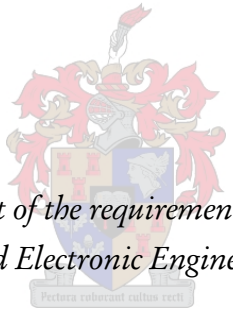
Real-Time Behaviour Classification Techniques in Low-Power Animal Borne Sensor Applications

by

Solomon Petrus le Roux



Thesis presented in partial fulfilment of the requirements for the degree of Doctor of Philosophy in the Department of Electrical and Electronic Engineering at the University of Stellenbosch

Promoters: Prof TR Niesler and Dr R Wolhuter

April 2019

Stellenbosch University https://scholar.sun.ac.za

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Solomon Petrus le Roux

April 2019

Copyright © 2019 Stellenbosch University
All rights reserved

Stellenbosch University https://scholar.sun.ac.za

Abstract

The ability to study animal behaviour is important in many areas of science, including behavioural ecology, conservation and precision farming. These studies typically employ biotelemetry tags attached to animals that collect raw sensor data from tri-axial accelerometers. However, conventional animal behaviour classification techniques are performed offline as a post-processing step and does not provide real-time data analysis. Furthermore, the lifespan of such tags is constrained by their power and memory usage, which are often limiting factors when performing behavioural studies for extended periods of time. The focus of this project was to investigate methods to possibly mitigate these limitations. The main contributions of the work set out in this dissertation are three-fold. First, a novel embedded automatic behaviour classification system which captures and automatically classifies three-dimensional accelerometer data in real-time is presented. All computation occur on specially designed biotelemetry tags while attached to the animal. This allows the probable real-time behaviour to be transmitted continuously, thereby providing an enhanced level of detail and immediacy. As a result of the sustained and serious rhinoceros poaching in South Africa, the behaviour classification system was developed to assist with activities combating this problem. An onboard linear support vector machine with 11 features achieves an accuracy of $99.61\,\%$ among three behavioural classes (standing, walking and lying down). Stock theft is another significant problem as experienced in the agricultural sector. The behaviour classification system was, therefore, also implemented for sheep. In this case, logistic regression with 34 features achieves a classification accuracy of $89.59\,\%$ among five behavioural classes (standing, walking, grazing, running and lying down). The estimated behaviour was established approximately every $6.5\,\mathrm{s}$ and transmitted to a receiver station for both rhinoceros and sheep. Secondly, a novel energy-aware feature and model selection technique is presented. A greedy sequential feature selection algorithm was utilised to minimise a cost function that weighs the energy expense of adding specific features with the change in classification error afforded by the features. In addition, the energy expense of specific classification techniques are considered in selecting the optimal models, which is often neglected in literature. Our technique, therefore, favours both classifiers and features which are less energy expensive to compute during runtime. It is shown that, for the rhinoceros dataset, a random forest classifier with two fea-

Stellenbosch University https://scholar.sun.ac.za

tures is selected as optimal, achieving an overall classification accuracy of 99.33 %. Extracting the features and performing classification consumes 363 times less energy, while only sacrificing 0.28 % in accuracy when compared to the 99.61 % achieved with the unconstrained system. For the sheep dataset, a linear support vector machine with nine features achieves an 88.40 % classification accuracy. Extracting the features and performing classification consumes 6.8 times less energy, at a cost of 1.19 % in accuracy compared to the 89.59 % achieved with the unconstrained system. Finally, the reduced power requirements and memory usage benefits of the embedded behaviour classification system were considered. Experiments using the biotelemetry tags demonstrated a 14-fold reduction in energy consumption and a 234-fold reduction in memory usage when classification was performed on the tag vs. processing raw data subsequent to transmission. It is concluded that real-time behavioural updates can be achieved by means of embedded behaviour classification with the technique significantly reducing the total energy consumption and memory requirements of the device. This enables long-term behavioural studies in applications such as the conservation of rhinoceros, which is a critically endangered species. It is also very applicable to precision farming applications. Moreover, this technique can be applied to general embedded machine learning applications employed in smart phones, smart watches and sensors within the internet of things.

Stellenbosch University https://scholar.sun.ac.za

Opsomming

Die vermoë om dieregedrag te bestudeer is belangrik in baie areas van die wetenskap, insluitend gedragsekologie, natuurbewaring en presisie boerdery. Hierdie studies gebruik tipies bio-telemetriese toestelle, vasgeheg aan diere, wat rou sensor data insamel vanaf drie-as-versnellings-sensors. Konvensionele gedragsklassifikasie-tegnieke word egter aflyn uitgevoer en verskaf dus nie intydse data nie. Die leeftyd van sulke toestelle word beperk deur hul krag- en geheueverbruik, wat dikwels beperkende faktore is tot die uitvoer van langtermyn gedragstudies. Hierdie projek het gefokus op tegnieke om hierdie beperkinge te verlig. Die belangrikste bydrae van die werk is drievoudig. Eerstens word 'n nuwe aanboord outomatiese gedragsklassifikasie stelsel aangebied, wat intyds drie-as-versnellingssensor data insamel en klassifiseer. Alle berekeninge vind plaas op die bio-telemetriese toestel terwyl dit aan die dier vas is. Dit stuur dan die waarskynlike gedrag deurlopend aan 'n ontvangerstasie en bied sodoende 'n verbeterde vlak van data beskikbaarheid. As gevolg van voortgesette en ernstige renosterstropery in Suid-Afrika, was 'n gedragsklassifikasie stelsel ontwikkel om aktiewe pogings teen renosterstropery te ondersteun. 'n Aanboord *linear support vector machine* met 11 kenmerke behaal 'n akkuraatheid van $99.61\,\%$ tussen drie gedragsklasse (staan, loop en lê). Veediefstal is nog 'n beduidende probleem wat deur die landbou sektor beleef word. 'n Gedragsklassifikasie stelsel was daarom ook vir skape ontwikkel. 'n Aanboord *logistic regression* model met 34 kenmerke behaal 'n klassifikasie akkuraatheid van $89.59\,\%$ tussen vyf gedragsklasse (staan, loop, wei, hardloop en lê). Die beraamde gedrag word ongeveer elke $6.5\,\mathrm{s}$ bepaal en gestuur na 'n ontvangerstasie vir beide renosters en skape. Verder, word 'n nuwe energie-bewuste kenmerk en model seleksie tegniek beskryf. 'n Gulsige sekwensiële kenmerk seleksie algoritme word gebruik om 'n kostefunksie te minimeer wat die energiekoste om spesifieke kenmerke uit te werk balanseer met die verandering in die klassifikasiefout wat deur die kenmerke behaal word. Die energiekoste van spesifieke klassifikasietegnieke word dan addisioneel oorweeg om die optimale modelle te kies. Daar word getoon dat vir die renosterdatastel, 'n *random forest* model met twee optimale kenmerke gekies word wat dan 'n algehele klassifikasie akkuraatheid van $99.33\,\%$ bereik. Die berekening van die kenmerke en die uitvoer van die klassifikasie model gebruik 363 keer minder energie, terwyl dit net $0.28\,\%$ in akkuraatheid prysgee as dit vergelyk word met die $99.61\,\%$ wat behaal word met

Stellenbosch University https://scholar.sun.ac.za

die onbeperkte stelsel. Vir die skaapdatastel, behaal 'n *linear support vector machine* met nege kenmerke, 'n klassifikasie akkuraatheid van $88.40\,\%$. Die berekening van die kenmerke en die uitvoer van die klassifikasie model gebruik 6.8 keer minder energie, teen 'n koste van $1.19\,\%$ verminderinge in akkuraatheid, wanneer dit vergelyk word met die $89.59\,\%$ wat behaal word met die onbeperkte stelsel. Laastens, is die voordele in terme van die verminderde krag- en geheueverbruik van die bio-telemetriese toestelle deur van die aanboord gedragsklassifikasie tegniek gebruik te maak, ondersoek. Eksperimente wys dat hierdie bio-telemetriese toestelle 'n 14 keer verlaging in energieverbruik en 'n 234 keer vermindering in geheueverbruik behaal. Dit kan toegeskryf word aan die onmiddellike klassifikasie wat uitgevoer word op die toestel self, i.p.v. dataverwerking op 'n rekenaar na die transmissie van rou-data. Ons kom tot die gevolgtrekking dat intydse gedrags-opdaterings bereken en beskikbaar gestel kan word d.m.v. aanboord gedragsklassifikasie en dat die tegniek die totale energieverbruik en geheuevereistes van die toestel drasties verminder. Dit stel langtermyn gedragsstudies in staat en kan in toepassings benewens die bewaring van renosters ook in presisie boerdery gebruik word. Daarbenewens, kan hierdie tegniek ook toegepas word in algemene aanboord masjienleer-toepassings in slimfone, slimhorlosies en toestelle binne die internet-van-dinge.

