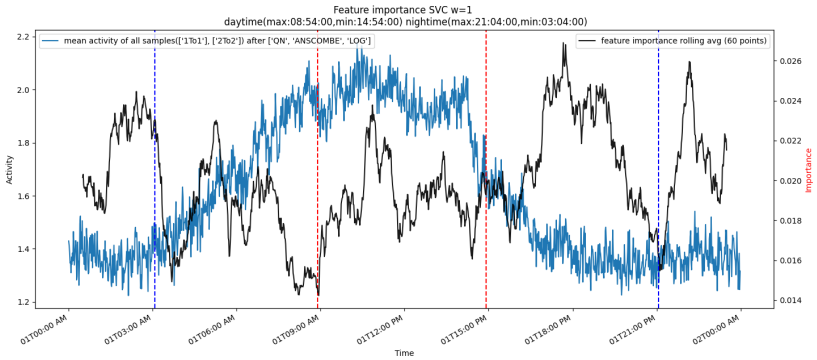
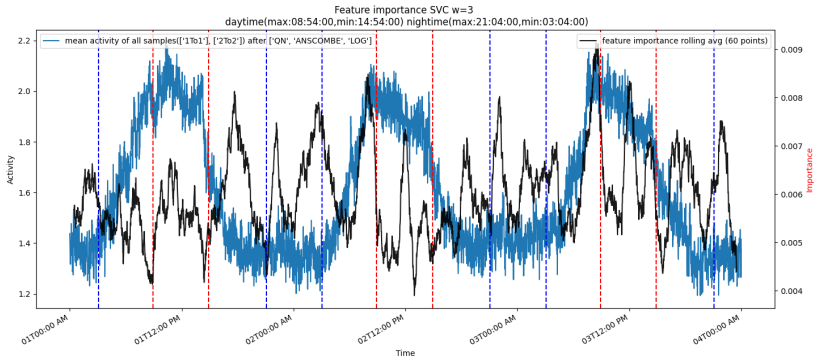


Figure 3.34: **SVM features importance for the sheep farm.** For every figure, the x-axis shows the time while the left y-axis shows the pre-processed activity value and the right y-axis the feature importance. The blue curve shows the mean of all of the samples, the black curve shows the feature importance. The red dotted vertical lines show the daytime segments, similarly, the blue vertical lines display the night time.

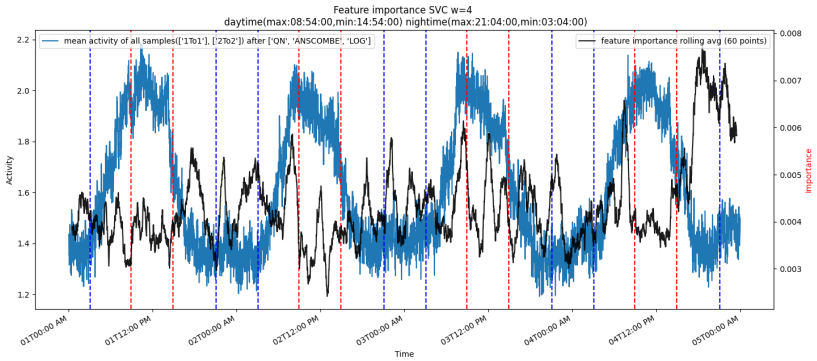
A – 1 day



B – 3 day



C – 5 day



D – 7 day

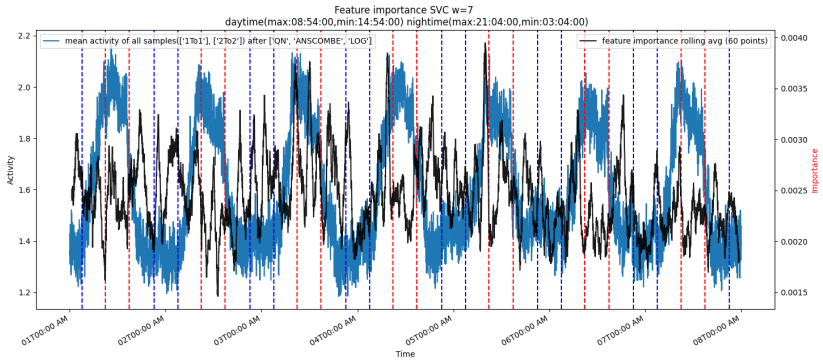


Figure 3.35: **SVM features importance for the goat farm.** For every figure, the x-axis shows the time while the left y-axis shows the pre-processed activity value and the right y-axis the feature importance. The blue curve shows the mean of all of the samples, the black curve shows the feature importance. The red dotted vertical lines show the daytime segments, similarly, the blue vertical lines display the night time.

CHAPTER 4

TRANSLATION TO COMPANION ANIMAL ACTIVITY MONITORING

In Chapter 4, we expanded our supervised ML pipeline for classifying activity data originally tailored for small ruminants to categorize the activity patterns of domestic cats. This chapter concentrates on addressing Degenerative Joint Disease (DJD), commonly known as arthritis, in felines. DJD, distinguished by irreversible cartilage degradation, induces joint inflammation and discomfort during movement. Given the susceptibility factors like obesity and joint surgery, DJD prevails among cats. The study aims to foresee initial indications of DJD in indoor cats, employing accelerometers and ML techniques. The hypothesis posits that the impact of DJD is more conspicuous during heightened activity. The investigation, deploying Actical® Z accelerometers, enlisted 85 cats, with comprehensive data amassed through proprietor surveys, VetMetrica evaluations, and orthopaedic examinations. The analysis encompasses constructing ML samples grounded in peak activity occurrences, contributing to early DJD identification and enhancing the well-being of cats.

In Chapter 4 we devised an activity data classification pipeline with supervised ML and accelerometers for small ruminants. The activity time series was revealed to contain enough information to characterise the healthy and lower-health status of sheep and goats. In this chapter, we demonstrate the flexibility of our pipeline by adapting it to classify the activity data of domestic cats.

4.1 Introduction

DJD, also known as arthritis, affects the cartilage within the joints. With age cartilage naturally wears down, however irreversible deterioration of the articular cartilage is characterised as DJD. The condition causes inflammation in the joint and pain when the affected subject moves. Known risk factors of DJD in cats include obesity, bone or joint surgery. It is estimated that most cats of

all ages have DJD [134], [81]. Because of the high frequency of DJD in cats, an accurate detection system can be a valuable tool to improve indoor cats' quality of life.

Clinical signs of decreased mobility are symptoms of DJD in cats and other felines, and are highly prevalent [134], while radiographic evaluation allows accurate assessment of the condition, clinical examination of cats' pain response is not always possible and there is no universal objective assessment method for DJD caused pain in cats [134]. Gruen et al [54] used collar-mounted accelerometers (Fig. 4.1) to compare the mean activity of cats with minimal signs of DJD to cats with DJD over a 7-day period and found that cats exhibited a sharp peak of activity in the morning and broader peak in the evening. They also found that cats with DJD showed different patterns of activity from cats without DJD, although activity and intensity were not always lower.

The development of an accurate evaluation model can help to alleviate and treat symptoms faster and increase cats' well-being. The aim of this study was to predict early signs of degenerative joint disease in indoor cats with the use of accelerometers and ML techniques. The study hypothesis was that the effect of degenerative joint disease would reflect more in a cat's activity when it performs higher activity behaviour such as jumping, running quickly etc.

Figure 4.1: **Cat wearing a Actical® Z wearable accelerometer based sensor.** Client-owned cat wearing a collar-mounted Actical® accelerometer in the typical position on the neck. The image is taken from source [54].

4.2 Study group

Recruitment for this study finished in November 2019, with a total of 85 cats recruited [88]. Of the 85 participants enrolled, 56 cats were fitted with accelerometers (Fig. 4.1) and data from 52 of these were included in the analysis. Out of the 56 included cats, 30 cats had early signs of DJD and 27 cats were disease-free based on multiple assessment methods.

An owner-completed questionnaire was distributed evaluating their cat's mobility based on the Feline Musculoskeletal Pain Index (FMPI), which consists of 18 questions that evaluate the cat's mobility, grooming and behaviour related to pain. The questionnaire is completed by the cat's owner or caregiver and is designed to provide a subjective measure of the cat's pain and disability. The FMPI can be used to monitor the cat's response to treatment over time and help guide the management of chronic pain associated with musculoskeletal disease [10]. VetMetrica questionnaires were also conducted, which are a set of tools used to assess various aspects of a cat's health and well-being, including chronic pain, quality of life, and appetite. The questionnaires were also completed by the cat's owner or caregiver and provide a subjective measure of the cat's condition. The results can be used by veterinarians to monitor the cat's health over time and guide treatment decisions. The VetMetrica questionnaires are designed to be user-friendly and can be completed online or on paper [27].

In addition, all eligible cats were visited in their own home and an orthopaedic examination was performed by a veterinarian to accurately assess their DJD status. An orthopaedic examination for DJD in cats involves a thorough evaluation of the cat's medical history, physical examination, joint palpation, range of motion, manipulation and stress testing, and radiographs. The examination aims to assess joint mobility, stability, and pain, and to rule out other potential conditions that may cause similar symptoms.

This study differs from the small-ruminant study described in previous chapters for many reasons. Although accelerometry data remained the primary measurement for analysis, the wearable sensors used here are more expensive, including memory on the device for recording with minimum noise and missingness (≤ 0.5 % of missing data points in the data set), therefore imputation was not required. The study length for small-ruminants was also significantly longer and span over multiple years while here the data collection time only lasted 14 days. While the sheep and goats' health status was based on a gradual shift in health caused by parasitic burden increase, here we adopted a binary approach where the cat is said healthy if it has DJD and unhealthy otherwise. Individual sheep and goats' health status fluctuated multiple times from healthy to unhealthy and vice versa during the study time, while here each cat always had the same health status. For this study, we used Actical® Z wearable devices (Philips Respironics, Bend, Oregon USA). The raw data takes the form of 57 MS Excel spreadsheets that contain the sensor data, in addition to a metadata spreadsheet (Table. 4.1) which describes the joint health of each cat assessed using an orthopaedic examination conducted by a veterinary surgeon was provided, as well as other metrics such as the age (Fig. 4.2) and the mobility score of the cats.

Cat ID	Age	Status	Mobility_Score
12	7.76	0	1
128	6.98	0	1
75	6.78	0	1
621	9.22	0	1
40	8.45	0	1
77	7.71	0	1
634	11.96	0	1
57	6.81	0	1
635	7.49	0	1
24	7.79	1	0.83
78	6.85	0	1
602	6.69	0	1
609	6.37	0	1
604	10.41	0	1
658	11.81	1	0.86
612	8.7	0	1
611	9.38	0	1
626	9.03	1	0.58
31	8.51	0	1
610	6.43	0	1
655	10.92	0	1
11	7.78	0	1
601	6.52	0	1
608	9.97	1	0.82
613	15.31	1	0.58
620	8.17	0	1
646	6.37	1	0.88
600	11.54	1	0.71

Cat ID	Age	Status	Mobility_Score
622	11.05	1	0.83
632	7.62	0	1
637	16.59	1	0.83
649	8.63	1	0.83
650	8.63	1	0.83
36	8.28	1	0.83
627	9.16	1	0.83
645	16.25	1	0.83
647	8.5	1	0.75
652	14.53	0	1
661	6.4	1	0.88
603	19.97	1	0.67
630	11.33	1	0.88
631	6.33	1	0.83
618	10.78	1	0.83
662	14.15	1	0.83
642	7.42	1	0.46
607	15.01	1	0.92
614	13.12	1	0.75
660	13.25	1	0.73
13	7.03	1	0.83
619	11.13	1	0.71
26	7.84	0	1
129	7.15	0	1
648	15	1	0.88
651	14.53	0	1
605	11.58	1	0.82
613	15.31	1	0.58

Table 4.1: **Metadata available for the cat population of the study.** Each cat was assigned a mobility score assessed by its owner. The Status was given based on the mobility score, for a score > 0.5 the status is 1 otherwise it is 0. The age (in years) of the cat at the start of the study was also reported.

Three cats were excluded from the current analysis as their sensors were calibrated differently from the rest of the population. This is done as a safeguard to avoid biasing the analysis, although post-processing approaches may tackle such an issue. In table 4.1 we present the metadata of each cat included in the analysis, the "Cat ID" column shows the unique identification number of the cat, "Age" shows the age of the cat in years, "Status" gives the health status, 0 for healthy and 1 for unhealthy, finally "Mobility Score" shows the questionnaire assessed mobility scores of the cats.

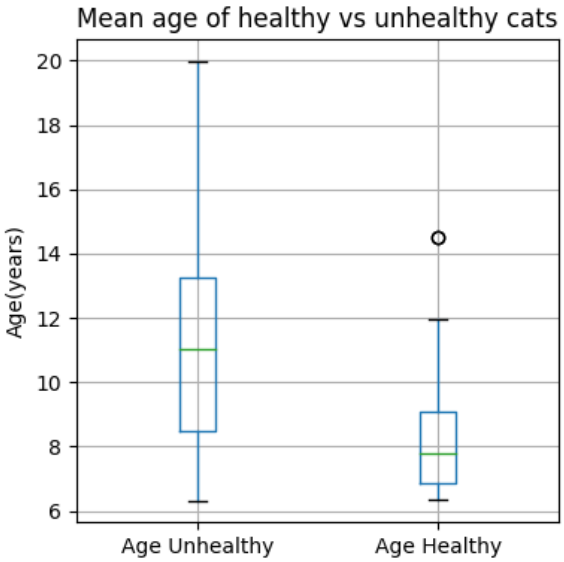


Figure 4.2: **Cats age.** Boxplot showing the age of the cats used in this study.

4.3 Methods

Five sensors were set up to measure activity counts every millisecond while the rest collected activity counts every second. For this project, all the activity data were resampled to the second resolution (i.e. each count contains the sum of activity counts within each second) if it was not already at that resolution.

In figure 4.3 we show the activity data (after Anscombe and Log transformation) of all the cats including the ones not used in the analysis. Each row shows the activity count data of a single cat during the entirety of the study. We can easily observe that 3 cat traces show a much higher magnitude of activity compared to the others, this is likely due to the sensor calibration and was taken into consideration for the analysis. Compared to the small-ruminant dataset (Fig. 3.4A) no obvious day/night cycle can be seen, each cat has a unique behaviour . Note that the resolution of the data is also higher at 1 measurement per second for the cat study and 1 measurement per minute for the small-ruminants.

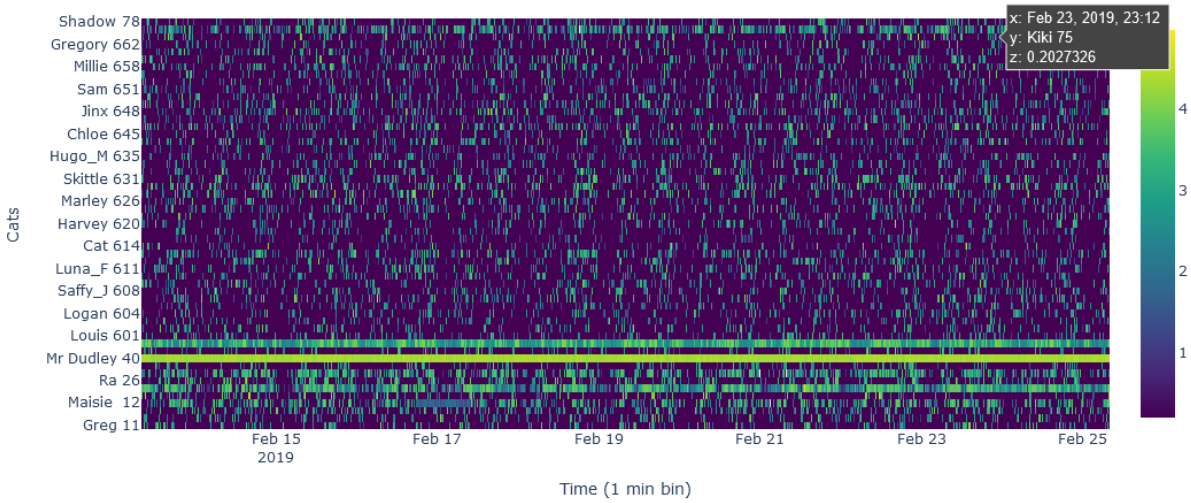


Figure 4.3: **Cats accelerometer heatmap.** This Heat-map shows the accelerometry data of each cat during the study time. The raw data counts are pre-processed with the Anscombe and the Log transform.