Stellenbosch University https://scholar.sun.ac.za

Publications

Sections of the work presented in this dissertation have been published as follows.

International journal articles:

- S. P. le Roux, J. Marais, R. Wolhuter, and T. Niesler, "Animal-borne behaviour classification for sheep (dohne merino) and rhinoceros (ceratotherium simum and diceros bicornis)," Animal Biotelemetry, vol. 5, no. 25, Nov 2017.
- S. P. le Roux, R. Wolhuter, N. Stevens, and T. Niesler, "Reduced energy and memory requirements by on-board behavior classification for animal-borne sensor applications," IEEE Sensors Journal, vol. 18, no. 10, pp. 4261-4268, May 2018.
- S. P. le Roux, R. Wolhuter, and T. Niesler, "Energy-Aware Feature and Model Selection for Onboard Behaviour Classification in Low-Power Animal Borne Sensor Applications," submitted to IEEE Sensors Journal.

International conference articles:

- S. P. le Roux, R. Wolhuter, T. Niesler, and N. Stevens, "Energy benefits of on-board behaviour classification for animal-borne sensor applications," in 2017 IEEE SENSORS, 2017.
- S. P. le Roux, R. Wolhuter, and T. Niesler, "An overview of automatic behaviour classification for animal-borne sensor applications in South Africa," in SAWACMMM'17, October 23, 2017, Mountain View, CA, USA.

Stellenbosch University https://scholar.sun.ac.za

Acknowledgements

I would like to thank you for taking the time to read this dissertation. I would also like to thank the following people:

- Prof Niesler and Dr Wolhuter for supervising this project, for their advice and support. Furthermore, for their friendship and character which, certainly, conveyed valuable life skills and perspectives.
- Prof Johan du Preez for his advice, friendship and for cultivating a deep appreciation of probabilistic reasoning.
- All my colleagues in the DSP lab.
- Wessel Croukamp and Sarel Rautenbach for the availability of their expertise and manufacturing facilities.
- Arrie van Deventer and The Rhino Orphanage for enabling physical measurements on rhinoceros.
- Barend, Amanda and Francois, my father, mother and brother, for their love, advice and support.
- Fanie and Elmien Dippenaar, my in-laws, for their love, advice, support and especially, for enabling practical measurements by providing access to their sheep, which graze the humble-rocky mountains of Rooivlei.
- My family and friends for their advice and support.
- Salomi, my wife, for her loving-kindness, friendship and support.

Soli Deo Gloria

Stellenbosch University https://scholar.sun.ac.za

Dedication

I dedicate this dissertation to my beloved wife, Salomi.

Stellenbosch University https://scholar.sun.ac.za

Contents

Stellenbosch University https://scholar.sun.ac.za

Stellenbosch University https://scholar.sun.ac.za

Stellenbosch University https://scholar.sun.ac.za

List of Figures

Stellenbosch University https://scholar.sun.ac.za

Stellenbosch University https://scholar.sun.ac.za

Stellenbosch University https://scholar.sun.ac.za

Stellenbosch University https://scholar.sun.ac.za

List of Tables

Stellenbosch University https://scholar.sun.ac.za

Stellenbosch University https://scholar.sun.ac.za

Stellenbosch University https://scholar.sun.ac.za

Nomenclature

Acronyms

CSMA/CA	carrier-sense multiple access with collision avoidance
DT	decision tree
FFT	fast Fourier transform
FRAM	ferro-electric non-volitile random acess memory
GPS	global posisioning system
IIC	inter-integrated circuit
k-NN	k-nearest neighbour
LDA	linear discriminant analysis
LED	light emitting diode
LR	logistic regression
LSVM	linear support vector machine
MCU	microcontroller unit

Stellenbosch University https://scholar.sun.ac.za

NB	naive Bayes
PCB	printed circuit board
QDA	quadratic discriminant analysis
RF	random forest
SD	secure digital
SFS	sequential freature selection
SPI	serial peripheral interface
UART	universal asynchronous receive transmit

Variables

$\mathbf{x} \in \mathbb{R}^n$	real-valued vector with dimension n
$\mathbf{M} \in \mathbb{R}^{n \times m}$	real-valued matrix with dimensions $n \times m$
$\mathcal{A}$	set
$\hat{\mathbf{x}}$	fast Fourier transform of $\mathbf{x}$

Symbols

$\varnothing$	empty set
A_{cc}	classification accuracy [%]
P_r	precision [%]
R_e	recall [%]
$F1$	$F1$ score [%]
E_{rr}	classification error [%]

Stellenbosch University https://scholar.sun.ac.za

e_{rr}	minimum classification error [%]
T_p	true positive
F_p	false positive
T_n	true negative
F_n	false negative
k	behavioural class number
K_{tot}	number of behavioural classes
K	number of folds in K-fold cross-validation
$\mathbf{x}$	input feature vector of dimension d
y	predicted class label
F_N	optimal number of features
α	model hyper-parameters
$C(\alpha)$	classification model with specific hyper-parameters
m	trained classification model
f	clock frequency [Hz]
C_e	effective switching capacitance [F]
V_{cc}	supply voltage [V]
i, $\boldsymbol{i}$	ampere [A]
R	resistance [Ω]
T_s	sampling interval [s]
v, $\boldsymbol{v}$, V	voltage [V]
P	power [W]
E	energy [J]
E_f	energy expense of feature extraction [J]
E_c	energy expense of classification [J]
E_t	total energy expense [J]
i_{avg}	average current consumption [A]
i_{active}	active mode current consumption [A]

Stellenbosch University https://scholar.sun.ac.za

i_{sleep}	sleep mode current consumption [A]
t_{active}	time required to process a frame [s]
t_{tot}	update interval [s]
$\mathbf{D}_R$	rhinoceros dataset
$\mathbf{D}_S$	sheep dataset
λ	cost function weight
N_s	number of training examples
K_{nn}	number of nearest neighbours
x	accelerometer x-axis measurements
y	accelerometer y-axis measurements
z	accelerometer z-axis measurements
γ	margin of support vector machine

Operators

$x \in \mathcal{X}$	x belongs to set $\mathcal{X}$
$P(\mathbf{x})$	probability of $\mathbf{x}$
$h(\mathbf{x})$	hypothesis function
$D(\mathbf{x}, \mathbf{y})$	euclidean distance between vectors $\mathbf{x}$ and $\mathbf{y}$
$\mathbf{X}^{(i)}$	i^{th} vector in the matrix $\mathbf{X}$
$\mathbf{x}^{(i)}$	i^{th} element in the vector $\mathbf{x}$
I	indicator function
J	cost function
$f_k(\mathbf{x})$	class conditional probability of $\mathbf{x}$
π_k	prior probability of class k

Stellenbosch University https://scholar.sun.ac.za

$\bar{x}$	mean of $\mathbf{x}$
σ_x	standard deviation of $\mathbf{x}$

Stellenbosch University https://scholar.sun.ac.za

Chapter 1

Introduction

1.1 Background to the project

The study of animal behaviour is important in many areas of science, including behavioural ecology, conservation and precision farming. Studying animal movement provides key insights on how animals utilise their habitat, their foraging strategies, as well as dispersal and migration. Furthermore, it provides insights on individual health, fitness, reproductive success and survival, which ultimately drive population dynamics and the evolution of species. Researchers collect large datasets from different animal-attached sensors and employ various statistical analysis and classification techniques to study animal behaviour. In particular, offline behaviour classification, based on tri-axial accelerometer data, has become common practice and has been applied to various taxa.

In recent years, however, researchers have experienced an increased demand for real-time data analysis. This presents a major challenge as biotelemetry tags are typically small and light weight, so as not to harm or impede the animal under study. As a consequence, the device must be powered by a small battery and necessitates the use of ultra low-power microcontrollers which are limited in terms of available processing power and memory. Furthermore, to enable real-time analysis, data needs to be transmitted wirelessly to a receiver station and such transmissions are energy demanding. As a result of the resource limitations of biotelemetry devices, behavioural studies are typically short-lived and range from days to weeks. This limits the insights that can be gained from such studies.

One application that may benefit greatly from long-lived animal-attached sensors is the conservation of endangered wildlife. For example, rhinoceros poaching in South Africa has reached alarming proportions. The work presented in this thesis was motivated by the goal of determining how the real-time provision of animal behaviour could aid nature conservationist

Stellenbosch University https://scholar.sun.ac.za

in the fight against this illegal activity. A secondary goal was the exploration of how the same technique can be applied to sheep, to aid precision farming. In both cases, systems that provide both long-term and real-time automatic behaviour classification are currently still lacking.

1.2 Project objectives

The work as presented in this dissertation set out to design and evaluate an embedded automatic classification system that performs behaviour classification in real-time on the biotelemetry tag itself. This is in contrast to the current approach of behaviour classification as a subsequent process carried out at a remote base station. Techniques are developed that allow the optimisation of the proposed classification system in terms of both energy consumption and model performance, which has the direct and important benefit of increasing its battery life. Specific objectives of the work as presented are as follows.

- To design, implement and evaluate an optimal offline automatic behaviour classification system for both rhinoceros and sheep in order to achieve the highest possible classification accuracy.
- To design, implement and evaluate a novel real-time embedded behaviour classification system for both rhinoceros and sheep in order to achieve the highest possible classification accuracy, while periodically providing behavioural updates.
- To design, implement and evaluate a novel energy-aware feature- and model-selection framework to optimise and select classification models and feature sets for both rhinoceros and sheep, which are optimal in terms of both energy consumption and model performance.
- To evaluate these optimal models and determine whether real-time automatic behaviour classification is advantageous in terms of energy consumption and memory utilisation, by means of accurate measurements on the physical hardware.

Stellenbosch University https://scholar.sun.ac.za

1.3 Project summary

This dissertation can be summarised as follows.

- A novel embedded automatic behaviour classification system is presented which captures and automatically classifies three-dimensional accelerometer data in real-time. All computations occur on specially designed biotelemetry tags while attached to the animal. This allows the probable behaviour to be transmitted continuously, thereby providing an enhanced level of detail and immediacy. For rhinoceros, a linear support vector machine with 11 features achieves an accuracy of $99.61\,\%$ among three behavioural classes (standing, walking and lying down). For sheep, logistic regression with 34 features achieves a classification accuracy of $89.59\,\%$ among five behavioural classes (standing, walking, grazing, running and lying down). The estimated behaviour was established approximately every $6.5\,\mathrm{s}$ for both rhinoceros and sheep and transmitted to a receiver station.

- A novel energy-aware feature and classification model selection technique is described. A greedy sequential feature selection algorithm was utilised to minimise a cost function. This function incorporates a linear weighting of the energy expense of adding specific features and the change in classification error afforded by the added features. In addition, the energy expense of different classification algorithms was considered in selecting the optimal models, which is disregarded in current studies. The proposed technique therefore favours classifiers and features which in combination are less energy expensive to compute at runtime. For the rhinoceros dataset, a random forest classifier with two features is selected as optimal and achieves an overall classification accuracy of $99.33\,\%$. Extracting the features and performing classification consumes 363 times less energy, while only sacrificing $0.28\,\%$ in accuracy when compared to the $99.61\,\%$ achieved with the unconstrained system. For the sheep dataset, a linear support vector machine with nine features achieves an $88.40\,\%$ classification accuracy. Extracting the features and performing classification consumes 6.8 times less energy, at a cost of $1.19\,\%$ less in accuracy compared to the $89.59\,\%$ achieved with the unconstrained system.

- Finally, the reduced energy requirements and memory usage benefits of the embedded behaviour classification system are considered. Experiments using the biotelemetry tags demonstrated a 14-fold reduction in energy consumption and a 234-fold reduction in memory usage when classification was performed on the tag, when contrasted with the processing subsequent to raw data transmission.

Stellenbosch University https://scholar.sun.ac.za

1.4 Project outcomes and contributions

The primary contributions of the work conducted under this project are:

- A novel real-time embedded behaviour classification system for both rhinoceros and sheep as presented in [2].
- A novel energy-aware feature and classification model selection technique that utilises a greedy sequential feature selection algorithm to minimise a cost function. This function incorporates a linear weighting of the energy expense of adding specific features and the change in classification error afforded by the added features. In addition, the energy expense of different classification algorithms was considered in selecting the optimal models, which is disregarded in current studies. The presented technique, therefore, favours classifiers and features which in combination are less energy expensive to compute at runtime.
- The significant reduction of energy and memory requirements of biotelemetry devices by means of real-time embedded behaviour classification, as set out in [3]. This is generally a very significant obstacle in most electronic devices utilised for on-animal behavioural studies.

The secondary contributions of the work set out in this dissertation are:

- A grouped nested and repeated cross-validation algorithm developed to perform model optimisation, validation and evaluation. The complete algorithm presented in this dissertation ensures model reproducibility for machine learning applications in general and is not limited to animal behaviour classification.
- An automatic behaviour classification system for both rhinoceros and sheep which achieve high model performance among common behaviours such as standing, grazing, walking, running and lying down. Furthermore, datasets of rhinoceros and sheep acceleration measurements are compiled in order to perform experimental evaluation.

1.5 Structure of this dissertation

This dissertation is structured as follows. Chapter 2 presents a broad literature study and evaluates recent advances in the fields of automatic behaviour classification and energy-aware feature selection. Chapter 3 provides theoretical and mathematical models for the different statistical classification techniques considered in this dissertation and defines the evaluation metrics used to indicate model performance. Chapter 4 describes the hardware used in this

Stellenbosch University https://scholar.sun.ac.za

study and explains how raw datasets were collected for both rhinoceros and sheep. Furthermore, the data pre-processing, dataset balancing and feature extraction steps are described. The feature selection and cross-validation frameworks employed are also considered. Chapter 5 describes how feature selection, model training and evaluation are performed within a grouped nested and repeated cross-validation framework for both the rhinoceros and sheep datasets. For each dataset the performance of each classification technique is presented. The best performing models are identified for both the rhinoceros and sheep behaviour classification systems. Chapter 6 shows how the optimal automatic behaviour classification systems, identified in Chapter 5 for both rhinoceros and sheep, are implemented on the physical hardware to enable real-time embedded behaviour classification. Chapter 7 presents an energy-aware feature- and model selection technique that utilises a greedy sequential feature selection algorithm and incorporates the energy expense of the classification technique to select an optimal configuration for both rhinoceros and sheep. Chapter 8 presents a set of detailed power measurements and memory utilisation comparisons that demonstrate the advantages of real-time embedded behaviour classification over conventional techniques. Chapter 9 concludes with final remarks and recommendations for future work.

Stellenbosch University https://scholar.sun.ac.za

Chapter 2

Literature review

In this work the feasibility of an animal borne behaviour classification system which captures and automatically classifies three-dimensional accelerometer data in real-time and on the animal is considered. Furthermore, we investigate whether real-time automatic classification of animal behaviour can result in a significant reduction in power consumption and memory utilisation. In addition, we explore the feasibility of optimising such an embedded system in terms of energy consumption and model performance by means of an energy-aware feature- and model-selection procedure. The goal of our system is to perform automatic behaviour classification for both sheep and rhinoceros in order to distinguish in real-time and in a power-efficient manner, between behaviours such as running, walking, standing, grazing and lying down.

2.1 Introduction

The ability to study animal behaviour is important in many fields of science, including behavioural ecology, conservation and precision farming. Studying animal movement provides key insights on how animals utilise their habitat, their foraging strategies, as well as dispersal and migration. Furthermore, it provides insights on individual health, fitness, reproductive success and survival, which ultimately drive population dynamics and the evolution of species [4]. In recent years, the use of animal attached sensors became common practice and provide researchers with high resolution spatial and temporal information detailing the behaviour of animals. Depending on the sensors used, aspects such as energy expenditure, behaviour activities, location, speed, heart rate and temperature can be monitored [5, 6, 7, 8, 9]. However, animal attached sensors have limited battery capacity, typically resulting in short lived studies, which provide high resolution information. Techniques that enable long-term real-time behavioural studies while maintaining acceptable levels of data resolution are still lacking. In this work we explore how the energy and memory requirements of animal-attached sensor-

Stellenbosch University https://scholar.sun.ac.za

tags can be reduced to enable long-term behavioural studies while maintaining good model performance. Due to sustained rhinoceros poaching and extensive stock theft activities in South Africa, we show the application of our techniques in the fields of nature conservation and precision farming.