4.3.1 Building samples

As in our developed ML pipeline in chapter 3, we used a supervised ML approach which requires us to build samples that are composed of an array of features and a ground truth value that will be the value we want to predict. In this study case, we decided to solely use activity counts as features and the health score as ground truth. Unlike herded free-range livestock, exogenous factors such as the weather are unlikely to be of significance in indoor cats' activity level or health.

Our sample-building strategy needed to differ from the "n day of activity" approach we used for small-ruminants, firstly because cats do not have a fixed daily routine like the sheep and goats in a herd, and secondly because the data was collected on a much shorter time frame. Multiple samples were built per cat to overcome the limitations of a relatively small sample size. To reflect the study hypothesis that DJD would have its greatest effects during higher activity events, samples were created by looking for peaks of activity in each cat's trace and selecting a fixed amount of activity data before and after the peak occurred, effectively building a window around the peaks (Fig. 4.5A). This also had the effect of aligning the samples (i.e. by the point of impact) which was not necessary for the small ruminants who all had a regular daily schedule. We fixed the window length to different values and selected the top 0.001% peaks based on the activity count. With this approach, we were able to generate 10 peaks per cat. which gave a total of 520 samples (Fig 4.4).

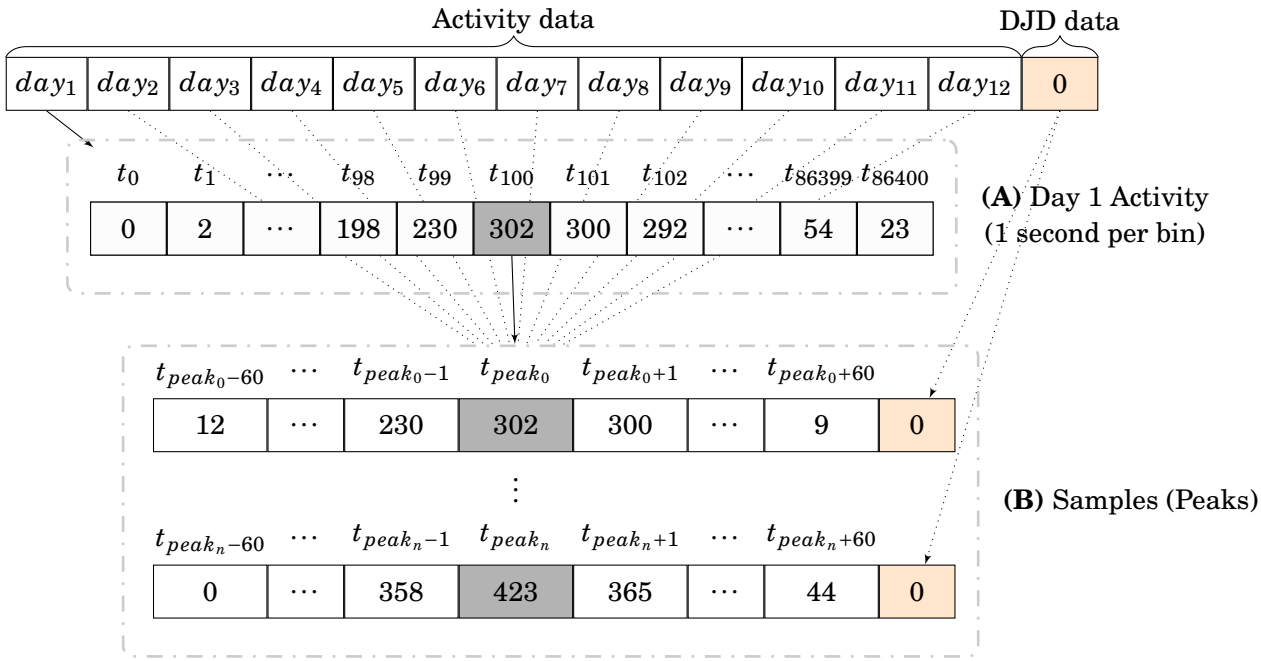
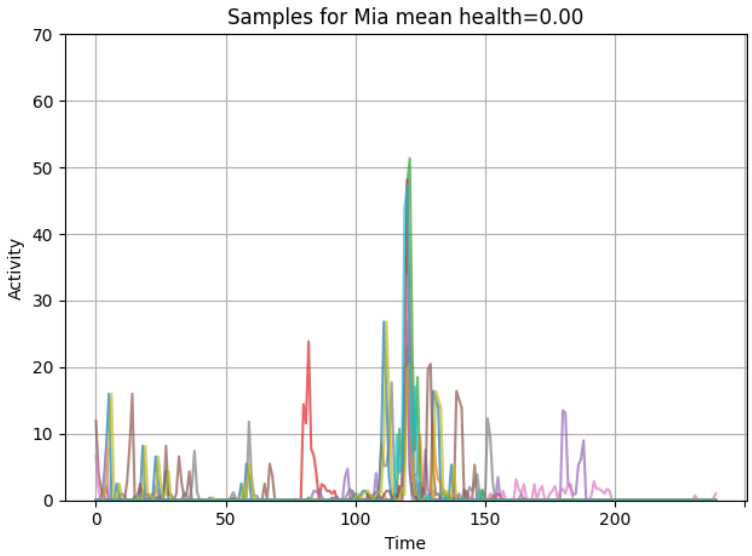


Figure 4.4: **Schematic of the ML samples for a single cat**. In this illustration, we explain how we created samples for a single cat. For each cat an activity count was recorded every second for 12 days and the cat was assigned a DJD health status (0 for healthy, 1 for unhealthy) by its owner. (A) shows the activity data of the first day. In this example, the first peak ($peak_0$) was found in Day_1 at $t = 100$, we will create a sample by selecting the activity in the window centred on $t_{100} = t_{peak_0}$ of length 2 minutes (120 seconds). By following this process we can build an array of n samples containing the activity data around the top n peaks (B).

Because ML algorithms aim to model generalisable patterns in training data and correctly predict unseen data, such techniques are sensitive to overfitting i.e when a model adjusts excessively to the training data and is unable to predict unseen data correctly. Data augmentation is a technique used in ML to artificially increase the size of a dataset by creating new, synthetic

A – 1-Peak samples for a single healthy cat.



B – 3-Peaks samples for a single healthy cat.

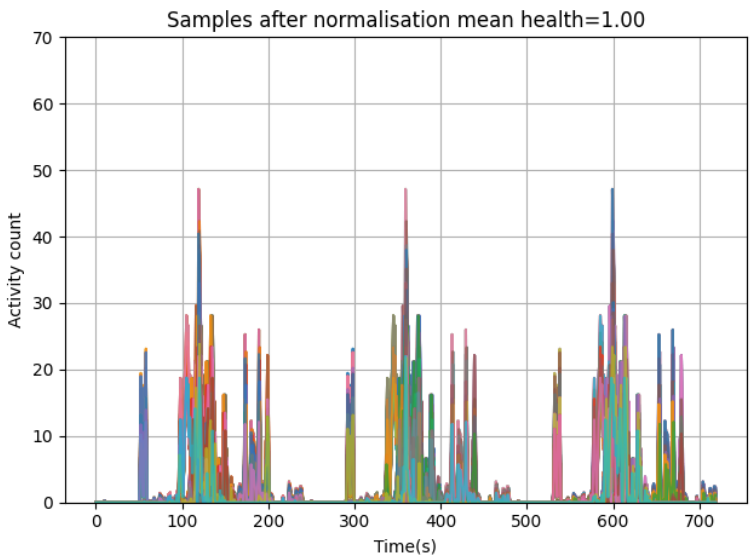


Figure 4.5: **1-Peak and 3-Peaks samples for a single healthy cat.** While (A) shows 1-Peak samples for a single healthy cat, (B) shows the augmented version of (A) by permutation of 3 peaks. The x-axis shows the time in seconds (length of the sample) the y-axis shows the activity count value.

data points that are similar to the original data points. Data augmentation is commonly used when training ML models, especially in computer vision tasks such as image classification, object detection, and semantic segmentation [87] [132]. The main idea behind data augmentation is to increase the diversity of the data available to the model, thereby improving its ability to generalise to new, unseen data. Data augmentation can be achieved by applying a variety of transformations to the original data, such as rotation, translation, scaling, flipping, cropping, and noise addition. These transformations can be applied randomly or with specific parameters, depending on the task.

Our small dataset is subject to overfitting because of the small number of samples, one solution is data augmentation that increases the number of samples from existing data [84] [142]. The initial dataset was augmented (increasing the number of samples) by building permutations of peaks from the initial sample dataset (Fig 4.6)(Fig 4.5B). We will also cap the number of possible permutations to a thousand per cat to limit computational expenses.

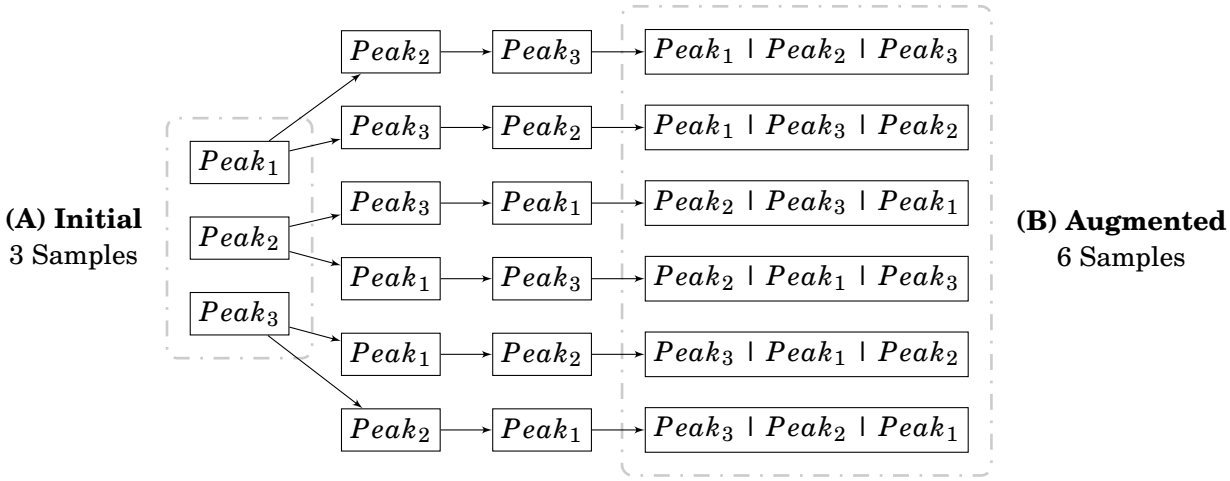


Figure 4.6: **Cat samples augmentation with permutation for a single cat**. By using permutations we can increase the number of samples from 3 samples containing 1 peak each (A) to 6 samples containing 3 peaks each (B).

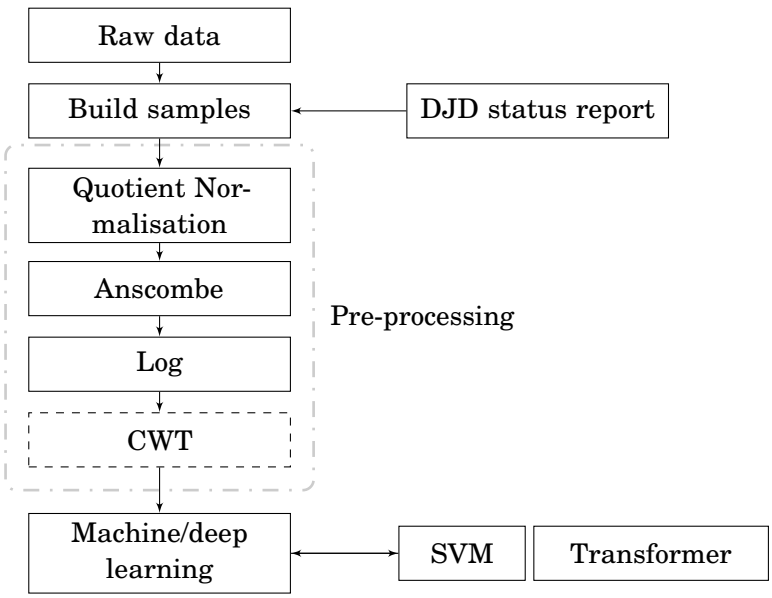


Figure 4.7: **Classification of health status pipeline**. Diagram showing the pipeline used for the classifying of healthy from unhealthy samples.

4.3.2 Pre-processing

To compare the different activity patterns within each sample we will align and normalise the data to avoid bias due to different sensor calibrations and positions of the sensor on the collar, different cats also naturally have different levels of activity depending on their breed and the architecture of the home they live in. We will use the same quotient normalisation technique (Section. 3.2.3) we used to normalise the small-ruminant samples. Similarly to the livestock research (Chapters 3 and 4), we will also use the same pre-processing approach of the activity count data obtained from the accelerometer-based sensor used to monitor the cats, Firstly we will use the Anscombe transform (Section. 2.2.2.1) followed by the Logarithm transform (Section. 2.2.2.2) (Fig 4.8).

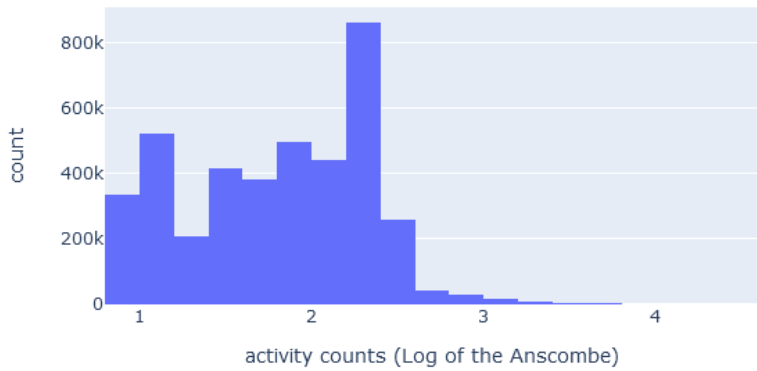


Figure 4.8: **Histogram of Log of the Anscombe of positive raw activity counts.** This histogram shows the distribution of the positive activity count data of all the cats.

4.3.3 Machine learning

In Chapter 4 section Results 3.3 we showed that the SVM algorithm was capable of accurately classifying activity counts time series and other commonly used ML algorithms gave similar to lower performances. We will train multiple SVM classifiers with the different cat "activity peaks" datasets we built. All samples were pre-processed by applying quotient normalisation and standard scaling before training the SVM.

4.3.4 Evaluation

4.3.4.1 Leave-one-out cross-validation

Data leakage is a common problem in ML where information from the test set is inadvertently used to train the model. This can result in the model being overly optimistic and performing well on the test set but poorly on new, unseen data. As our augmentation approach duplicates data across many samples from the same cat, it is important that the same peaks are not present in the training and test sets. To do this, we propose Leave-one-out cross-validation at the cat level

i.e. in turn training the model on all cats except one, and then testing on the cat left out. This is a variation of Leave-one-out cross-validation (LOOCV), a popular method used to evaluate the performance of prediction models.

The LOOCV method involves training a model on all the data except one observation, and then testing the model on the observation that was left out. This process is repeated for each observation in the dataset, and the results are aggregated to give an estimate of the model's performance. LOOCV has several advantages over other cross-validation methods. One of the main advantages is that it maximises the use of available data, as it uses all but one observation for training the model. This can be particularly useful when working with small datasets, where every data point is valuable. Another advantage of LOOCV is that it provides an unbiased estimate of the model's performance. This is because it uses all the available data to train the model, and the testing is performed on data that was not used in the training process. LOOCV also has some disadvantages. One of the main disadvantages is that it can be computationally expensive, especially when working with large datasets. This is because the training process needs to be repeated for each observation in the dataset, which can be time-consuming. Another drawback of LOOCV is its sensitivity to outliers in the dataset, as the model is trained on all available data except one observation, leading to potential performance issues on new data containing the outlier.

In our use case we will use LOOCV at the cat level i.e we will only test on the samples (peaks)(Fig.4.6) of a single cat while we will train on all of the other samples(Fig.4.9).

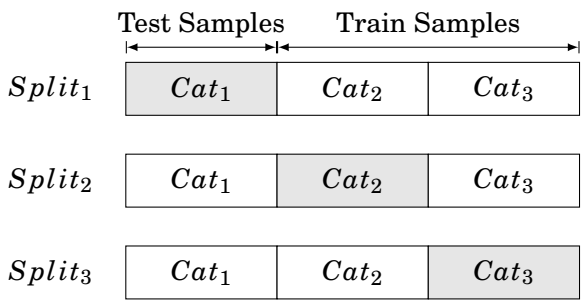


Figure 4.9: **Schematic of Leave-one-out cross-validation.**
In this example there are 3 cats in the dataset, and each cat contains multiple peak samples 4.6. Leave one out cross-validation allows training (white blocks) on the data from all the cats but 1 which is excluded for testing (grey blocks), with this approach, a data set containing the data of 3 cats can be split 3 times.

4.3.4.2 Bootstrapping for confidence intervals

We discussed in Chapter 4 section 3.2.6.2 how Receiver operating characteristic (ROC) curves are powerful tools to evaluate ML performance. For the small ruminant study, the cross-validation approach we used allowed us to show the different ROC curves per testing split (Fig. 3.22). Here because we used LOOCV, each test split only contain the samples of a healthy or unhealthy cat

which does not permit to create the confusion matrix necessary for ROC curve, we opted to create a single test ROC curve by concatenating the predictions and true health status labels (ground truth) of all the cats used for testing.

The process of bootstrapping involves randomly selecting a subset of the original dataset, with replacement, to create a new "bootstrap sample". This means that each data point in the original dataset has an equal chance of being selected multiple times in the new sample. This bootstrap samples has (approximately) the same statistical properties as the original dataset, and could feasible have been generated instead of our original dataset. This process is repeated multiple times to create several new bootstrap samples. By passing each sample through the same analysis we can then determine the different the effect random chance has on our results.

Bootstrap can be performed at any point in the pipeline but applying it prior to ML would lead to us needing to perform the whole LOOCV ML 100 times, once for each bootstrap sample. While this would have captured all uncertainty, it would have also have biased the performance of the ML downwards by effectively reducing the samples size further for what is already a small study. In this study, we therefore chose to apply it on the output of a single LOOCV ML analysis of the original dataset. We therefore generate 100 bootstrap samples on the predictions of each unique fold obtained by LOOCV, which are then sorted to determine the 95% quantiles, which correspond to the 95% confidence interval of the model's results.

4.4 Results

4.4.1 Classification of DJD status

Table 4.2 shows the bootstrap performances of the different models. Our best SVM model was trained on 5 minutes samples with permutations of 5 peaks and had a 72% median AUC with a (0.61 – 0.81) 95% confidence interval while the precision (section 3.2.6.2) was 0.67% with a (0.53 – 0.80) confidence interval, we applied QN normalisation on the samples.

Because we test our models on the samples of a unique single cat we can evaluate how accurate it is at classifying the high activity peaks for each cat - Figure(4.10). We can observe that while the model almost perfectly classified the samples from cat 603 with 98% accuracy it was not possible to classify the samples of some other cats with good accuracy.

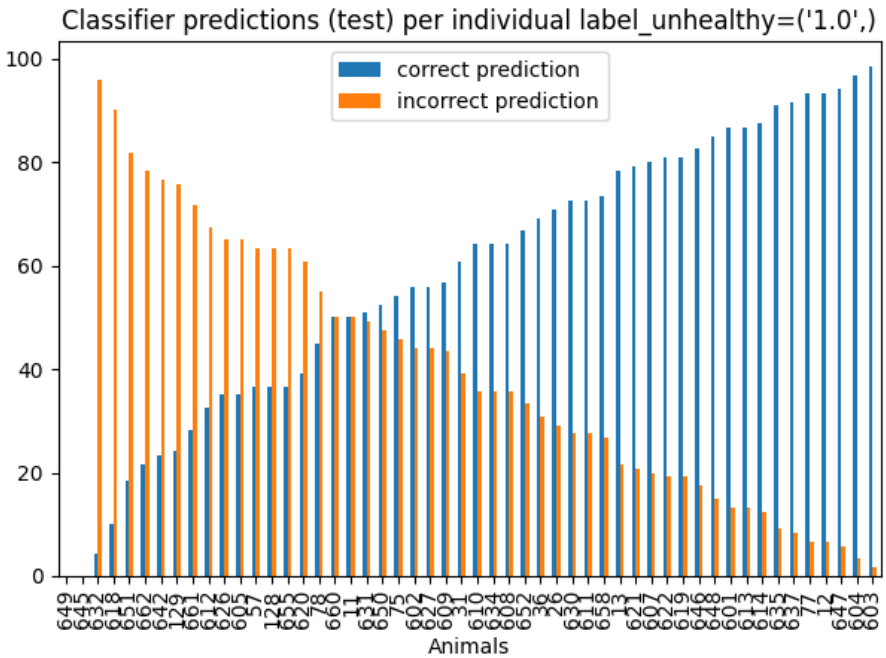
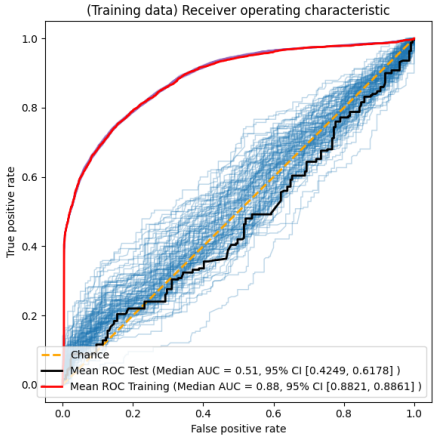


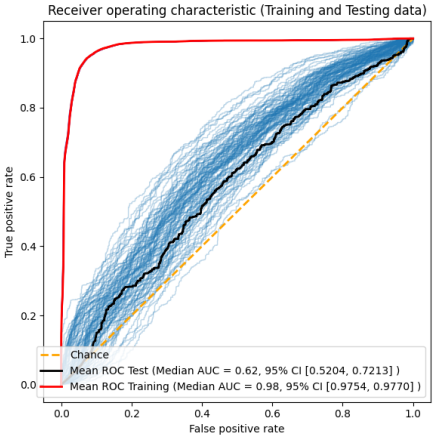
Figure 4.10: **Accuracy of classification per cat.** The x-axis shows the id of the cat tested while the y-axis shows the accuracy of the model.

In figure 4.11 we display the ROC curves for different datasets with and without data augmentation. in figure 4.11A we obtained a 51% AUC with the 1-peak dataset (no data augmentation). In figure 4.11B we used a 2-peaks permutation which increase the AUC to 62% and in figure 4.11C we obtained a 68% AUC with the 3-peaks dataset, 65% and 72% with 4 and 5 peaks respectively figure 4.11D figure 4.11E.

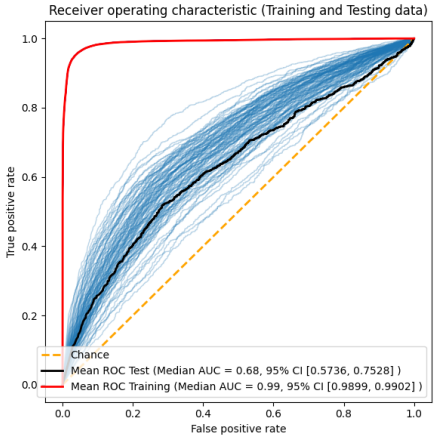
A – 1 Peak



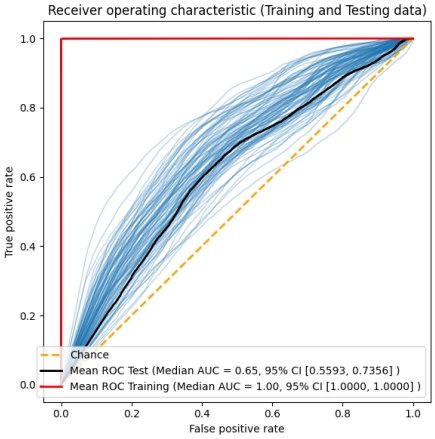
B – 2 Peaks



C – 3 Peaks



D – 4 Peaks



E – 5 Peaks

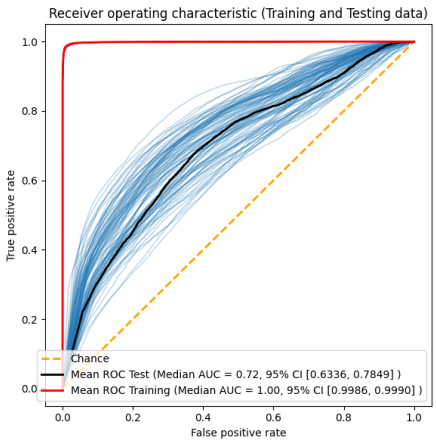


Figure 4.11: **AUC and ROC curves when increasing the number of peaks.** The red curve shows the training AUC while the black curve shows the testing AUC. The blue curves show the confidence interval.

AUC test (95% CI)*	AUC train*	Precision test*	Precision train*	N train	N test	N peaks	L(min)	Classifier	Pre-proc
0.72 (0.63-0.78)	1.00 (1.00-1.00)	0.69 (0.53-0.80)	0.99 (0.99-0.99)	51000	1000	5	5	SVM(rbf)	QN
0.69 (0.61-0.76)	1.00 (1.00-1.00)	0.65 (0.48-0.77)	0.98 (0.98-0.98)	51000	1000	4	4	SVM(rbf)	QN
0.68 (0.59-0.78)	0.99 (0.99-0.99)	0.65 (0.52-0.78)	0.96 (0.96-0.96)	36720	720	3	3	SVM(rbf)	QN
0.67 (0.57-0.76)	1.00 (1.00-1.00)	0.62 (0.44-0.79)	1.00 (1.00-1.00)	51000	1000	4	8	SVM(rbf)	QN
0.67 (0.60-0.76)	1.00 (1.00-1.00)	0.64 (0.49-0.80)	0.99 (0.99-0.99)	36720	720	3	6	SVM(rbf)	QN
0.66 (0.57-0.74)	1.00 (1.00-1.00)	0.62 (0.45-0.74)	1.00 (1.00-1.00)	36720	720	3	12	SVM(rbf)	QN
0.65 (0.54-0.75)	1.00 (1.00-1.00)	0.59 (0.44-0.75)	1.00 (1.00-1.00)	51000	1000	5	10	SVM(rbf)	QN
0.65 (0.55-0.73)	0.96 (0.96-0.96)	0.62 (0.47-0.79)	0.90 (0.90-0.90)	4590	90	2	4	SVM(rbf)	QN
0.65 (0.56-0.71)	1.00 (1.00-1.00)	0.60 (0.49-0.72)	1.00 (1.00-1.00)	51000	1000	4	4	SVM(rbf)	QN->STD
0.65 (0.53-0.75)	1.00 (1.00-1.00)	0.59 (0.44-0.74)	1.00 (1.00-1.00)	51000	1000	4	16	SVM(rbf)	QN
0.65 (0.55-0.72)	0.99 (0.99-0.99)	0.61 (0.45-0.74)	0.96 (0.96-0.96)	4590	90	2	8	SVM(rbf)	QN
0.65 (0.53-0.72)	1.00 (1.00-1.00)	0.59 (0.47-0.75)	0.99 (0.99-0.99)	51000	1000	4	4	SVM(rbf)	STD
0.64 (0.52-0.74)	1.00 (1.00-1.00)	0.60 (0.47-0.72)	1.00 (1.00-1.00)	51000	1000	5	5	SVM(rbf)	QN->STD
0.64 (0.55-0.71)	1.00 (1.00-1.00)	0.61 (0.44-0.74)	1.00 (1.00-1.00)	36720	720	3	3	SVM(rbf)	QN->STD
0.64 (0.54-0.71)	0.91 (0.91-0.91)	0.63 (0.52-0.75)	0.85 (0.85-0.85)	4588	90	2	2	SVM(rbf)	QN
0.64 (0.54-0.72)	1.00 (1.00-1.00)	0.57 (0.41-0.72)	1.00 (1.00-1.00)	51000	1000	4	8	SVM(rbf)	QN->STD
0.64 (0.54-0.72)	1.00 (1.00-1.00)	0.56 (0.44-0.72)	1.00 (1.00-1.00)	36720	720	3	6	SVM(rbf)	QN->STD
0.62 (0.53-0.70)	1.00 (1.00-1.00)	0.57 (0.44-0.73)	0.98 (0.98-0.98)	36720	720	3	3	SVM(rbf)	STD
0.62 (0.49-0.72)	1.00 (1.00-1.00)	0.55 (0.41-0.72)	1.00 (1.00-1.00)	51000	1000	5	10	SVM(rbf)	QN->STD
0.62 (0.52-0.73)	1.00 (1.00-1.00)	0.56 (0.37-0.68)	1.00 (1.00-1.00)	51000	1000	5	5	SVM(rbf)	STD

Table 4.2: **Comparison of model results for DJD peak samples classification.** "N Train" and "N test" show the number of samples used for training and testing respectively while "L" indicate the sample length in minute and "Pre-proc" shows the preprocessing used on the sample time series before training (QN: Quotient Normalisation, "STD": Standard Scaling).

In figure 4.12 we show the impact of the data augmentation on the AUC performances. Each curve shows the training or testing AUCs for different window lengths (i.e how much time before and after the peak event) and pre-processing pipelines. Using the 5-peak samples gives the optimal test AUC values with 3 and 4 peaks giving slightly slower results when using a 60 seconds sample window (30 seconds before and after the peak of activity). For all models, we can observe an increase in AUC when using more than one peak with a plateau in the AUC increase after 3 peaks.