2.2 Rhino poaching and stock theft

South Africa is home to a wide variety of wildlife. This includes the iconic Big Five (African lion, African elephant, Cape buffalo, African leopard, and rhinoceros) which are recognised both locally and internationally. However, some of these animals have increasingly become the victims of illegal poaching activity. Poaching is sometimes motivated by superstitious beliefs that certain animal body parts have medicinal value [10]. It may also be motivated by the social status bestowed by the possession of trophies, skins, tusks or horns. For example, in Yemen men wear short curved daggers with the handles often sculpted out of rhino horn as a sign of their masculinity, wealth and social class [11]. As animal populations decline, the body parts in question become rarer and their monetary value in illegal trading increases. This is especially true for rhinoceros horn [12] and has led to a dramatic population decline [13, 14, 15, 16] in areas such as the Kruger National Park, located in the North East of South Africa. As a result of poaching to sustain the illegal trade of rhinoceros horn, the world's rhinoceros population is currently under severe stress. The estimated population numbers reported in 2018 for the five rhinoceros species show that a mere 67 Javan rhinoceros and less than 80 Sumatran rhinoceros are left on earth. Black rhinoceros are estimated at between 5040 and 5458 individuals and White rhinoceros between 19666 and 21085. The greater one-horned rhinoceros has a estimated population of just over 3500 animals [1]. Figure 2.1 shows the rapid acceleration of poaching over the last 10 years in South Africa alone [1]. Over the past three years anti-poaching strategies employed in South Africa aided towards a slight reduction in this illegal activity. However, the country is still losing more than 1000 animals per year. In total South Africa has lost more than 7143 rhinoceros in the last decade due to poaching. This rapid population decline, as well as the brutal methods employed by the poachers, has recently afforded international attention to the fate of the African rhinoceros.

Stock theft is also a major problem in South Africa and has serious economic consequences for local farmers. Stock to the value of R4.6 billion was stolen in a recent period of six years [17], with R819 million lost in the 2015/16 financial year alone [18]. Local farmers are desperately searching for new techniques to monitor their herds in order to proactively fight against stock theft.

Stellenbosch University https://scholar.sun.ac.za

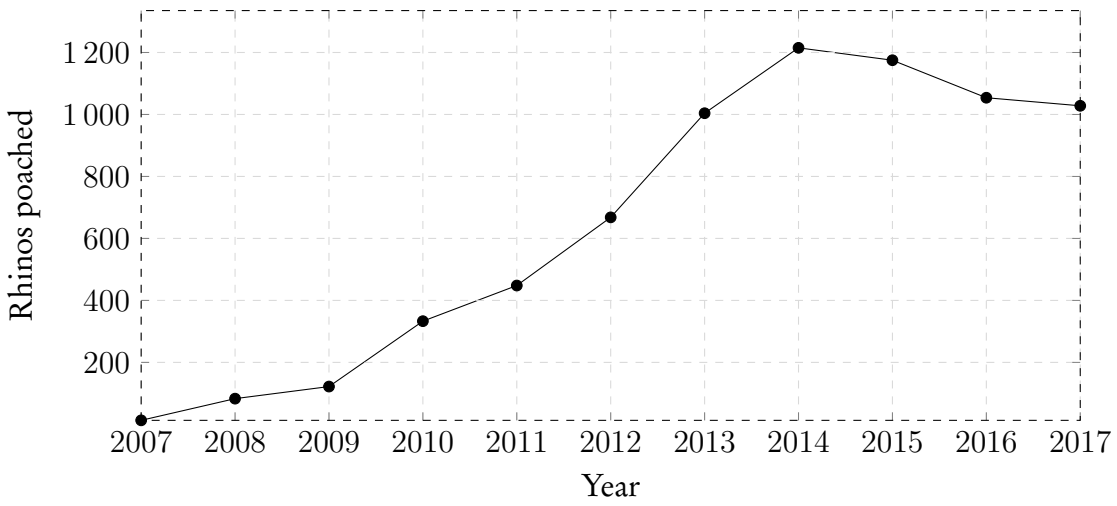


Figure 2.1: Number of rhinos poached in South Africa from 2007 to 2017 [1].

We believe that new technological advances with relevant and real-time information of animal behaviour are required to aid nature conservationists and law enforcement personnel so that they can assess the behaviour profile and act accordingly. Monitoring specific animal behaviours or activities in real-time can provide valuable insights in determining whether the animal is showing normal or abnormal behaviour. Animal wariness can be a valuable indicator of illegal hunting activities and the extent of human exploitation [19]. Caro studied the behaviour, especially the wariness, of different animal species in hunted and non-hunted areas. The author found that upon a human-animal interaction the animals typically stand still for a short duration, whereafter they flee a short distance and subsequently stand to intensely watch the intruder. The study revealed that animals in hunted areas were more likely to flee than animals in non-hunted areas. The findings were also consistent over species experiencing greater pressure from hunters, indicating that individual species learn and react to the pressure applied to them. The author suggests that, in general, the rule also applies to most large African mammals, and if an observer notices some species to either flee or watch the observer on more than half of the encounters, it is likely that the animals are suffering from heavy anthropogenic exploitation. The author concludes that animal behaviour should play a direct and important part in conversation biology [19].

Based on these findings we believe that rhinoceros would exhibit the same behavioural traits due to sustained pressure from poaching activities. Signs that an animal is wary, in distress and trying to escape from a poacher can possibly be detected when a rhino starts walking into previously sub-utilised areas or suddenly shows excessive or unexpected running behaviour. Similarly, if a rhinoceros lies in the same location for an abnormal duration of time, it might indicate that the animal is dead. The study of animal behaviour within a spatial- and temporal-context could provide possible clues to the normality of the observed be-

Stellenbosch University https://scholar.sun.ac.za

haviour. Similarly, stock theft can promptly be detected when a herd of sheep suddenly starts walking or running at night, when they typically lay down and rest. Apart from poaching detection, animal behaviour can also be evaluated in precision farming settings. For example, monitoring animal behaviour can lead to the prompt detection and treatment of diseases such as lameness since it affects animal lying behaviour by increasing total lying time and the number of lying bouts [20]. Lameness has serious economic consequences as it causes a loss in milk production which leads to the early culling of animals [21]. Furthermore, estimating forage intake and monitoring the feeding behaviour of animals are key activities to evaluate their health and welfare state [22].

It is important to make the subtle distinction that we do not propose an anti-poaching solution, but rather a device which provides real-time updates on animal behaviour. This device can be used in different fields of research and applications, including anti-poaching and conservation which provides relevant parties with real-time information for further data analysis and action. For this purpose, we investigate recent advances in the field of automatic behaviour classification.

2.3 Offline automatic behaviour classification

The study of animal behaviour has greatly advanced through the use of animal-attached sensor tags which include sensors such as global posisioning system (GPS) trackers, tri-axial accelerometers, temperature sensors, pressure sensors and magnetometers [5, 6, 7, 8, 9]. These tags are attached to the animal of interest in order to record and later analyse the raw sensor measurements. A commonly used sensor to determine specific animal activities is the accelerometer [23, 24]. Animal-attached tri-axial accelerometers provide a detailed picture of the activity patterns and allow the collection of data from animals in a non-invasive manner and the subsequent analysis of their behaviour. However, the manual monitoring and analysis of such accelerometer measurements can be tedious, time consuming and not feasible for large or open-ended data volumes.

Recently researchers have begun to consider machine learning techniques to automatically classify these large datasets of tri-axial accelerometer measurements into behavioural classes. This is achieved by simultaneously recording raw accelerometer measurements by means of animal attached sensors and recording high resolution time-stamped video recordings of the animal's behaviour during data collection. These recordings serve as ground truth for the raw data which is used to manually label the data according to specific be-

Stellenbosch University https://scholar.sun.ac.za

haviours. The labelled data is subsequently used to train and evaluate statistical classifiers. These classifiers can then automatically categorise large datasets according to the chosen behaviours with high accuracy. For example, McClune et al. use machine learning to classify the behaviour of badgers (Meles meles) as walking, trotting, snuffling and resting [25]. Automatic behaviour classification has already been applied to a range of animals using various statistical classification and machine learning techniques, including, artificial neural networks [26, 27, 28, 29], decision trees [25, 26, 30, 31, 32, 33, 34, 35, 36], discriminant function analysis [37], hidden Markov models [38, 39], k-nearest neighbours [25, 30, 40, 41, 42], logistic regression [28, 43, 44], linear discriminant analysis [26, 45], moving averages with thresholds [46], naive Bayes classifiers [47, 42], quadratic discriminant analysis [48], random forests [26, 49, 41, 50, 28, 51, 52, 44, 43, 53, 36, 54, 55] and support vector machines [26, 38, 56, 47, 42, 44, 39]. These systems have accomplished automatic behaviour classification with high accuracy, as shown in Table 2.1. The table lists examples of classified behaviours for a range of animal species which have recently been studied by using automatic behaviour classification techniques.