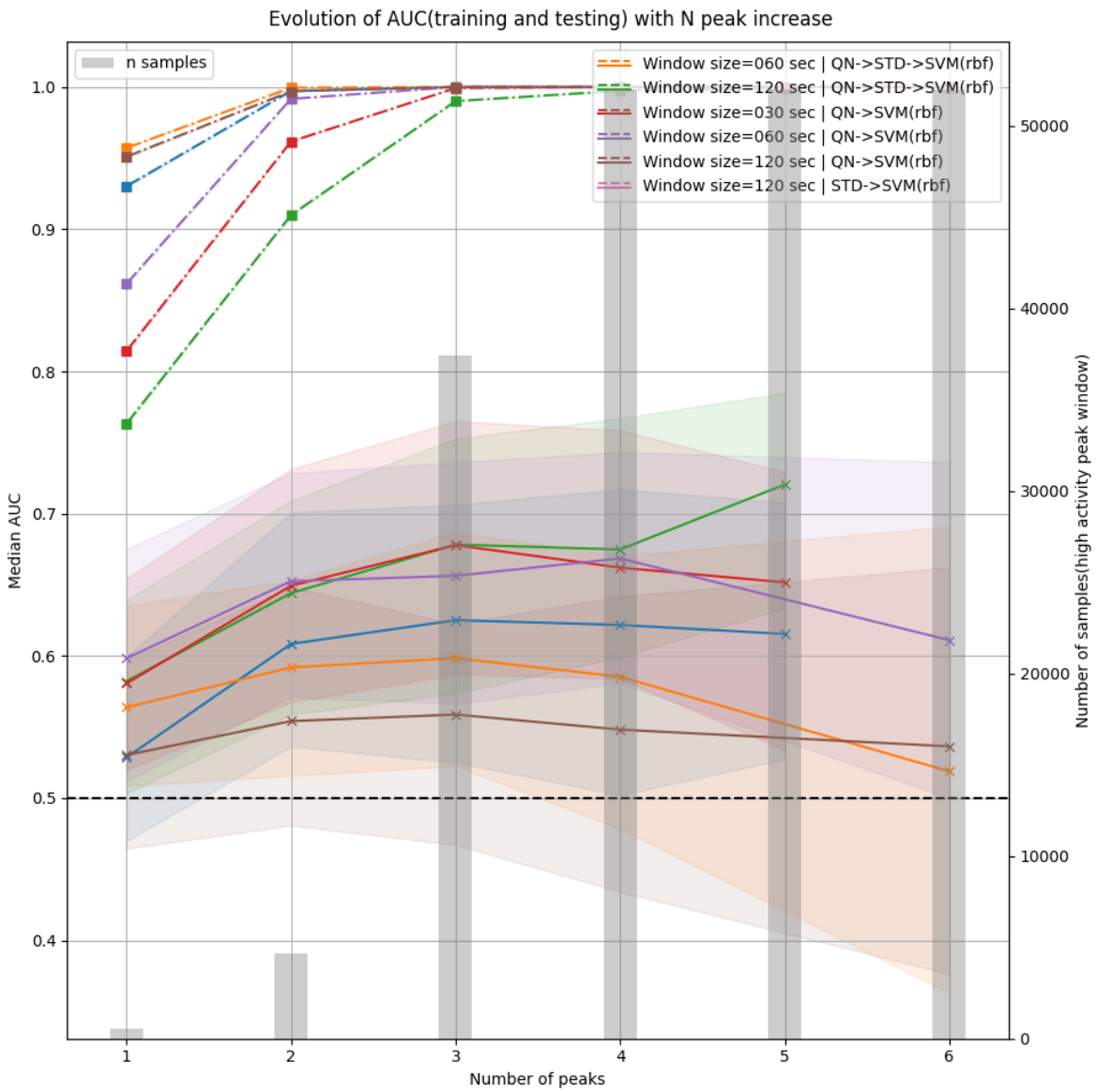
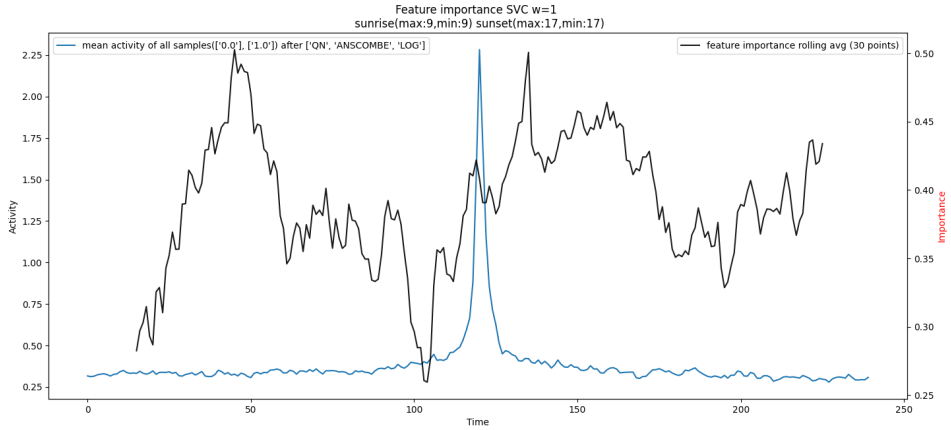


Figure 4.12: **Evolution of the AUC with the increase of the number of peaks.** While the x-axis shows the number of peaks used in the data set, the y-axis shows the Median AUC. The training AUC is shown by the dotted lines while the testing AUC is shown by the solid lines.

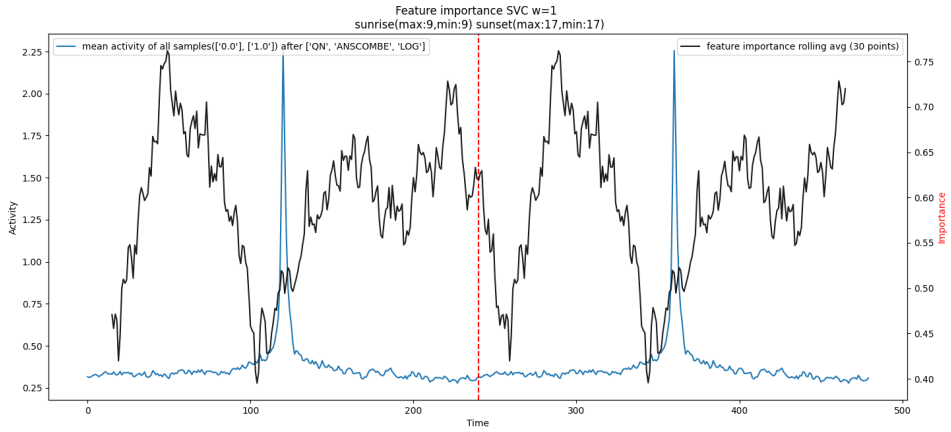
4.4.2 Model explainability

It is paramount to interpret the output of our ML algorithms as the ultimate goal for this research is on the field application. In the small ruminant study, we were able to identify that the night-time activity of the animals contained important information to classify the goats' and sheep's health status (section 3.3.7), such information is valuable for farm management as it might allow farmers to monitor their animals further in the night or might help to devise a bio-logger that would put more importance in collecting high-resolution data in the night time while tacking less measurement in the day time hence saving battery life. For this study, interpretability is key for veterinarians as it would allow them to devise appropriate solutions and treatments to alleviate symptoms and increase cats' well-being. After fitting our ML model it is possible to visualise what parts of the data in the samples are most important for the classification process (Fig 4.13). Because we fitted the model with accelerometry time series we can directly see what part of the time series is most useful for the model to discriminate between healthy and unhealthy peaks. The results show that the activity before the highest peak appears to be the most important for classification. This might indicate that cat apprehension is causing a change in activity right before engaging in a high-activity move. Although in Figure 4.13 we used the non-augmented dataset we could observe a sharp decrease of feature importance approximately 8 seconds before the high activity event which was also observable in the 2, 3, 4, 5 and 6 peak versions of the dataset this pattern may be due to cats stopping in preparation of a strenuous effort, because they remain stationary, the model may perceive this portion of the data as less significant.

A – 1 Peak



B – 2 Peaks



C – 3 Peaks

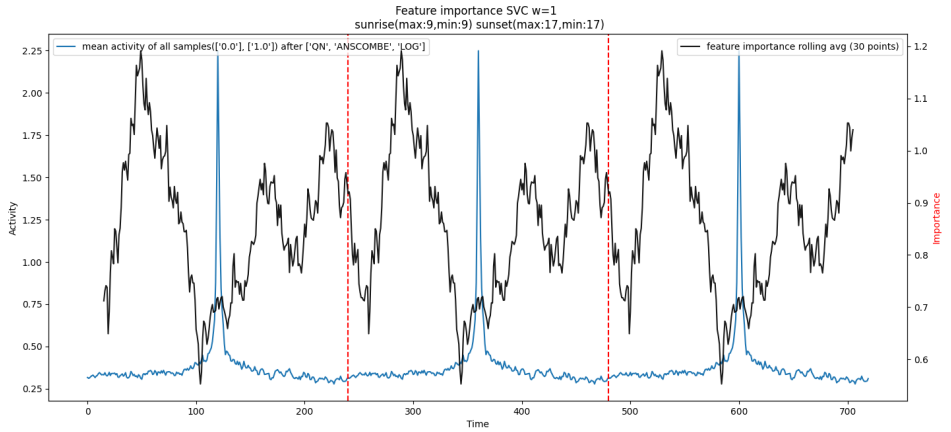
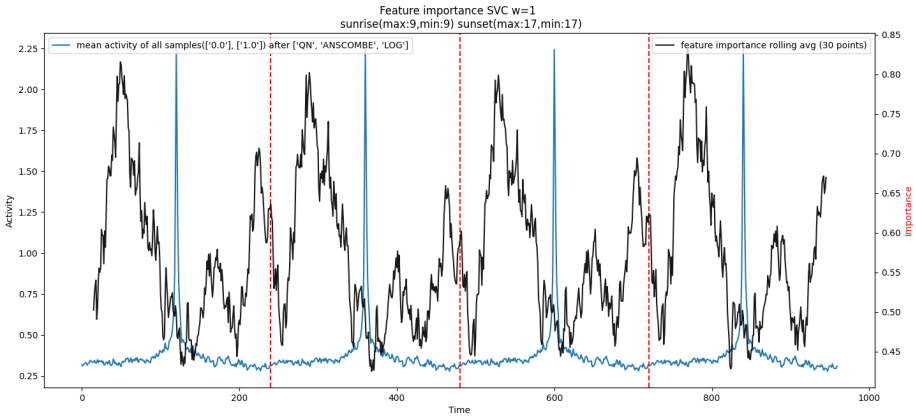
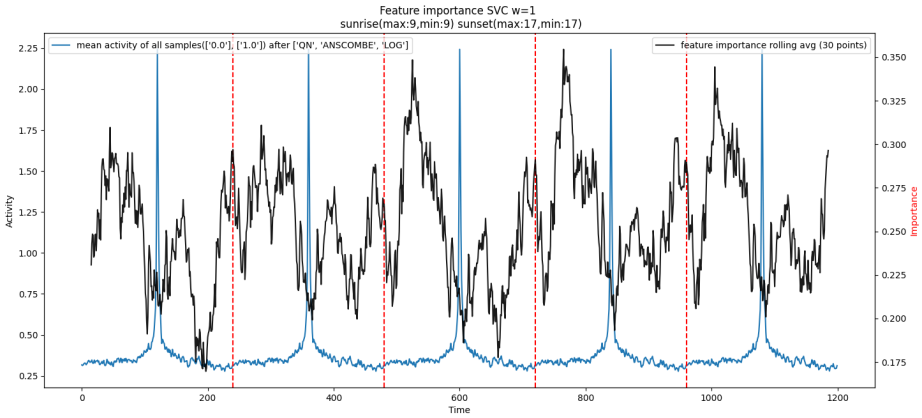


Figure 4.13

D – 4 Peaks



E – 5 Peaks



F – 6 Peaks

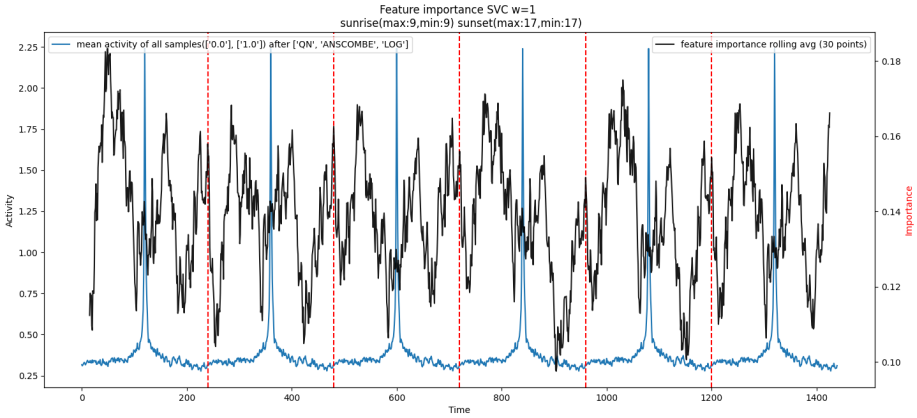


Figure 4.13: **Feature importance.** The Blue curve shows the mean trace of all the samples (3 peaks) used to fit the SVM model. The Black curve displays the feature importance derived from the fitted model. While the x-axis shows the time, the vertical red dotted lines separate the different peaks of activity within the samples, the y-axis on the left shows the mean activity count of all cats after normalisation and pre-processing, the y-axis on the right shows the SVM feature importance.

4.5 Conclusion

Despite the limited amount of data available, our machine-learning approach demonstrated promising results in the prediction of early joint disease in cats with the help of our data augmentation approach. It has also been found that the data within small windows centred around bursts of activity contains relevant information to discriminate a healthy cat from an unhealthy cat. However, the prediction accuracy varied for each cat; cats have behaviour unique to their own genetics and personality, classifying the data of unseen cats with a modest-sized training data set is a challenging task, further work may allow better predictive power. Further work is currently ongoing to determine which part of the windows of activity is most important for prediction, this will give novel insight into how cats with early joint disease differ in their movements when trying to perform actions of high activity.

Although our approach allowed us to reveal the presence of discriminative markers within the accelerometry data of the cats, this was achieved by looking at specific zones of high-intensity activity of the cats which is not generalisable to the entire data trace of a cat. We were able to build a peak classifier but to develop a cat health classifier, we would need to use the results obtained as part of the training of another model. Furthermore, while data augmentation is a common alternative to a large dataset, large data collection remains key for a generalisable ML model. Early results are a good incentive to extend or renew this study with further participants.

CHAPTER 5

CONCLUSIONS

In this thesis, we have presented a ML pipeline capable to classify the activity data of small ruminants measured by affordable wearable accelerometers despite a relatively small size data set and the challenging quality of the data. We also showed promising progress toward the detection and classification of degenerative joint disease in domestic cats.

5.1 Exploratory data analysis

Exploratory data analysis (EDA) is a critical first step in data analysis that involves exploring and summarising data to gain a better understanding of its characteristics and relationships. EDA typically involves techniques such as visualising data using graphs and charts, computing summary statistics, identifying outliers and missing values, and testing assumptions about the data. The goal of EDA is to generate hypotheses and insights that can guide further analysis and modelling of the data. We have extensively analysed the raw data provided to understand its key characteristics and devise an appropriate imputation and pre-processing pipeline. The activity monitoring of small-ruminant in resource-poor communities required the use of affordable sensors for financial viability, however, our analysis showed the difficulties associated with the hardware used namely inconvenient data access with no redundancy and telecommunication faults of the transponder used which cause a high amount of missingness in the data. We tackled this issue by using advanced deep-learning-based imputation techniques which were capable to model activity data patterns from the existing data and estimate with minimal bias the missing points in our dataset, which in turn allowed us to maximise the amount of usable data for our ML pipeline.

5.2 Machine learning pipeline

Our analyses reveal that an increase in FAMACHA score from 1 ("optimal") to 2 ("acceptable") and remaining at 2 for up 1 week for Sheep and 2 weeks for goats, which is considered a sub-clinical disease, can be predicted from behaviour measured using low-cost biologgers. We discovered that the discriminative ability of our classifier increases during the night for the sheep and in inconclusive for the goats.

While efforts in developed countries have been focused on building high-precision, securely mounted and precisely fitted sensors, we demonstrate that robust results can be gained from much simpler, low-cost systems with rudimentary maintenance requirements suitable for both commercial and RP farmers in developing countries. It is important to note that due to significant calibration and mounting variation between transponders, including loosening of the transponder over time, it was necessary to perform normalisation of each activity trace to the herd/flock mean. This meant that uniform reductions in activity level from week to week are likely to be normalised out of our data. Nevertheless, a biological reason for a completely uniform reduction in activity level is implausible; instead, the intensities of some daily activities are likely to be impacted more than others. We have shown that changes in the variation of activity levels constitute a strong predictor of early changes in health status, as regards haemonchosis and are robust to technical variation.

In the small-ruminant study, we have focused on accelerometry data, but exogenous factors such as temperature, rainfall, body weight, and yield farming data could all be beneficial to the prediction, particularly as *H. contortus* is well known to hatch after humid, hot weather, and to require rainfall for movement onto pasture [151, 153]. Nevertheless, optimal incorporation of these data types is challenging because of their potential non-linear and/or lagged or cumulative effect on health. Conversely, the fundamental advantage of high-dimensional longitudinal data from accelerometers is that the end effect of these covariates could intrinsically be contained within the data directly, which ML approaches have the potential to deconvolute. The ability of ML to detect health-relevant changes in behaviour under variable climatic conditions could make it especially useful as climate change drives increasingly unpredictable transmission patterns among helminths [123], and hence to support adaptation to climate change by resource-poor farmers [140].

For the feline study, our approach allowed us to reveal the presence of discriminative markers within the accelerometry data of the cats, this was achieved by looking at specific zones of high-intensity activity of the cats which is not generalisable to the entire data-trace of a cat. We were able to build a peak classifier but to develop a cat health classifier, we would need to use the results obtained as part of the training of another model. Furthermore, while data augmentation is a common alternative to a large dataset, large data collection remains key for a generalisable ML model. Early results are a good incentive to extend or renew this study with further participants.

5.3 Translation into practice

Helminths negatively impact livestock productivity worldwide, and in resource poor settings are considered 'neglected cold spot' diseases, in that they are preventable in principle, but farmers continue to struggle to manage their effects [122]. Technical improvements in helminth control consequently have especially high potential to positively impact farmer livelihoods, with knock-on benefits for human nutrition and health [112]. The FAMACHA system has been successfully adopted by smallholder farmers in Africa, but sustained use is difficult because of high training and labour requirements [152]. Our results show that it is feasible to apply ML approaches to data streams that are attainable on smallholder farms in Africa to detect early changes in health status and support timely and targeted intervention.

Notably, in this study, we have focused on FAMACHA evaluation of *H. contortus* infection; whether the multi-label classification of a range of different disease states and transient events is possible is currently unknown but would require extensive studies to attain the appropriate predictive power. In addition, as our accelerometry data is based on a simple activity count paradigm, we hypothesise that activity traces could be derived from other sensor types, such as video [91] [6], for direct input into our prediction model.

The existing pipeline presents numerous impactful practical applications. The developed ML pipeline, tailored for animal health monitoring using accelerometer data, finds effective implementation in real-world scenarios. The focus on helminth control, particularly the successful application of the FAMACHA system by smallholder farmers in Africa, underscores the immediate relevance of ML approaches. Farmers can leverage this technology to identify early changes in the health status of their livestock, facilitating timely and targeted interventions. Practically, the current ML pipeline can seamlessly integrate with various types of sensors, such as collars, leg bands, or ear tags, depending on the species and farming practices. This flexibility allows for customization based on specific needs and conditions. While the ML algorithm would require an initial training period for training data collection on farm, , it could be deployed for real-time classification of new data once trained. The ensuing classification results can trigger alerts for farmers or veterinarians, indicating potential health problems and prompting prompt actions.

Collecting robust annotated datasets is especially challenging in resource-poor farming systems where farming practices are generally less consistent, regulated and well-funded. Because of this and the need for intensive manual labour over a prolonged period, our datasets are highly valuable. Although the training data obtained is dependent on farm topology, location and management, we have shown that a basic ML pipeline can discriminate on behavioural cues dominated by fluctuations in activity levels. Some concept drift was found, particularly among animals with increasing parasitic burden.

Another challenge is the need for careful monitoring of the algorithm's performance in the real-world environment. Because the algorithm is continuously learning and adapting, there is a risk that it may develop sub-optimal strategies or make incorrect decisions that could harm

animal health or welfare. It is therefore important to have robust monitoring and feedback systems in place to ensure that the algorithm is performing as intended and making decisions that align with the goals of the livestock operation.

In practice, streaming the accelerometer data of livestock in real-time would be most useful for farmers and animal caregivers. Data streaming involves collecting data from wearable sensors in real-time, processing the data, and applying ML algorithms to classify the data into different health states. The accelerometry data can be collected using various sensors, such as collars, leg bands, or ear tags and the data at an appropriate frequency which would allow for maximising battery life while keeping high-enough resolution information, our small-ruminant study revealed that a recording frequency of 1 measurement per minute is enough, while the cats' data was using 1 measurement per second, lower resolution may be sufficient. We could hypothesise that because indoor cats do not exist in herds a higher resolution might be preferable, unlike the livestock animals there is no intrinsic herd information that could help ML algorithm. Further work is required to assess the optimal resolution in this case. The first step in real-time health monitoring is to preprocess the accelerometer data. This involves cleaning the data, removing noise, and extracting relevant features from the data. The preprocessing step can be challenging, as the data is constantly changing and can contain artefacts from sensor movement or environmental factors. Once the data has been preprocessed, ML algorithms can be used to classify the data into different health states. The ML algorithm needs to be trained on a labelled dataset of accelerometer data, where each data point is labelled with the corresponding health state, such as healthy, sick, or injured. The choice of ML algorithm will depend on the specific problem being addressed, but common algorithms include decision trees, SVM, and neural networks. The ML algorithm can be trained on historical data and then deployed to classify new data in real-time. The classification results can be used to alert farmers or veterinarians of potential health problems, enabling early intervention and treatment. For example, if the ML algorithm detects a change in activity levels or behaviour that suggests an animal is in distress, an alert can be sent to the farmer or veterinarian to investigate further.

Real-time health monitoring of livestock using streaming accelerometer data and ML has several benefits. It can enable early detection of health problems, which can improve animal welfare and reduce economic losses. It can also reduce the need for manual health monitoring, which can be time-consuming and labour-intensive. Additionally, real-time health monitoring can provide farmers and veterinarians with valuable insights into animal behaviour and activity levels, which can inform management practices and improve overall animal health and well-being.

In conclusion, reinforcement learning techniques offer a promising approach to improving livestock management and decision-making based on real-time data. While there are challenges associated with this approach, with careful planning and monitoring, reinforcement learning can help optimise the performance of livestock operations and improve animal health and welfare [41] [106].

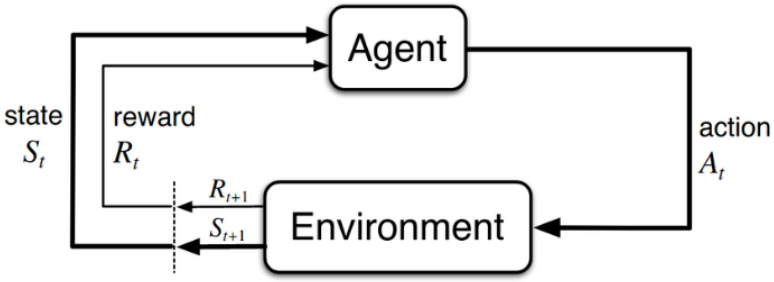


Figure 5.1: **Schematic of online reinforcement learning.** Contrary to traditional ML, reinforcement learning allows the retraining of the model(agent) in real-time. Image taken from source: Sutton and Barto, 2018 [137]

5.4 Summary

We have shown that activity count data collected with wearable accelerometers is a predictor of animal health for goats, sheep and cats. For small ruminants, we used activity data mapped to the level of a parasitic infection of the animals while for the cat we used degenerative joint disease as the health metric both had impacts on animal behaviour and activity that we could detect with ML algorithms. The small-ruminant study was particularly challenging as the association of accelerometry data with worm infection is novel work and there was no control of data quality, nonetheless, we leveraged the use of powerful imputation techniques which allowed us to use the available data to the fullest.

5.5 Future work

Animal health monitoring is a crucial area of research, as it is essential to ensure the well-being and productivity of livestock. ML and accelerometers have emerged as promising tools in this domain, with the potential to offer real-time, non-invasive, and cost-effective monitoring of animal behaviour and health.

Accelerometers are small, low-cost sensors that can be attached to animals to measure their movement and activity patterns. ML algorithms can be trained on these data to identify patterns of behaviour that indicate changes in health, such as reduced activity or changes in posture. In recent years, researchers have explored a range of ML approaches for animal health monitoring, including artificial neural networks, decision trees, and SVM, however, most of the research tries to infer animal health based on the analysis of the occurrence of common behaviours such as grazing, running, sleeping etc... We showed in this thesis that by using the accelerometry data of consecutive days it is possible to detect health markers in the data of goat and sheep without the need to categorise different behaviour. Similarly, although novel the recent work on dog and cats

focus more on the categorisation of behaviour related to poor health while our approach classifies segment of activity themselves.

One area of future research in animal health monitoring with ML and accelerometers is the development of more sophisticated algorithms that can detect subtle changes in behaviour that may indicate the onset of illness or injury. This may involve the use of deep learning techniques such as convolutional neural networks, which are capable of analysing large amounts of data and identifying complex patterns and the recent increasingly popular Transformer, first introduced in 2017. Transformer deep learning models have revolutionised natural language processing (NLP) tasks. These models are based on the transformer architecture, which was introduced by Vaswani et al [146], its based on a neural network architecture that uses self-attention mechanisms to compute contextual relationships between words in a sentence or sequence. The transformer architecture has several advantages over traditional recurrent neural network (RNN) architectures. One of the main advantages is that the transformer architecture can process entire sequences in parallel, whereas RNNs process sequences sequentially. This parallel processing leads to faster training and inference times, making transformers more efficient than RNNs. Another advantage of transformers is their ability to capture long-range dependencies between words in a sequence. Traditional NLP models struggle with this because they rely on sequential processing, making it difficult to capture long-range dependencies. Transformers overcome this limitation by using self-attention mechanisms, which enable them to capture contextual relationships between words across the entire sequence. Transformers have been used in a variety of NLP tasks, including machine translation, text classification, question answering, and language modelling. They have achieved state-of-the-art performance in many of these tasks, surpassing previous models that relied on RNN architectures. One of the most popular transformer models is the BERT (Bidirectional Encoder Representations from Transformers) model [31], which was introduced by Google in 2018. BERT is a pre-trained language model that uses a transformer architecture and is trained on large amounts of text data. It has been used for a wide range of NLP tasks and has achieved impressive results. Transformers can also be used for text-to-image prediction [120]. Transformers can also be used for binary classification of activity time series and have shown promising results [35].

Two additional broad avenues for future research may include: (a) Starting from deployments of our pre-trained model, use of online reinforcement learning techniques to create a 'life-long learning' decision support system which identifies animals for FAMACHA evaluation and feeds back the results to dynamically update the model calibration and improve future predictions; (b) Multivariate time-course statistical modelling to further characterise the nature of sheep and goat behaviour in health and disease.

Reinforcement learning is a branch of ML that involves training an agent to make decisions in a given environment based on trial and error. In the context of livestock data, reinforcement learning can be used to develop intelligent systems that can make decisions related to animal

health, nutrition, and management based on real-time data, for example, we can imagine a system where farmers could annotate specific events or self-assessed new conditions via the use of a simple website or smartphone application at any time. Online reinforcement learning, in particular, is a technique that allows the models to learn and adapt in real-time as new data becomes available.

Online reinforcement learning for livestock data involves several key steps. First, a reinforcement learning algorithm is developed and trained using historical data to predict the optimal action to take in a given situation. This could involve, for example, predicting the optimal feed ratio for a particular animal based on its weight, activity level, and other factors. Once the algorithm is trained, it is deployed in the real-world environment, and the model interacts with the environment by classifying new data and receiving feedback. The model then uses this feedback to update its decision-making strategy in real-time, continuously improving its performance based on the latest data. There are several advantages to using online reinforcement learning for livestock data. One of the main benefits is that it allows for more responsive and adaptive decision-making based on the latest data. For example, if an animal's health status changes suddenly, the reinforcement learning algorithm can quickly adjust its recommendations for feeding, medication, or other interventions based on this new information (Fig. 5.1).

Another advantage of online reinforcement learning is that it can help optimise the performance of livestock operations over time. By continuously learning and adapting to new data, the algorithm can identify patterns and trends that may not be apparent to human operators, leading to more efficient and effective management strategies.

However, there are also several challenges to using online reinforcement learning for livestock data. One of the main challenges is the need for high-quality data to train the algorithm. This may require significant investment in data collection and processing systems, as well as careful consideration of data quality and consistency.

Another important area of research is the integration of other sensor data, such as heart rate, respiration rate, and temperature, into the analysis. This will enable more comprehensive monitoring of animal health and provide more accurate insights into the underlying causes of changes in behaviour. Furthermore, the application of ML and accelerometers to animal health monitoring can be extended beyond livestock to other species, such as companion animals and wildlife. For example, researchers can develop models to monitor the activity patterns of dogs and cats to identify changes in behaviour that may indicate health issues. Similarly, wildlife researchers can use accelerometers to monitor the movements of animals in their natural habitats and gain insights into their behaviour and ecology. In addition the development of more practical and cost-effective monitoring systems that can be deployed on a large scale. This will involve the integration of ML algorithms into low-cost, battery-powered devices that can be attached to animals and transmit data wirelessly to a central server for analysis.

Unsupervised ML could also be a viable direction to take for further research, it is a type

of ML that involves training an algorithm to identify patterns and structures in data without the need for labelled training examples. For livestock monitoring, unsupervised ML can be used to analyse large datasets of animal behaviour and cluster different patterns that may indicate changes in health. Clustering algorithms are used to group animals together based on similarities in their behaviour or other attributes. This can help identify subgroups of animals that may have different health or welfare needs, or that may respond differently to management interventions. Another example of unsupervised ML for livestock monitoring is anomaly detection. Anomaly detection algorithms are used to identify outliers or unusual data points that may indicate a problem or abnormality. For example, an anomaly detection algorithm could be used to identify a cow that is exhibiting unusual activity patterns or that has stopped eating, which could indicate a health problem. Principal component analysis (PCA) [1] is a commonly used unsupervised ML technique that can be used for livestock monitoring [117] [95] [42]. PCA involves identifying the most important features or variables in a dataset and reducing the dimensionality of the data by combining these features into a smaller number of principal components, for example in the figure. 5.2 we used goat activity samples to find relevant clusters, however, it was not possible to find significant differences with this approach, further work is required. This can help identify patterns or relationships in the data that may not be apparent in the original dataset. One advantage of unsupervised ML for livestock monitoring is that it does not require labelled training data, which can be difficult or expensive to obtain. This can make it easier to analyse large datasets of animal behaviour and identify patterns that may be relevant to animal health or welfare. However, there are also challenges associated with unsupervised ML for livestock monitoring. One challenge is the need to carefully interpret the results of the analysis. Because unsupervised ML algorithms do not rely on labelled training data, it can be difficult to know whether the patterns or clusters identified by the algorithm are meaningful or useful for making decisions about animal management. Another challenge is the need to ensure that the data being analysed is accurate and representative of the population of interest. For example, if the data is collected using sensors that are only attached to a subset of animals, the clustering or anomaly detection algorithms may not be representative of the entire herd.