A major obstacle in the study of animal behaviour remains the fairly short lifespan of the tags, which typically range from days to a few weeks due to the limited battery power and available onboard storage. For example, a study monitoring the behaviour of badgers reports the battery of the device to be depleted during the third night after deployment [30]. A study monitoring the behaviour of chipmunks had a 4.5 day data collection period due to the memory constraints of the device [39]. Similarly a study monitoring sheep behaviour achieved 11 hours of data collection on each day of deployment [36]. Different durations of 20, 10 and five hours were achieved in a study monitoring sheep behaviour, for respective sampling frequencies of 8 Hz, 16 Hz and 32 Hz, respectively [52]. Related work on red foxes reports a battery life of five days, while a study monitoring the behaviour of cattle reports an effective battery life of 12 to 14 days [40, 62, 27, 47]. A study monitoring suckling events in northern Australian beef calves reports the effective on-duration of their device as 20 days due to data storage limits and battery capacity [58]. When the behaviour of housed dairy cows was monitored by transmitting raw accelerometer data to a receiver station, the battery was found to be depleted within two days [31]. Moreover, a recent study focusing on sensors used in sheep research, reviewed 71 peer-reviewed articles which included 82 independent experiments ranging in publication date from 1983 to 2017. The study includes most types of common sensors and tracking technologies and is not limited to behaviour classification studies using accelerometers. The authors found that most sensors are being deployed for short periods of time, whereafter it is removed and subsequently re-deployed a number of times within a single experiment. The authors continue that this is likely due to the battery and

Stellenbosch University https://scholar.sun.ac.za

Table 2.1: Summary of some offline automatic animal behaviour classification systems based on the statistical classification of tri-axial accelerometer data. The percentages shown were calculated by the various authors in different ways. Please refer to the studies to determine how the individual scores were calculated.

Animal	Behaviour	Score (%)	Source
Baboon	Forage, run, rest, stand, walk	88	[49]
Badger	Walking, trotting, snuffling, resting	83 - 92	[25, 30]
Cattle	Walking, standing, lying down, ruminating, grazing, drinking, searching, chewing, suckling, lameness	55 - 100	[31, 48, 56, 32, 57, 27, 21, 47, 50, 34, 35, 29, 42, 58, 59]
Cheetah	Feeding, mobile, stationary	83 - 94	[38]
Chipmunk	Still, locomotion, in-place movement, eating, grooming	73 - 90	[39]
Dog	Walk, trot, gallop, sleep, inactive, eat, drink, head-shake	86 - 95	[60]
Eagle	Flapping, soaring, sitting	87 - 92	[41]
Elephant	Feeding, bathing, walking, swaying	88	[37]
Goat	Resting, eating, walking	61 - 82	[46]
Hare	Running, feeding, vigilance	95 - 100	[53]
Otariids	Foraging, resting, travelling, grooming	47 - 94	[44]
Oystercatcher	Flying, foraging, handling prey, sitting, standing, walking	68 - 87	[33]
Polar Bear	Standing, sitting, lying down, swimming	50 - 99	[54]
Red Fox	Foraging, leaps, trotting	95	[40]
Sea Turtle	Breathing, grazing, resting, swimming	72 - 81	[51]
Shark	Chafing, burst swimming, head-shaking, resting, swimming	72 - 89	[28]
Sheep	Lying down, standing, walking, grazing, ruminating, resting, urinating	90 - 95	[61, 45, 52, 36, 55]
Vulture	Eating, lying down, active flight, passive flight, running, standing, preening	80 - 90	[26]

memory constraints of the sensor tags [63]. The above studies are only a few that provide information on the power consumption of their animal-attached sensor tags. It is clear that the duration of behavioural studies are limited by the lifespan of the tags which hinders long-term behavioural studies.

Stellenbosch University https://scholar.sun.ac.za

Currently automatic behaviour classification is performed offline as a post-processing step, after data collection. This is the case for all systems described above and listed in Table 2.1. Such configurations provide valuable, yet historical, information. However, the ability to analyse animal behaviour in real-time has great potential for applied ecological monitoring and wildlife conservation [64]. Some studies have considered the real-time monitoring of animal behaviour using other types of sensors. For example, Cangar et al. considered the automatic real-time monitoring of locomotion and posture of pregnant cows within a confined space prior to calving using top view cameras and online image analysis [65]. The system achieved high accuracies with an average of $85\,\%$ for standing and lying and $87\,\%$ for eating or drinking behaviour for eight cows during the last 24 hours before calving. However, video cameras consume a lot of energy, generate large amounts of data and are therefore not suitable for long-term deployment in remote locations. Furthermore, image or video processing is computationally demanding, which presents challenges for the successful application in real-time nature conservation systems. Satellite and cellular network enabled GPS transmitters, on the other hand, have a long track record of successfully monitoring the location of animals in real-time in remote locations [5, 64, 66]. Wall et al. demonstrate the importance of not only collecting, but also analysing GPS data in real-time using tracking collars on African Elephants [64]. It was possible to promptly detect and avoid an elephant bull from breaking through expensive electrical fencing into neighbouring subsistence farming land in order to forage in fields of maize. It was also possible to inform wildlife managers when elephant herds moved close to specific locations, such as the A2 highway on the Cape-to-Cairo route [64]. The real-time nature of the system allowed immediate alerts to be issued. The same author shows that the system was able to identify a wounded elephant using movement-rate analysis. This led to quick veterinary response and the recovery of the animal. Finally, the system was able to successfully detect elephant mortality by means of immobility analysis, which is key in their anti-poaching and real-time monitoring activities. Wall et al. go on to point out that, while accelerometer data in addition to GPS locations, can provide valuable information for real-time management applications, the increased volumes of data produced by the accelerometers introduces new challenges for both the tag's limited memory and battery capacities, as well as the wireless communication of this data to a control room. The author concludes that some form of onboard processing is inevitable. To date no study has considered the onboard embedded classification of tri-axial accelerometer data for real-time automatic animal behaviour classification.

Related studies have considered the real-time monitoring of human activities based on accelerometer data [67, 68, 69, 70, 71, 72, 73]. Although these studies demonstrate good performance, they are less constrained in terms of battery life and communication bandwidth

Stellenbosch University https://scholar.sun.ac.za

since users are co-operative and can be relied on to recharge the batteries and be within reception of standard communication technologies, such as cellular networks, which can be used to transfer data. For animal monitoring purposes these approaches are not feasible. Replacing batteries involves recapturing animals which is difficult, expensive and places additional stress on the animals. Furthermore, wild animals usually live in areas with limited or no communication coverage [74].

2.4 Energy-aware feature selection

Energy-aware feature selection is the process of balancing the energy-consumption of extracting the features employed by a classifier with the classification accuracy achieved with those features. Recent advances made in human related studies demonstrate the potential of this technique in the application of animal behaviour classification. For example, Ghasemzadeh et al. employed a graph model to select features based on the correlation between individual features and computing complexity of features [75]. They achieved a $30\,\%$ reduction in energy consumption while maintaining a $96.7\,\%$ classification accuracy. However, this study only considered effective feature selection and did not consider the energy cost of the classification technique employed. The k-nearest neighbour classification technique was used, which often achieves good classification results, but is an energy and memory intensive computational technique and not an optimal choice for resource constrained embedded systems.

Lauwereins et al. demonstrated the viability of a context and cost aware feature selection approach applied in a mobile voice activity detection application. The system was tasked with classifying between voice and noise using a decision tree classifier [76]. The authors utilised the decision tree's ability to sort features in order of decreasing information gain and used an information-gain-per-watt metric. They demonstrated a $70\,\%$ reduction in power consumption with no loss in classification performance.

Korpela et al. utilised a tree-structured feature extraction model that balances accuracy with computation- and transmission-costs of performing both human activity and gesture recognition based on accelerometer data at the same time. The authors adapted a standard decision tree algorithm to incorporate energy consumption into the algorithm to achieve a balance between energy cost and classification accuracy. The employed algorithm dynamically determines which features to extract, based on decision thresholds which form the tree nodes. For example, their model extracts a single low-cost feature utilised at the top of the tree. If the feature value is below a threshold it implies the need for high cost features and the raw data buffer is transmitted to a smartphone where the gesture is classified using a k-NN

Stellenbosch University https://scholar.sun.ac.za

classifier. Alternatively, if the feature is above the threshold value, further features are extracted and evaluated at different nodes in the tree to classify the activity. In one example the proposed model reduced the energy consumption by $50\,\%$ while maintaining a recognition accuracy of $95.5\,\%$ [77].

In a related human activity recognition study, Cui et al. demonstrate how wavelet coefficient features can be used to reduce the energy cost of feature computation while increasing the classification accuracy relative to a set of classical features [78].

Saeedi et al. focused on developing an energy-aware sensor localisation classifier used to determine the location of human wearable sensors a priori to activity recognition. The technique considered the trade-off between classification accuracy and power consumption of node localisation, based on features extracted from both accelerometers and gyroscopes. A greedy feature selection approach was used to evaluate both the importance and energy cost of different features presented to a k-NN classifier. The technique achieved accuracies between $73.15\,\%$ and $99.85\,\%$ while resulting in energy savings ranging form $88\,\%$ to $99.59\,\%$ [79].