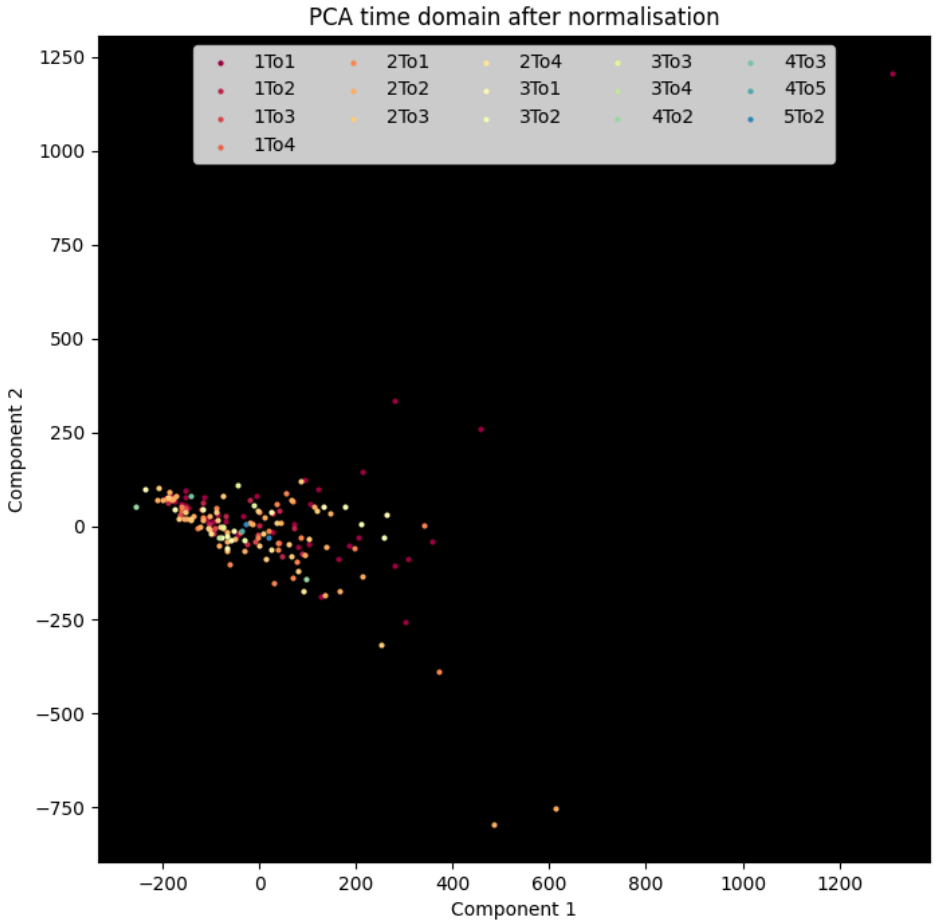


Figure 5.2: **Principal component analysis (PCA) of goat activity samples.** While the x-axis shows the first PCA component, the y-axis shows the second. Each point shows a dimensionality-reduced (2 dimensions) representation of the different samples.

In conclusion, animal health monitoring with ML and accelerometers is an exciting and rapidly evolving field that has the potential to revolutionise the way we monitor and manage animal health. With further research and development, this technology could be applied to a wide range of animal species and have a significant impact on animal welfare, productivity, and conservation.

5.6 Conclusion

This thesis aims to advance the field of animal health monitoring by employing ML techniques on accelerometer data. The primary research questions revolve around the effectiveness of the ML pipeline in classifying the activity data of small ruminants, despite challenges such as a limited dataset and data quality issues. Another key objective is the exploration of ML applications in detecting DJD in domestic cats, with an emphasis on predicting early signs of health issues. The research aims to shed light on the impact of health conditions, specifically parasitic infections in small ruminants and DJD in cats, on animal behavior as detected through ML algorithms. The objectives include showcasing the viability of using affordable sensors in resource-poor farming settings and considering the integration of exogenous factors like temperature, rainfall, for more comprehensive predictions. Future research directions involve the development of advanced algorithms, potentially incorporating deep learning techniques like transformers, to identify subtle changes in behavior indicative of early illness or injury. Additionally, the study explores extending ML and accelerometry applications to diverse species, including companion animals and wildlife, highlighting potential avenues for further exploration and addressing the challenges of real-time monitoring. The overarching goal is to contribute valuable insights to the evolving field of animal health monitoring, driven by ML and accelerometry, with broad applications in various farming practices and species.

APPENDIX A

APPENDIX A

A.1 Software and Packages

The analysis and development in this thesis were conducted using Python [118] 3.10 along with the following packages:

- `typer==0.4.1`
- `pandas==1.1.5`
- `scikit-learn==1.0.2`
- `matplotlib==3.5.1`
- `plotly==5.6.0`
- `umap-learn==0.5.2`
- `plotnine==0.8.0`
- `pycwt==0.3.0a22`
- `PyWavelets==1.3.0`
- `BaselineRemoval==0.0.7`
- `datashader==0.13.0`
- `scikit-image==0.19.2`
- `colorcet==3.0.0`
- `seaborn==0.11.2`
- `nlopt==2.7.1`
- `natsort==8.1.0`
- `tensorflow==2.8.0`
- `holoviews==1.14.8`
- `pydot==1.4.2`
- `pydotplus==2.0.2`
- `graphviz==0.20`
- `colour`

These packages were instrumental in conducting various aspects of the research, including data analysis, machine learning, and visualisation.

A.2 Regularisation

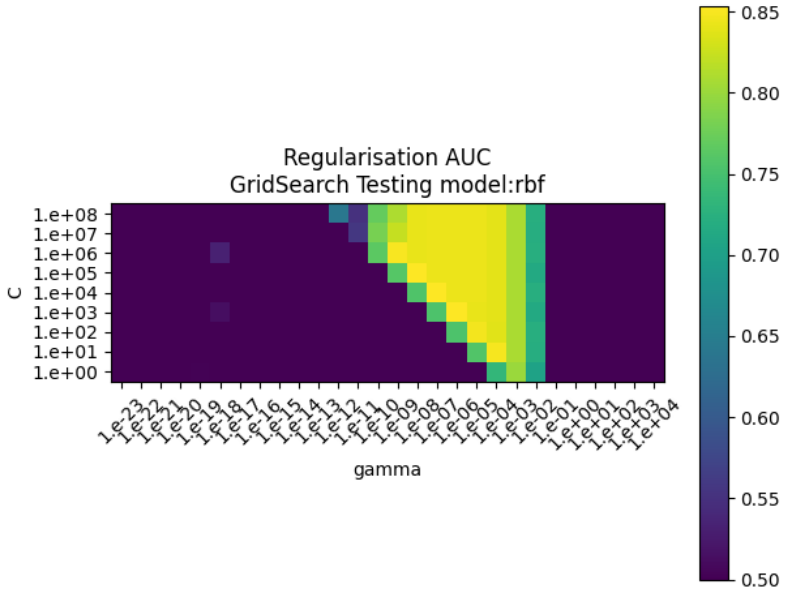
A.2.1 Method

The technique utilises a parameter grid search method to refine hyperparameters for a Support Vector Machine (SVM) with a radial basis function (RBF) kernel. Two pivotal hyperparameters, "C" (regularisation parameter) and "gamma" (RBF kernel coefficient), undergo a systematic exploration across specified candidate values. The range for "C" encompasses a series of powers of 10, ranging from 1 to 100,000,000, while "gamma" spans powers of 10 from -25 to 3. The parameter grid search exhaustively generates all conceivable combinations of these hyperparameter values utilising the ParameterGrid class from the scikit-learn library. Each combination constitutes a unique set of hyperparameters that is subsequently applied to train and evaluate the SVM RBF model. This thorough exploration enables the identification of the optimal hyperparameter configuration by evaluating the model's performance across the entire parameter grid. The technique facilitates a comprehensive search for the hyperparameter values that yield the best model performance on a given dataset.

A.2.2 Result

Figure A.1A shows that on the sheep farm higher C values > $1e^{06}$ allows for a wider selection of gamma that will result in AUCs ≥ 80% while figure A.1B shows that there are fewer combination of C and gamma that result in optimal AUC value where $C = 1e^{07}$ and $gamma = 1e^{-10}$ give the best result.

A –Sheep



B –Goats

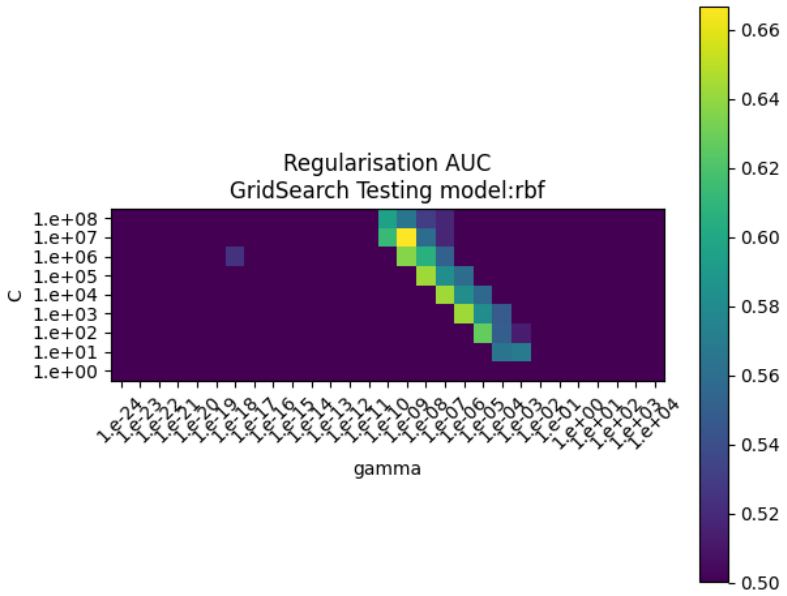


Figure A.1: **Improving Generalisation of SVM(RBF) with parameters gridsearch.** Tuning C and Gamma allows for fine-tuning the SVM model to achieve better generalization performance on unseen data. Regularization helps the model adapt to the underlying patterns in the data without being overly influenced by noise or outliers, making it more robust and reliable.

BIBLIOGRAPHY

[1] H. ABDI AND L. J. WILLIAMS, *Principal component analysis*, Wiley interdisciplinary reviews: computational statistics, 2 (2010), pp. 433–459.

[2] M. N. AHMADI, K. A. PFEIFFER, AND S. G. TROST, *Physical activity classification in youth using raw accelerometer data from the hip*, Measurement in Physical Education and Exercise Science, 24 (2020), pp. 129–136.

[3] V. K. B. ALMAZAN, F. I. B. MAHIPUS, J. R. M. SANTOS, AND M. J. C. SAMONTE, *Cahm: companion animal health monitoring system*, in Proceedings of the 2020 11th International Conference on E-Education, E-Business, E-Management, and E-Learning, 2020, pp. 417–421.

[4] F. ALVARENGA, I. BORGES, V. ODDY, AND R. DOBOS, *Discrimination of biting and chewing behaviour in sheep using a tri-axial accelerometer*, Computers and Electronics in Agriculture, 168 (2020), p. 105051.

[5] M. ANDREJAŠIC, *Mems accelerometers*, in University of Ljubljana. Faculty for mathematics and physics, Department of physics, Seminar, vol. 49, 2008.

[6] W. ANDREW, J. GAO, S. MULLAN, N. CAMPBELL, A. W. DOWSEY, AND T. BURGHARDT, *Visual identification of individual holstein-friesian cattle via deep metric learning*, Computers and Electronics in Agriculture, 185 (2021), p. 106133.

[7] F. J. ANSCOMBE, *The transformation of poisson, binomial and negative-binomial data*, Biometrika, 35 (1948), pp. 246–254.

[8] E. AYAZ, A. OZTURK, AND S. SEKER, *Continuous wavelet transform for bearing damage detection in electric motors*, in MELECON 2006 - 2006 IEEE Mediterranean Electrotechnical Conference, 2006, pp. 1130–1133.

[9] N. D. BABAYANI, *Novel approaches to an automated decision support system for on-farm management of internal parasites of small ruminants.*, Doctoral dissertation, University of Pretoria, South Africa, (2016).

[10] J. BENITO, B. HANSEN, V. DEPUY, G. DAVIDSON, A. THOMSON, W. SIMPSON, S. ROE, E. HARDIE, AND B. LASCELLES, *Feline musculoskeletal pain index: responsiveness and testing of criterion validity*, Journal of Veterinary Internal Medicine, 27 (2013), pp. 474–482.

[11] B. BESIER AND J. DUNSMORE, *The ecology of haemonchus contortus in a winter rainfall climate in australia: the survival of infective larvae on pasture*, Veterinary parasitology, 45 (1993), pp. 293–306.

[12] R. BESIER, L. KAHN, N. SARGISON, AND J. VAN WYK, *Diagnosis, treatment and management of* Haemonchus contortus *in small ruminants*, Advances in Parasitology, 93 (2016), pp. 181–238.

[13] R. BESIER, L. KAHN, N. SARGISON, AND J. A. VAN WYK, *The pathophysiology, ecology and epidemiology of haemonchus contortus infection in small ruminants*, Advances in parasitology, 93 (2016), pp. 95–143.

[14] A. V. BIANCO, S. ABOOD, A. MUTSAERS, J. P. WOODS, J. B. COE, AND A. VERBRUGGHE, *Unconventional diets and nutritional supplements are more common in dogs with cancer compared to healthy dogs: An online global survey of 345 dog owners*, Veterinary and Comparative Oncology, 18 (2020), pp. 706–717.

[15] B. E. BOSER, I. M. GUYON, AND V. N. VAPNIK, *A training algorithm for optimal margin classifiers*, (1992), p. 144–152.

[16] W. BOTEJU, H. HERATH, M. PEIRIS, A. WATHSALA, P. SAMARASINGHE, AND L. WEERASINGHE, *Deep learning based dog behavioural monitoring system*, in 2020 3rd International Conference on Intelligent Sustainable Systems (ICISS), IEEE, 2020, pp. 82–87.

[17] A. BOURKE, J. O'BRIEN, AND G. LYONS, *Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm*, Gait Posture, 26 (2007), pp. 194–199.

[18] R. N. BRACEWELL AND R. N. BRACEWELL, *The Fourier transform and its applications*, vol. 31999, McGraw-Hill New York, 1986.

[19] R. CADDIELL, M. FOSTER, AND C. DAIGLE, *A scoping review: The impact of housing systems and environmental features on beef cattle welfare*, Animals, 10 (2020), p. 565.

[20] T. CHAI AND R. R. DRAXLER, *Root mean square error (rmse) or mean absolute error (mae)*, Geoscientific Model Development Discussions, 7 (2014), pp. 1525–1534.

[21] Y.-W. CHANG AND C.-J. LIN, *Feature ranking using linear svm*, in Causation and prediction challenge, PMLR, 2008, pp. 53–64.

[22] N. V. CHAWLA, *Data mining for imbalanced datasets: An overview*, Data mining and knowledge discovery handbook, (2010), pp. 875–886.

[23] W.-Y. CHUNG, S. BHARDWAJ, A. PUNVAR, D.-S. LEE, AND R. MYLLYLAE, *A fusion health monitoring using ecg and accelerometer sensors for elderly persons at home*, in 2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2007, pp. 3818–3821.

[24] A. CRESWELL, T. WHITE, V. DUMOULIN, K. ARULKUMARAN, B. SENGUPTA, AND A. A. BHARATH, *Generative adversarial networks: An overview*, IEEE Signal Processing Magazine, 35 (2018), pp. 53–65.

[25] S. E. CROUTER, K. G. CLOWERS, AND D. R. BASSETT, *A novel method for using accelerometer data to predict energy expenditure*, Journal of Applied Physiology, 100 (2006), pp. 1324–1331.
PMID: 16322367.

[26] I. DAUBECHIES, J. LU, AND H.-T. WU, *Synchrosqueezed wavelet transforms: An empirical mode decomposition-like tool*, Applied and Computational Harmonic Analysis, 30 (2011), pp. 243–261.

[27] V. DAVIES, E. M. SCOTT, M. L. WISEMAN-ORR, A. K. WRIGHT, AND J. REID, *Development of an early warning system for owners using a validated health-related quality of life (hrql) instrument for companion animals and its use in a large cohort of dogs*, Frontiers in veterinary science, 7 (2020), p. 575795.

[28] S. DELDALLE AND F. GAUNET, *Effects of 2 training methods on stress-related behaviors of the dog (canis familiaris) and on the dog–owner relationship*, Journal of Veterinary Behavior, 9 (2014), pp. 58–65.

[29] C. L. DELGADO, *Rising consumption of meat and milk in developing countries has created a new food revolution*, The Journal of Nutrition, 133 (2003), pp. 3907S–3910S.

[30] I. DEN UIJL, C. B. GÓMEZ ÁLVAREZ, D. BARTRAM, Y. DROR, R. HOLLAND, AND A. COOK, *External validation of a collar-mounted triaxial accelerometer for second-by-second monitoring of eight behavioural states in dogs*, Plos One, 12 (2017), p. e0188481.

[31] J. DEVLIN, M.-W. CHANG, K. LEE, AND K. TOUTANOVA, *Bert: Pre-training of deep bidirectional transformers for language understanding*, arXiv preprint arXiv:1810.04805, (2018).

[32] R. DEY AND F. M. SALEM, *Gate-variants of gated recurrent unit (gru) neural networks*, in 2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS), IEEE, 2017, pp. 1597–1600.

[33] G. Di Leo and F. Sardanelli, *Statistical significance: p value, 0.05 threshold, and applications to radiomics—reasons for a conservative approach*, European radiology experimental, 4 (2020), pp. 1–8.

[34] F. Dieterle, A. Ross, G. Schlotterbeck, and H. Senn, *Probabilistic quotient normalization as robust method to account for dilution of complex biological mixtures. application in 1h nmr metabonomics*, Analytical Chemistry, 78 (2006), pp. 4281–4290. PMID: 16808434.

[35] I. Dirgová Luptáková, M. Kubovčík, and J. Pospíchal, *Wearable sensor-based human activity recognition with transformer model*, Sensors, 22 (2022).

[36] K. Doughty, R. Lewis, and A. McIntosh, *The design of a practical and reliable fall detector for community and institutional telecare*, Journal of Telemedicine and Telecare, 6 (2000), pp. 150–154.

[37] L. Eduardo Virgilio Silva and L. Otavio Murta, *Evaluation of physiologic complexity in time series using generalized sample entropy and surrogate data analysis*, Chaos, 22 (2012), p. 043105.

[38] A. Einstein, *Relativity, the Special and the General Theory: A Popular Exposition. Translated by Robert W. Lawson*, Crown, 1961.

[39] J. P. Fabiyi, *Production losses and control of helminths in ruminants of tropical regions*, International Journal for Parasitology, 17 (1987), pp. 435 – 442.

[40] C. Fang and C. Wang, *Time series data imputation: A survey on deep learning approaches*, arXiv preprint arXiv:2011.11347, (2020).

[41] S. Faryadi and J. Mohammadpour Velni, *A reinforcement learning-based approach for modeling and coverage of an unknown field using a team of autonomous ground vehicles*, International Journal of Intelligent Systems, 36 (2021), pp. 1069–1084.

[42] A. Fischer, T. Luginbühl, L. Delattre, J. Delouard, and P. Faverdin, *Rear shape in 3 dimensions summarized by principal component analysis is a good predictor of body condition score in holstein dairy cows*, Journal of dairy science, 98 (2015), pp. 4465–4476.

[43] M. Fordellone, A. Bellincontro, and F. Mencarelli, *Partial least squares discriminant analysis: A dimensionality reduction method to classify hyperspectral data*, arXiv preprint arXiv:1806.09347, (2018).

[44] E. Friedmann, *The role of pets in enhancing human well-being: physiological*, The Waltham book of human-animal interaction: Benefits and responsibilities of pet ownership, 33 (2013).

[45] R. GARCIA, J. AGUILAR, M. TORO, A. PINTO, AND P. RODRIGUEZ, *A systematic literature review on the use of machine learning in precision livestock farming*, Computers and Electronics in Agriculture, 179 (2020), p. 105826.

[46] T. GARNETT, *Livestock-related greenhouse gas emissions: impacts and options for policy makers*, Environmental science & policy, 12 (2009), pp. 491–503.

[47] Z. GELER, V. KURBALIJA, M. IVANOVIĆ, AND M. RADOVANOVIĆ, *Weighted knn and constrained elastic distances for time-series classification*, Expert Systems with Applications, 162 (2020), p. 113829.

[48] M. GILBERT, G. NICOLAS, G. CINARDI, T. P. VAN BOECKEL, S. O. VANWAMBEKE, G. R. W. WINT, AND T. P. ROBINSON, *Global distribution data for cattle, buffaloes, horses, sheep, goats, pigs, chickens and ducks in 2010*, 5, p. 180227.

[49] C. GKEREKOS, I. LAZAKIS, AND G. THEOTOKATOS, *Machine learning models for predicting ship main engine fuel oil consumption: A comparative study*, Ocean Engineering, 188 (2019), p. 106282.

[50] E. GOMEZ-LUNA, G. APONTE MAYOR, AND J. PLEITE GUERRA, *Application of wavelet transform to obtain the frequency response of a transformer from transient signals—part ii: Practical assessment and validation*, IEEE Transactions on Power Delivery, 29 (2014), pp. 2231–2238.

[51] H. M. GORDON, *The epidemiology of parasitic diseases, with special reference to studies with nematode parasites of sheep*, Australian Veterinary Journal, 24 (1948), pp. 17–45.

[52] G. D. GRAY, J. G. CONNELL, AND V. PHIMPHACHANHVONGSOD, *Worms in smallholder livestock systems: Technologies and practices that make a difference*, Veterinary Parasitology, 186 (2011), pp. 124–31.

[53] A. GROSSMANN AND J. MORLET, *Decomposition of hardy functions into square integrable wavelets of constant shape*, SIAM J. Math. Anal., 15 (1984), pp. 723–736.

[54] M. E. GRUEN, M. ALFARO-CÓRDOBA, A. E. THOMSON, A. C. WORTH, A.-M. STAICU, AND B. D. X. LASCELLES, *The use of functional data analysis to evaluate activity in a spontaneous model of degenerative joint disease associated pain in cats*, PloS one, 12 (2017), p. e0169576.

[55] I. GUYON, J. WESTON, S. BARNHILL, AND V. VAPNIK, *Gene selection for cancer classification using support vector machines*, Machine Learning, 46 (2002), pp. 389–422.

[56] P. S. HALVORSEN, L. A. FLEISCHER, A. ESPINOZA, O. J. ELLE, L. HOFF, H. SKULSTAD, T. EDVARDSEN, AND E. FOSSE, *Detection of myocardial ischaemia by epicardial accelerometers in the pig*, BJA: British Journal of Anaesthesia, 102 (2008), pp. 29–37.

[57] A. B. HASSANAT, M. A. ABBADI, G. A. ALTARAWNEH, AND A. A. ALHASANAT, *Solving the problem of the k parameter in the knn classifier using an ensemble learning approach*, arXiv preprint arXiv:1409.0919, (2014).

[58] H. HE AND E. A. GARCIA, *Learning from imbalanced data*, IEEE Transactions on knowledge and data engineering, 21 (2009), pp. 1263–1284.

[59] R. HECHT-NIELSEN, *Theory of the backpropagation neural network*, in Neural networks for perception, Elsevier, 1992, pp. 65–93.

[60] M. HEIDEMAN, D. JOHNSON, AND C. BURRUS, *Gauss and the history of the fast fourier transform*, Archive for History of Exact Sciences, 34 (1985), pp. 265–277.

[61] S. HOCHREITER, *The vanishing gradient problem during learning recurrent neural nets and problem solutions*, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 6 (1998), pp. 107–116.

[62] S. HOCHREITER AND J. SCHMIDHUBER, *Long short-term memory*, Neural computation, 9 (1997), pp. 1735–1780.

[63] P. J. HOLDEN AND W. P. GARRIGUS, *Livestock farming*.
Encyclopedia Britannica, sep 2023.

[64] Y.-J. HONG, I.-J. KIM, S. C. AHN, AND H.-G. KIM, *Mobile health monitoring system based on activity recognition using accelerometer*, Simulation Modelling Practice and Theory, 18 (2010), pp. 446–455.
Modeling and Simulation Techniques for Future Generation Communication Networks.

[65] L. J. HORSPOOL, *Animal health markets and opportunities: Companion animal landscape*, Long Acting Animal Health Drug Products: Fundamentals and Applications, (2013), pp. 15–46.

[66] R. HUBBARD, *P–Values*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 1144–1145.

[67] A. HUSSAIN, S. ALI, H.-C. KIM, ET AL., *Activity detection for the wellbeing of dogs using wearable sensors based on deep learning*, IEEE Access, 10 (2022), pp. 53153–53163.

[68] INDUSTRIAL ELECTRONICS.COM, *Accelerometer*.

[69] K. A. JOHNSON, A. H. LEE, AND K. S. SWANSON, *Nutrition and nutraceuticals in the changing management of osteoarthritis for dogs and cats*, Journal of the American Veterinary Medical Association, 256 (2020), pp. 1335–1341.

[70] A. JOVIĆ, K. BRKIĆ, AND N. BOGUNOVIĆ, *Decision tree ensembles in biomedical time-series classification*, in Pattern Recognition, A. Pinz, T. Pock, H. Bischof, and F. Leberl, eds., Berlin, Heidelberg, 2012, Springer Berlin Heidelberg, pp. 408–417.

[71] A. KAMPOURAKI, G. MANIS, AND C. NIKOU, *Heartbeat time series classification with support vector machines*, IEEE transactions on information technology in biomedicine, 13 (2008), pp. 512–518.

[72] K. KARIMI AND H. HAMILTON, *Generation and interpretation of temporal decision rules*, International Journal of Computer Information Systems and Industrial Management Applications, 3 (2011), pp. 314–323.

[73] N. KATHPALIA AND T. GULATI, *3 axis gyro accelerometer artificial intelligence based enhancement of gps accuracy*, Measurement: Sensors, (2022), p. 100618.

[74] S. K. KATTI AND A. V. RAO, *Handbook of the poisson distribution*, Technometrics, 10 (1968), pp. 412–412.

[75] S. KHUNCHAIKARN, P. MANKEB, AND S. SUWANMANEEPONG, *Economic efficiency of beef cattle production in thailand*, 25 (2022), pp. 1–9.

[76] J.-H. KIM, *Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap."*, Comput. Stat. Data Anal., 53 (2009), pp. 3735–3745.

[77] M. KIM, *Generalized linear tree space nearest neighbor*, arXiv preprint arXiv:2103.16408, (2021).

[78] R. KRECEK, H. GROENEVELD, AND J. VAN WYK, *Effects of time of day, season and stratum on haemonchus contortus and haemonchus placei third-stage larvae on irrigated pasture*, Veterinary parasitology, 40 (1991), pp. 87–98.

[79] S. KRIEGER, G. SATTLECKER, F. KICKINGER, W. AUER, M. DRILLICH, AND M. IWERSEN, *Prediction of calving in dairy cows using a tail-mounted tri-axial accelerometer: A pilot study*, Biosystems Engineering, 173 (2018), pp. 79–84.
Advances in the Engineering of Sensor-based Monitoring and Management Systems for Precision Livestock Farming.

[80] K. J. KUBOTA, J. A. CHEN, AND M. A. LITTLE, *Machine learning for large-scale wearable sensor data in parkinson's disease: Concepts, promises, pitfalls, and futures*, Movement disorders, 31 (2016), pp. 1314–1326.

[81] B. LASCELLES, V. DEPUY, A. THOMSON, B. HANSEN, D. MARCELLIN-LITTLE, V. BIOURGE, AND J. BAUER, *Evaluation of a therapeutic diet for feline degenerative joint disease*, Journal of veterinary internal medicine, 24 (2010), pp. 487–495.

[82] S. LeBlanc, K. Lissemore, D. Kelton, T. Duffield, and K. Leslie, *Major advances in disease prevention in dairy cattle*, Journal of Dairy Science, 89 (2006), pp. 1267–1279.

[83] F. Levinzon, *Piezoelectric accelerometers with integral electronics*, 06 2015.

[84] B. Liu, Z. Zhang, and R. Cui, *Efficient time series augmentation methods*, in 2020 13th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), 2020, pp. 1004–1009.

[85] Q. Luo, Y. Meng, L. Liu, Z. Xiaofeng, and Z. Zhou, *Cloud classification of ground-based infrared images combining manifold and texture features*, Atmospheric Measurement Techniques, 11 (2018), pp. 5351–5361.

[86] Y. Luo, Z. Chen, and T. Yoshioka, *Dual-path rnn: efficient long sequence modeling for time-domain single-channel speech separation*, in ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2020, pp. 46–50.

[87] K. Maharana, S. Mondal, and B. Nemade, *A review: Data pre-processing and data augmentation techniques*, Global Transitions Proceedings, 3 (2022).

[88] E. Maniaki, *Risk factors, activity monitoring and quality of life assessment in cats with early degenerative joint disease*, 2020.
Master's Thesis at University of Bristol.

[89] I. Maqbool, Z. Wani, R. A. Shahardar, I. Allaie, and M. Shah, *Integrated parasite management with special reference to gastro-intestinal nematodes*, Journal of Parasitic Diseases, 41 (2016).

[90] J. Marais, S. P. Roux, R. Wolhuter, and T. Niesler, *Automatic classification of sheep behaviour using 3-axis accelerometer data*, 2015.

[91] C. A. Martinez-Ortiz, R. M. Everson, and T. Mottram, *Video tracking of dairy cows for assessing mobility scores*, (2013).

[92] L. R. Medsker and L. Jain, *Recurrent neural networks*, Design and Applications, 5 (2001), pp. 64–67.

[93] W. Mei, X. Yang, Y. Zhao, X. Wang, X. Dai, and K. Wang, *Identification of aflatoxin-poisoned broilers based on accelerometer and machine learning*, Biosystems Engineering, 227 (2023), pp. 107–116.

[94] S. Mejia, F. Duerr, and M. Salman, *Comparison of activity levels derived from two accelerometers in dogs with osteoarthritis: Implications for clinical trials*, The Veterinary Journal, 252 (2019), p. 105355.

[95] B. MIEKLEY, I. TRAULSEN, AND J. KRIETER, *Principal component analysis for the early detection of mastitis and lameness in dairy cows*, Journal of dairy research, 80 (2013), pp. 335–343.

[96] A. X. MONTOUT, R. S. BHAMBER, D. S. LANGE, D. Z. NDLOVU, E. R. MORGAN, C. C. IOANNOU, T. H. TERRILL, J. A. VAN WYK, T. BURGHARDT, AND A. W. DOWSEY, *Early prediction of declining health in small ruminants with accelerometers and machine learning*, bioRxiv, (2023).