Padmanabhan et al. developed a real-time embedded human emotion classifier able to distinguish between valence and arousal using a linear discriminant analysis classifier. In their application four different operational modes were defined based on energy budget constraints. The authors utilised a greedy energy-aware feature selection approach based on an objective function, which weighs the trade-off between emotion analysis and the available energy. For each of the modes an optimal set of features were selected. In its lowest-power mode the technique resulted in a 7.7 times increase in battery life, while achieving classification accuracies of $75\,\%$ and $70\,\%$ for arousal and valence, respectively. The trade-off in energy consumption and model performance are apparent when comparing the latter results with the classification accuracies achieved, $95\,\%$ and $89\,\%$ for the same two classes, by the unconstrained system [80].

Wang et al. utilised a low-power feature selection technique to reduce the energy consumption of an accelerometer and barometer-based fall detection system used for monitoring elderly people. A backward wrapper method was used to perform feature selection which reduces the energy cost of feature extraction while performance is kept within a predefined $10\,\%$ margin of error. The procedure selected four out of ten possible features. The system achieved power savings of $73.3\,\%$ while limiting the error rate of a binary classification decision tree to $7.1\,\%$ [81].

Stellenbosch University https://scholar.sun.ac.za

Although energy-aware feature selection has been done in human-related studies with great success, we are not aware of similar studies employed for animal-behaviour classification. Furthermore, the above-mentioned studies only considered the energy consumption of feature extraction, but not the energy cost of the specific classification techniques employed. In this work a technique is proposed where the energy consumption of both the feature extraction and classification steps are balanced with the achievable classification accuracies for animal borne sensor applications.

2.5 Conclusion

The study of animal behaviour by means of animal attached sensors is important and commonplace in many fields of research. However, automatic behaviour classification is typically performed offline using field collected data and observations. Most studies are short-lived as a result of the limited battery- and memory capacity of these devices. It is strongly believed that performing embedded behaviour classification in real-time and on the animal will result in major power and memory savings mainly attributed to the reduced volume of information that needs to be transmitted, and/or stored, to achieve real-time updates. Current literature supports this notion and suggests that some form of onboard processing is inevitable. Furthermore, we anticipate that selecting the embedded system based on an energy-aware feature- and model-selection technique can greatly reduce the energy consumption of animal-attached sensors while maintaining good model performance. Although various studies have successfully trained and utilised offline automatic behaviour classification systems to accurately organise large datasets into different behavioural classes, we are not aware of a real-time automatic animal behaviour classification system. Furthermore, the design and evaluation of such a system which is optimised both in terms of energy-consumption and model performance is still lacking.

Stellenbosch University https://scholar.sun.ac.za

Chapter 3

Classification Techniques

This chapter provides a brief description of each classification technique evaluated for our automatic behaviour classification system. The following eight techniques were considered: linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), naive Bayes (NB), logistic regression (LR), decision trees (DT), random forests (RF), k-nearest neighbours (k-NN) and linear support vector machines (LSVM). For a more detailed description of each classification technique, we refer the reader to the work in [82, 83, 84].

3.1 Bayes classifier

First a probabilistic classifier derived from probability theory called the Bayes classifier will be considered, since many other probabilistic classifiers are variations of the Bayes classifier. Bayes' theorem is given by Equation 3.1, where $\mathbf{y}$ is the class and $\mathbf{x}$ an observation.

$$P(\mathbf{y}|\mathbf{x}) = \frac{P(\mathbf{x}|\mathbf{y})P(\mathbf{y})}{P(\mathbf{x})} \tag{3.1}$$

In Equation 3.1, the probability of $\mathbf{x}$ is non-zero, $P(\mathbf{x}) \neq 0$. $P(\mathbf{y}|\mathbf{x})$ is the conditional probability representing the likelihood of $\mathbf{y}$ given that $\mathbf{x}$ is observed. This is also called the posterior probability. $P(\mathbf{x}|\mathbf{y})$ is a conditional probability representing the likelihood of observing $\mathbf{x}$ given the class $\mathbf{y}$. This is also called the class conditional probability. The marginal probabilities $P(\mathbf{x})$ and $P(\mathbf{y})$ are the probability of observing $\mathbf{x}$ and $\mathbf{y}$ independent of a knowledge of the other. $P(\mathbf{y})$ is typically referred to as the prior probability while $P(\mathbf{x})$ is sometimes referred to as the evidence [83]. For a simple binary classification problem with $\mathbf{y} \in \mathcal{Y} = \{0, 1\}$ we can write Equation 3.1 as

$$P(\mathbf{y} = 1|\mathbf{x} = x) = \frac{P(\mathbf{x} = x|\mathbf{y} = 1)P(\mathbf{y} = 1)}{P(\mathbf{x} = x)} \tag{3.2}$$

Stellenbosch University https://scholar.sun.ac.za

where

$$P(\mathbf{x} = x) = P(\mathbf{x} = x|\mathbf{y} = 1)P(\mathbf{y} = 1) + P(\mathbf{x} = x|\mathbf{y} = 0)P(\mathbf{y} = 0). \tag{3.3}$$

Using Equation 3.2 we can evaluate the posterior probability of each class and assign a new sample x as belonging to either the class $\mathbf{y} = 1$ or $\mathbf{y} = 0$ using a hypothesis function,

$$h(\mathbf{x}) = \begin{cases} 1 & \text{if} \quad P(\mathbf{y} = 1|\mathbf{x} = x) > P(\mathbf{y} = 0|\mathbf{x} = x) \\ 0 & \text{Otherwise.} \end{cases} \tag{3.4}$$

From the hypothesis function we see that the decision boundary separating the two classes is given by all the points that satisfy $P(\mathbf{y} = 1|\mathbf{x} = x) = P(\mathbf{y} = 0|\mathbf{x} = x)$. For the multi-class problem where $\mathbf{y}$ takes on more than two values, $\mathbf{y} \in \mathcal{Y} = \{1, 2, ..., K_{tot}\}$, we expand Equation 3.1 to calculate the posterior probability for each class as shown in Equation 3.5, where $f_k(\mathbf{x}) = P(\mathbf{x}|k)$ is the class-conditional probability of $\mathbf{x}$ given class k, $\pi_k = P(\mathbf{y} = k)$ the prior probability of class k and $\sum_{y\in\mathcal{Y}} f_y(\mathbf{x})\pi_y = P(\mathbf{x})$ the marginal probability of $\mathbf{x}$. Multi-class classification is done by evaluating a hypothesis function for each class and assigning the new sample to the class that results in the maximum posterior probability.

$$P(\mathbf{y} = k|\mathbf{x}) = \frac{f_k(\mathbf{x})\pi_k}{\sum_{y\in\mathcal{Y}} f_y(\mathbf{x})\pi_y}. \tag{3.5}$$

The hypothesis function is given by

$$h(\mathbf{x}) = \underset{k}{\operatorname{argmax}}\, P(\mathbf{y} = k|\mathbf{x}). \tag{3.6}$$

3.2 Linear discriminant analysis

Linear discriminant analysis performs classification using Bayes rule, presented in Equation 3.5, with two additional assumptions. First, that the class conditional distributions are multivariate Gaussian such that

$$f_k(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{d}{2}}|\boldsymbol{\Sigma}_k|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^{\mathrm{T}}\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)}. \tag{3.7}$$

In Equation 3.7, $\boldsymbol{\Sigma}_k$ are the class covariance matrices and $\boldsymbol{\mu}_k$ the class means. The second assumption is that the covariance matrices of all classes are the same $\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}$. Substituting the two assumptions into Equation 3.5 provides the expression for the LDA classifier. LDA

Stellenbosch University https://scholar.sun.ac.za

separates classes using linear decision boundaries. For the two-class case this can be shown by solving $P(\mathbf{y} = 1|\mathbf{x} = x) = P(\mathbf{y} = 0|\mathbf{x} = x)$, using Equation 3.5:

$$\begin{aligned} \frac{f_1(\mathbf{x})\pi_1}{\sum_{y\in\mathcal{Y}} f_y(\mathbf{x})\pi_y} &= \frac{f_0(\mathbf{x})\pi_0}{\sum_{y\in\mathcal{Y}} f_y(\mathbf{x})\pi_y} \\ f_1(\mathbf{x})\pi_1 &= f_0(\mathbf{x})\pi_0. \end{aligned} \tag{3.8}$$