[97] M. MOREAU, S. SIEBERT, A. BUERKERT, AND E. SCHLECHT, *Use of a tri-axial accelerometer for automated recording and classification of goats' grazing behaviour*, Applied Animal Behaviour Science, 119 (2009), pp. 158–170.

[98] L. A. MORENO, R. MEYER, S. M. DONOVAN, O. GOULET, J. HAINES, F. J. KOK, AND P. VAN'T VEER, *Perspective: striking a balance between planetary and human health—is there a path forward?*, Advances in Nutrition, 13 (2022), pp. 355–375.

[99] R. MORRISON, V. PENPRAZE, A. BEBER, J. REILLY, AND P. YAM, *Associations between obesity and physical activity in dogs: a preliminary investigation*, Journal of Small Animal Practice, 54 (2013), pp. 570–574.

[100] F. W. MWANGI, E. CHARMLEY, C. P. GARDINER, B. S. MALAU-ADULI, R. T. KINOBE, AND A. E. MALAU-ADULI, *Diet and genetics influence beef cattle performance and meat quality characteristics*, Foods, 8 (2019), p. 648.

[101] S. NEETHIRAJAN, *Recent advances in wearable sensors for animal health management*, Sensing and Bio-Sensing Research, 12 (2017), pp. 15–29.

[102] M. T. NOVAES, O. L. F. DE CARVALHO, P. H. G. FERREIRA, T. L. N. TIRABOSCHI, C. S. SILVA, J. C. ZAMBRANO, C. M. GOMES, E. DE PAULA MIRANDA, O. A. DE CARVALHO JÚNIOR, AND J. DE BESSA JÚNIOR, *Prediction of secondary testosterone deficiency using machine learning: A comparative analysis of ensemble and base classifiers, probability calibration, and sampling strategies in a slightly imbalanced dataset*, Informatics in Medicine Unlocked, 23 (2021), p. 100538.

[103] H. J. NUSSBAUMER, *The Fast Fourier Transform*, Springer Berlin Heidelberg, Berlin, Heidelberg, 1981, pp. 80–111.

[104] L. O'CONNOR, L. KAHN, AND W.-B. S.W, *The effects of amount, timing and distribution of simulated rainfall on the development of haemonchus contortus to the infective larval stage*, Veterinary parasitology, 146 (2007), pp. 90–101.

[105] ——, *Moisture requirements for the free-living development of haemonchus contortus: Quantitative and temporal effects under conditions of low evaporation*, Veterinary parasitology, 150 (2007), pp. 128–38.

[106] H. OVERWEG, H. N. BERGHUIJS, AND I. N. ATHANASIADIS, *Cropgym: a reinforcement learning environment for crop management*, arXiv preprint arXiv:2104.04326, (2021).

[107] F. P. O'MARA, *The significance of livestock as a contributor to global greenhouse gas emissions today and in the near future*, Animal Feed Science and Technology, 166 (2011), pp. 7–15.

[108] L. PALMERINI, J. KLENK, C. BECKER, AND L. CHIARI, *Accelerometer-based fall detection using machine learning: Training and testing on real-world falls*, Sensors, 20 (2020), p. 6479.

[109] H. M. PARSONS, D. R. EKMAN, T. W. COLLETTE, AND M. R. VIANT, *Spectral relative standard deviation: a practical benchmark in metabolomics*, Analyst, 134 (2009), pp. 478–485.

[110] M. PASTELL, J. TIUSANEN, M. HAKOJÄRVI, AND L. HÄNNINEN, *A wireless accelerometer system with wavelet analysis for assessing lameness in cattle*, Biosystems Engineering, 104 (2009), pp. 545–551.

[111] 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 ÉDOUARD DUCHESNAY, *Scikit-learn: Machine learning in python*, Journal of Machine Learning Research, 12 (2011), pp. 2825–2830.

[112] B. PERRY AND D. GRACE, *The impacts of livestock diseases and their control on growth and development processes that are pro-poor*, Philos. Trans. R. Soc. B, 364 (2009), pp. 2643–2655.

[113] B. PERRY, T. RANDOLPH, J. MCDERMOTT, K. SONES, AND P. THORNTON, *Investing in animal health research to alleviate poverty*, International Livestock Research Institute (ILRI), (2002).

[114] B. D. PERRY, D. GRACE, AND K. SONES, *Current drivers and future directions of global livestock disease dynamics*, Proceedings of the National Academy of Sciences of the U.S.A., 110 (2013), pp. 20871–20877.

[115] P. PONS, J. JAEN, AND A. CATALA, *Assessing machine learning classifiers for the detection of animals' behavior using depth-based tracking*, Expert Systems with Applications, 86 (2017), pp. 235–246.

[116] I. P. E. S. PUTRA, J. BRUSEY, E. GAURA, AND R. VESILO, *An event-triggered machine learning approach for accelerometer-based fall detection*, Sensors, 18 (2017), p. 20.

[117] W. P. B. PUTRA, S. SYAHRUDDIN, AND J. ARİFİN, *Principal component analysis (pca) of body measurements and body indices in the pasundan cows*, Black Sea Journal of Agriculture, 3 (2020), pp. 49–55.

[118] W. PYTHON, *Python*, Python Releases for Windows, 24 (2021).

[119] Y. QIAO, D. SU, H. KONG, S. SUKKARIEH, S. LOMAX, AND C. CLARK, *Individual cattle identification using a deep learning based framework*, IFAC-PapersOnLine, 52 (2019), pp. 318–323.

[120] A. RAMESH, M. PAVLOV, G. GOH, S. GRAY, C. VOSS, A. RADFORD, M. CHEN, AND I. SUTSKEVER, *Zero-shot text-to-image generation*, in International Conference on Machine Learning, PMLR, 2021, pp. 8821–8831.

[121] Y. RAMONET AND C. BERTIN, *Use of accelerometers to measure physical activity of group-housed pregnant sows. method development and use in six pig herds*, arXiv preprint arXiv:1805.00674, (2018).

[122] T. F. RANDOLPH, E. SCHELLING, D. GRACE, C. F. NICHOLSON, J. LEROY, D. COLE, M. DEMMENT, A. OMORE, J. ZINSSTAG, AND M. RUEL, *Invited review: Role of livestock in human nutrition and health for poverty reduction in developing countries*, J. Anim. Sci., 85 (2007), pp. 2788–2800.

[123] H. ROSE, C. CAMINADE, M. B. BOLAJOKO, P. PHELAN, J. VAN DIJK, M. BAYLIS, D. WILLIAMS, AND E. R. MORGAN, *Climate-driven changes to the spatio-temporal distribution of the parasitic nematode,* Haemonchus contortus*, in sheep in Europe*, Global Change Biology, 22 (2016), pp. 1271–1285.

[124] R. J. ROSSI, *Mathematical Statistics: An Introduction to Likelihood Based Inference*, 07 2018.

[125] M. SACCAREAU, G. SALLÉ, C. ROBERT-GRANIÉ, T. DUCHEMIN, P. JACQUIET, A. BLANCHARD, J. CABARET, AND C. R. MORENO, *Meta-analysis of the parasitic phase traits of* Haemonchus contortus *infection in sheep*, Parasites Vectors, 10 (2017).

[126] M. SANTOS, B. SILVA, AND A. AMARANTE, *Environmental factors influencing the transmission of haemonchus contortus*, Veterinary parasitology, 188 (2012), pp. 277–84.

[127] L. SCHNEIDER GHELLER, M. MARTINS, T. SILVA, G. FREU, M. SALLES, L. JÚNIOR, W. SOARES, AND A. NETTO, *The performance and metabolism of dairy cows receiving*

an ultra-diluted complex in the diet during the transition period and early lactation, Animals, 13 (2023), p. 3261.

[128] E. SEJDIĆ, I. DJUROVIĆ, AND J. JIANG, *Time–frequency feature representation using energy concentration: An overview of recent advances*, Digital Signal Processing, 19 (2009), pp. 153–183.

[129] M. S. SHAHRIAR, D. SMITH, A. RAHMAN, M. FREEMAN, J. HILLS, R. RAWNSLEY, D. HENRY, AND G. BISHOP-HURLEY, *Detecting heat events in dairy cows using accelerometers and unsupervised learning*, Computers and Electronics in Agriculture, 128 (2016), pp. 20–26.

[130] C. E. SHANNON, *A mathematical theory of communication*, The Bell System Technical Journal, 27 (1948), pp. 379–423.

[131] A. SHARMA, A. JAIN, P. GUPTA, AND V. CHOWDARY, *Machine learning applications for precision agriculture: A comprehensive review*, IEEE Access, 9 (2020), pp. 4843–4873.

[132] C. SHORTEN AND T. M. KHOSHGOFTAAR, *A survey on image data augmentation for deep learning*, Journal of big data, 6 (2019), pp. 1–48.

[133] A. SILVA, A. ZARZO, J. M. MACHUCA GONZÁLEZ, AND J. M. MUNOZ-GUIJOSA, *Early fault detection of single-point rub in gas turbines with accelerometers on the casing based on continuous wavelet transform*, Journal of Sound and Vibration, 487 (2020), p. 115628.

[134] L. SLINGERLAND, H. HAZEWINKEL, B. MEIJ, P. PICAVET, AND G. VOORHOUT, *Cross-sectional study of the prevalence and clinical features of osteoarthritis in 100 cats*, The veterinary journal, 187 (2011), pp. 304–309.

[135] N. P. SPRINGER AND F. DUCHIN, *Feeding nine billion people sustainably: conserving land and water through shifting diets and changes in technologies*, Environmental science & technology, 48 (2014), pp. 4444–4451.

[136] A. S. A. SUKOR, A. ZAKARIA, AND N. A. RAHIM, *Activity recognition using accelerometer sensor and machine learning classifiers*, in 2018 IEEE 14th International Colloquium on Signal Processing Its Applications (CSPA), 2018, pp. 233–238.

[137] R. S. SUTTON AND A. G. BARTO, *REINFORCEMENT LEARNING: AN INTRODUCTION, Adaptive Computation and Machine Learning series, MIT Press (Bradford Book)*, vol. 17, 03 1999.

[138] F. H. K. D. S. TANAKA AND C. ARANHA, *Data augmentation using gans*, arXiv preprint arXiv:1904.09135, (2019).

[139] A. THAVAMANI, T. J. SFERRA, AND S. SANKARARAMAN, *Meet the meat alternatives: The value of alternative protein sources*, Current nutrition reports, 9 (2020), pp. 346–355.

[140] P. K. THORNTON, P. J. ERICKSEN, M. HERRERO, AND A. J. CHALLINOR, *Climate variability and vulnerability to climate change: a review*, Global Change Biology, 20 (2014), pp. 3313–3328.

[141] A. M. TSOTETSI, S. NJIRO, T. C. KATSANDE, G. MOYO, F. BALOYI, AND J. MPOFU, *Prevalence of gastrointestinal helminths and anthelmintic resistance on small-scale farms in Gauteng Province, South Africa*, Tropical Animal Health and Production, 45 (2013), pp. 751–761.

[142] T. T. UM, F. M. PFISTER, D. PICHLER, S. ENDO, M. LANG, S. HIRCHE, U. FIETZEK, AND D. KULIĆ, *Data augmentation of wearable sensor data for parkinson's disease monitoring using convolutional neural networks*, in Proceedings of the 19th ACM international conference on multimodal interaction, 2017, pp. 216–220.

[143] J. A. VAN WYK AND G. F. BATH, *The FAMACHA system for managing haemonchosis in sheep and goats by clinically identifying individual animals for treatment*, Veterinary Research, 33 (2002), pp. 509–529.

[144] J. A. VAN WYK AND D. P. REYNECKE, *Blueprint for an automated specific decision support system for countering anthelmintic resistance in Haemonchus spp. at farm level*, Veterinary Parasitology, 177 (2009), pp. 212–23.

[145] J. A. VAN WYK, M. STENSON, J. VAN DER MERWE, R. VORSTER, AND P. VILJOEN, *Anthelmintic resistance in South Africa: Surveys indicate an extremely serious situation in sheep and goat farming*, Onderstepoort Journal of Veterinary Research, 66 (1999), pp. 273–84.

[146] A. VASWANI, N. SHAZEER, N. PARMAR, J. USZKOREIT, L. JONES, A. N. GOMEZ, Ł. KAISER, AND I. POLOSUKHIN, *Attention is all you need*, Advances in neural information processing systems, 30 (2017).

[147] A. VATTA AND A. LINDBERG, *Managing anthelmintic resistance in small ruminant livestock of resource-poor farmers in South Africa*, J. S. Afr. Vet. Assoc., 77 (2006), pp. 2–8.

[148] J. VAZQUEZ DIOSDADO, V. PAUL, K. ELLIS, D. COATES, R. LOOMBA, AND J. KALER, *A combined offline and online algorithm for real-time and long-term classification of sheep behaviour: Novel approach for precision livestock farming*, Sensors, 19 (2019), p. 3201.

[149] J.-P. VERT, *Kernel Methods in Genomics and Computational Biology*, 09 2008, pp. 294–313.

[150] J. WAINER AND P. FONSECA, *How to tune the rbf svm hyperparameters? an empirical evaluation of 18 search algorithms*, Artificial Intelligence Review, 54 (2021), pp. 4771–4797.

[151] S. WALKDEN-BROWN, L. KAHN, ET AL., *Ecology of the free-living stages of major trichostorngylid parasites of sheep*, Veterinary Parasitology, 14 (2006), pp. 1–15.

[152] J. G. WALKER, M. OFITHILE, F. M. TAVOLARO, J. A. VAN WYK, K. EVANS, AND E. R. MORGAN, *Mixed methods evaluation of targeted selective anthelmintic treatment by resource-poor smallholder goat farmers in Botswana*, Veterinary Parasitology, 214 (2015), pp. 80–88.

[153] T. WANG, J. VAN WYK, A. MORRISON, AND E. MORGAN, *Moisture requirements for the migration of* Haemonchus contortus *third stage larvae out of faeces*, Veterinary Parasitology, 204 (2014), pp. 258–264.

[154] G. M. WEISS, A. NATHAN, J. KROPP, AND J. W. LOCKHART, *Wagtag: a dog collar accessory for monitoring canine activity levels*, in Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication, 2013, pp. 405–414.

[155] P. J. WERBOS, *Backpropagation through time: what it does and how to do it*, Proceedings of the IEEE, 78 (1990), pp. 1550–1560.

[156] J. WERNER, C. UMSTATTER, L. LESO, E. KENNEDY, A. GEOGHEGAN, L. SHALLOO, M. SCHICK, AND B. O'BRIEN, *Evaluation and application potential of an accelerometer-based collar device for measuring grazing behavior of dairy cows*, animal, 13 (2019), p. 2070–2079.

[157] S. XU, Q. LUO, H. LI, AND L. ZHANG, *Time series classification based on attributes weighted sample reducing knn*, in 2009 Second International Symposium on Electronic Commerce and Security, vol. 2, 2009, pp. 194–199.

[158] Y. YOO AND J.-G. BAEK, *A novel image feature for the remaining useful lifetime prediction of bearings based on continuous wavelet transform and convolutional neural network*, Applied Sciences, 8 (2018).

[159] J. YOON, J. JORDON, AND M. VAN DER SCHAAR, *Gain: Missing data imputation using generative adversarial nets*, 2018.

[160] J. YOON, W. R. ZAME, AND M. VAN DER SCHAAR, *Estimating missing data in temporal data streams using multi-directional recurrent neural networks*, IEEE Transactions on Biomedical Engineering, 66 (2019), pp. 1477–1490.

[161] A. ZIEN, N. KRÄMER, S. SONNENBURG, AND G. RÄTSCH, *The feature importance ranking measure*, arXiv preprint arXiv:0906.4258, (2009).