Substituting multivariate Gaussian distributions for $f_0(\mathbf{x})$ and $f_1(\mathbf{x})$ it follows that

$$\frac{\pi_1}{(2\pi)^{\frac{d}{2}}|\boldsymbol{\Sigma}_1|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_1)^T\boldsymbol{\Sigma}_1^{-1}(\mathbf{x}-\boldsymbol{\mu}_1)} = \frac{\pi_0}{(2\pi)^{\frac{d}{2}}|\boldsymbol{\Sigma}_0|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_0)^T\boldsymbol{\Sigma}_0^{-1}(\mathbf{x}-\boldsymbol{\mu}_0)} \tag{3.9}$$

and when assuming $\boldsymbol{\Sigma}_0 = \boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}$, Equation 3.9 becomes

$$\begin{aligned} \frac{\pi_1}{(2\pi)^{\frac{d}{2}}|\boldsymbol{\Sigma}|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_1)^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_1)} &= \frac{\pi_0}{(2\pi)^{\frac{d}{2}}|\boldsymbol{\Sigma}|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_0)^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_0)} \\ \pi_1 e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_1)^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_1)} &= \pi_0 e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_0)^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_0)} \\ -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_1)^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_1) + \log(\pi_1) &= -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_0)^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_0) + \log(\pi_0). \end{aligned}$$

By expanding the quadratic parts of the equation we see,

$$\begin{aligned} -\frac{1}{2}\mathbf{x}^T\boldsymbol{\Sigma}^{-1}\mathbf{x} - \frac{1}{2}\boldsymbol{\mu}_1^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 + \mathbf{x}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 + \frac{1}{2}\mathbf{x}^T\boldsymbol{\Sigma}^{-1}\mathbf{x} \\ + \frac{1}{2}\boldsymbol{\mu}_0^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_0 - \mathbf{x}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_0 + \log(\frac{\pi_1}{\pi_0}) = 0 \end{aligned}$$

$$-\frac{1}{2}\boldsymbol{\mu}_1^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 + \mathbf{x}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 + \frac{1}{2}\boldsymbol{\mu}_0^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_0 - \mathbf{x}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_0 + \log(\frac{\pi_1}{\pi_0}) = 0$$

$$\mathbf{x}^T(\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 - \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_0) + \frac{1}{2}(\boldsymbol{\mu}_0^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1) + \log(\frac{\pi_1}{\pi_0}) = 0, \tag{3.10}$$

$$\text{which is in the form,} \qquad \mathbf{x}^T\mathbf{v} + b = 0. \tag{3.11}$$

From Equations 3.10 and 3.11 we see that LDA has a linear decision boundary as a result of the two initial assumptions. In the multi-class case it can be shown [84] that the Bayes classifier for LDA becomes

Stellenbosch University https://scholar.sun.ac.za

$$h(\mathbf{x}) = \underset{k}{\operatorname{argmax}}\, \delta_k(\mathbf{x}) \tag{3.12}$$

where

$$\delta_k(\mathbf{x}) = \mathbf{x}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_k - \frac{1}{2} \boldsymbol{\mu}_k^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_k + \log(\pi_k). \tag{3.13}$$

Classification is achieved by selecting the class that maximises Equation 3.13.

3.3 Quadratic discriminant analysis

Quadratic discriminant analysis is a more general form of LDA where the first assumption, that the class conditional distributions are multivariate Gaussian, is maintained. However, the second assumption, that the covariance matrices between classes are the same, is relaxed. Using the same approach as for LDA, it can be shown [84] that the decision boundary of QDA is quadratic. The multi-class Bayes classifier becomes

$$h(\mathbf{x}) = \underset{k}{\operatorname{argmax}}\, \delta_k(\mathbf{x}) \tag{3.14}$$

where

$$\delta_k(\mathbf{x}) = -\frac{1}{2} \log(|\boldsymbol{\Sigma}_k|) - \frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\mathbf{x} - \boldsymbol{\mu}_k) + \log(\pi_k). \tag{3.15}$$

Classification is again achieved by evaluating Equation 3.15 for each class and selecting the class which results in die maximum value of $\delta_k(\mathbf{x})$.

3.4 Naive Bayes

Naive Bayes classifiers are based on Bayes rule, given by Equation 3.5, with the naive assumption that features are conditionally independent given the class label. This model is called naive since the features will in practise generally not be independent in this way [84]. However, even when the naive assumption is not met, the classifier often produces good results and has the advantage of being relatively immune to overfitting. For convenience, we repeat the expression of Bayes rule here

$$P(k|\mathbf{x}) = \frac{f_k(\mathbf{x})\pi_k}{\sum_{y \in \mathcal{Y}} f_y(\mathbf{x})\pi_y}.$$

If we assume independence of features given the class label, the class-conditional probabilities $f_k(\mathbf{x})$ become,

Stellenbosch University https://scholar.sun.ac.za

$$P(x_i|\mathbf{y}=k,x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_d)=P(x_i|\mathbf{y}=k), \tag{3.16}$$

for all dimensions i, yielding the naive Bayes expression

$$P(\mathbf{y}=k\mid\mathbf{x})=\frac{\pi_k\prod_{i=1}^{d}f_k(x_i)}{\sum_{y\in\mathcal{Y}}f_y(x_i)\pi_y}. \tag{3.17}$$

Since the denominator in Equation 3.17 is constant, we can write

$$P(\mathbf{y}=k\mid\mathbf{x})\propto\pi_k\prod_{i=1}^{d}f_k(x_i) \tag{3.18}$$

and hence classification is achieved by evaluating

$$h(\mathbf{x})=\underset{k}{\operatorname{argmax}}\,\pi_k\prod_{i=1}^{d}f_k(x_i). \tag{3.19}$$

In this work we use the Gaussian naive Bayes model, which assumes that the class conditional distributions to be multivariate Gaussian, as shown in Equation 3.7.

3.5 Logistic regression

Logistic regression is a generalised linear model used for binary classification. With logistic regression the input values are combined linearly using weights, whereafter, a logistic (or sigmoid) function is applied to produce an output between zero and one indicating the probability of a single trial [84, 83]. Multi-class classification is achieved by means of a linear machine which divides the feature space into k decision regions $\mathcal{R}_1,\mathcal{R}_2,...,\mathcal{R}_k$ using k linear discriminant functions

$$\delta_k(\mathbf{x})=\frac{e^{\mathbf{v}_k^{\mathrm{T}}\mathbf{x}+v_{k0}}}{1+e^{\mathbf{v}_k^{\mathrm{T}}\mathbf{x}+v_{k0}}}, \tag{3.20}$$

where the vectors $\mathbf{v}_k$ and the intercept terms v_{k0} indicate the decision boundaries between the k classes. The maximum discriminant indicates the region within which $\mathbf{x}$ falls. Accordingly, we assign $\mathbf{x}$ to class $k=i$ when $\delta_i(\mathbf{x})>\delta_j(\mathbf{x})$ for all $j\neq i$ [82]. The hypothesis function of logistic regression is therefore given as

$$h(\mathbf{x})=\underset{k}{\operatorname{argmax}}\,\delta_k(\mathbf{x}). \tag{3.21}$$

Stellenbosch University https://scholar.sun.ac.za

3.6 Decision trees

The goal of a decision tree classifier is to create a model that can be used to predict the most likely class label of an unseen sample by means of a succession of binary decision rules inferred from prior data. These decision rules can be represented as a tree where each node corresponds to an attribute and each leaf to a class label. The tree structure recursively partitions the input space with the goal of defining a local model in each resulting region [84]. Consider the example shown in Figure 3.1a. At the top of the tree, the first node asks whether x_1 is less than some threshold value θ_1. If true, we are in the left quadrant of space $\mathcal{R}_1$, as shown in Figure 3.1b. If false, we ask whether x_2 is less than θ_2. If false, we are in the top right region of space $\mathcal{R}_2$. If true we determine, as a final step, whether x_1 is greater than or equal θ_3 which results in the regions $\mathcal{R}_4$ and $\mathcal{R}_3$ for the true and false conditions, respectively. The maximum tree depth is a hyper-parameter that can be set to limit the tree growth and therefore avoid very deep trees.

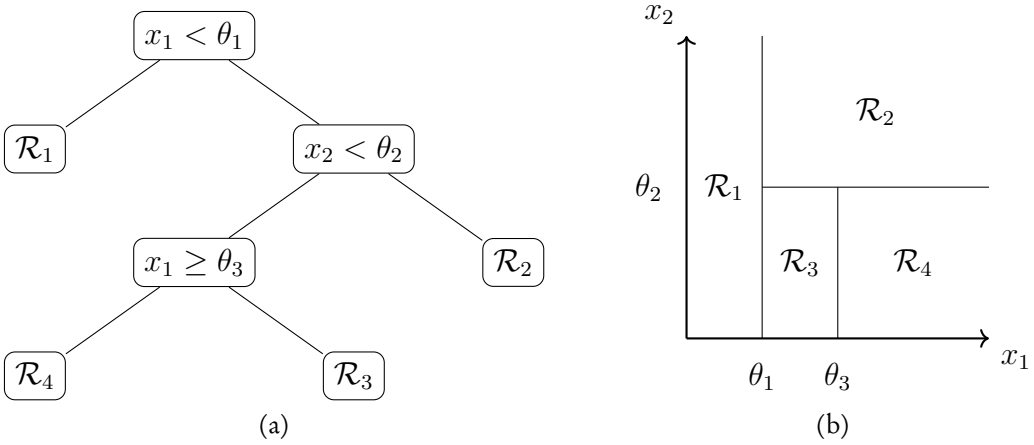


Figure 3.1: (a) Binary Decision tree representation with two input variables and four decision regions. (b) The two-dimensional representation of the binary decision tree.

3.7 Random forests

A random forest classifier evaluates the mode of the predicted output classes determined using multiple decision tree classifiers, each trained on a different random partition of the same training set. A random forest could, for example, train and combine the output of 10 different decision tree classifiers. Training on random data partitions and with different feature vector configurations reduces model over-fitting. The underlying individual decision tree classifiers are identical to those described in Section 3.6. The number of trees in the forest is a hyper-parameter of the classifier.

Stellenbosch University https://scholar.sun.ac.za

3.8 K-nearest neighbours

K-nearest neighbours is a method for classifying data based on the closest training examples within a feature space [84]. K-NN is an instance-based (or memory-based) learning algorithm sometimes also referred to as a lazy-learning algorithm, as it delays the generalization process until classification is performed [85]. Lazy-learning algorithms therefore require less computation during the training phase, but more during the classification process. K-NN stores all training examples in memory and calculates the relative distance of an unseen feature vector to all the training examples in the set, during classification at runtime. This can be achieved with different distance metrics, but typically the Euclidean distance is employed,

$$D(\mathbf{t}^{(i)}, \mathbf{x}) = \sqrt{\sum_{j=1}^{d} |\mathbf{t}_j^{(i)} - \mathbf{x}_j|^2} \qquad \text{for} \quad i = 1, 2...N_s. \tag{3.22}$$

In Equation 3.22, $\mathbf{t}^{(i)}$ denote the N_s feature vectors in the training set, $\mathbf{x}$ is the vector to be classified and d is the dimension of the feature vectors with j indicating a specific dimension. Once the relative distance to each training example has been calculated, the k-nearest (or closest) neighbours are determined after which $\mathbf{x}$ is classified as the most frequent class label among the k-neighbours using the expression

$$P(k|\mathbf{x}, K_{nn}) = \frac{1}{K_{nn}} \sum_{i \in N_K(\mathbf{x})} I(k_i = k) \tag{3.23}$$

where $N_K(\mathbf{x})$ denotes the indexes of the K_{nn} points nearest to $\mathbf{x}$ and I is an indicator function used to count the number of data points in the current class. The number of nearest points to consider is a hyper-parameter of the model.

3.9 Linear support vector machines

Support vector machines are non-probabilistic classifiers used for binary classification. A linear support vector machine separates data using a linear decision boundary, denoted by $\mathbf{v}^{\mathrm{T}} \cdot \mathbf{x} + b$. Objects lying on the positive side of the line are classified as belonging to one class and objects on the other side of the line as belonging to the other. Using class labels of {1,-1} rather than {1,0}, the decision function is given as

$$k = \text{sign}(\mathbf{v}^{\mathrm{T}} \cdot \mathbf{x} + b). \tag{3.24}$$

Training a LSVM involves choosing the values of $\mathbf{v}$ and b while utilising the concept of a margin that is chosen, based on the training data, in such a way that the distance to the closest data point is maximised for both sides of the decision boundary. This concept is illustrated in

Stellenbosch University https://scholar.sun.ac.za

Figure 3.2. From the figure we see that $\mathbf{v}$ is a vector perpendicular to the decision boundary and γ denotes the margin. Maximising the margin intuitively positions the decision boundary as far as possible form both the positive and negative samples. It can be shown that $\gamma = \frac{1}{\|\mathbf{v}\|}$ and that a procedure exists to maximise γ [83]. Although this is a binary classifier, it can easily be extended to perform multi-class classification. For example, a one-versus-rest technique can be employed [83].

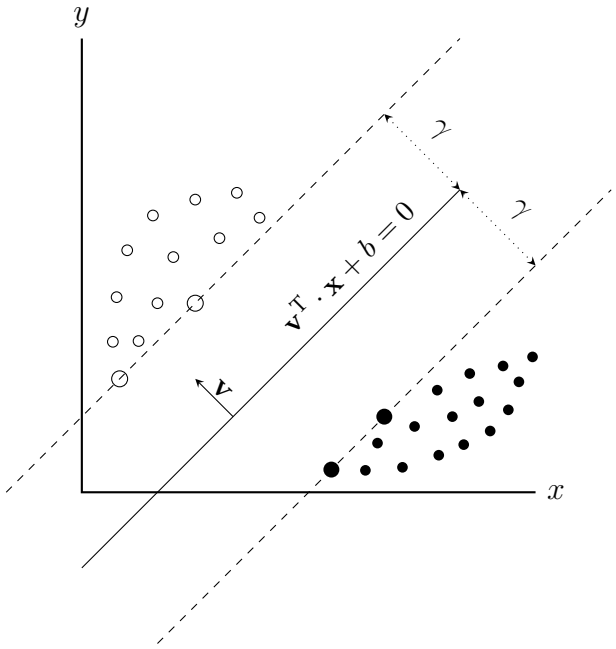


Figure 3.2: Linear support vector machine decision boundary and margins.

3.10 Model performance

Given a statistical classification model and a sample that is classified, there are four possible outcomes. If the sample is positive and it is classified as positive, we say it is a true positive (T_p). However, if it is classified as negative while it is in fact positive we say it is a false negative (F_n). If the sample is negative and it is classified as negative, we say it is a true negative (T_n). However, if it is classified as positive while it is in fact negative, we say it is a false positive (F_p) [86]. The resulting outcomes can be combined in a confusion matrix that represents the four possible outcomes. In a confusion matrix values along the major diagonal represent correct predictions, while off-diagonal terms represent errors [86]. Four common metrics typically calculated from a confusion matrix are the classification accuracy (A_{CC}), precision (P_r), recall (R_e) and the F1 score ($F1$). The respective definitions of these metrics are given by Equations 3.25, 3.26, 3.27 and 3.28. Model accuracy is given by the total number of correctly predicted samples divided by the total number of predictions,

Stellenbosch University https://scholar.sun.ac.za

$$A_{CC} = \frac{T_P + T_N}{T_P + F_P + F_N + T_N}. \tag{3.25}$$

Precision is the number of true positives divided by the sum of the number of true positives and false positives,

$$P_r = \frac{T_p}{T_p + F_p}. \tag{3.26}$$

The precision provides intuition on the classification performance by evaluating how many labels of all the labels assigned to a specific class, actually belongs to that class. Recall is the number of true positives divided by the sum of the number of true positives and false negatives,

$$R_e = \frac{T_p}{T_p + F_n}. \tag{3.27}$$

The recall provides intuition on the classification performance by evaluating how many samples out of all the samples in a class, were correctly labelled. The $F1$ score is the harmonic mean (weighted average) of precision and recall which is a useful metric when you have uneven class distributions,

$$F1 = 2\frac{P_r \times R_e}{P_r + R_e}. \tag{3.28}$$

3.11 Conclusion

This chapter provided a summarised description of the classification techniques employed in our study. The eight techniques utilised for automatic behaviour classification are linear discriminant analysis, quadratic discriminant analysis, naive Bayes, logistic regression, decision trees, random forests, k-nearest neighbours and linear support vector machines. Some common metrics used to evaluate classification performance were also considered.

Stellenbosch University https://scholar.sun.ac.za

Chapter 4

Methods

This chapter provides a detailed hardware description of the animal-attached sensor tags used during this study for both raw data collection and for the real-time embedded behaviour classification system. Furthermore, the raw data collected during field work are described for both sheep and rhinoceros. The data pre-processing and balancing steps are subsequently discussed, whereafter, the feature extraction process and cross-validation framework is described. This chapter is based on our earlier work presented in [2].

4.1 Hardware

For this study specially designed low-power bio-telemetry tags for the real-time embedded behaviour classification system were utilised. The hardware design of these tags is extensively explained in our earlier work [87]. An overview of the most important hardware components utilised are provided here. Figure 4.1 shows an assembled bio-telemetry tag and points out the most important components.

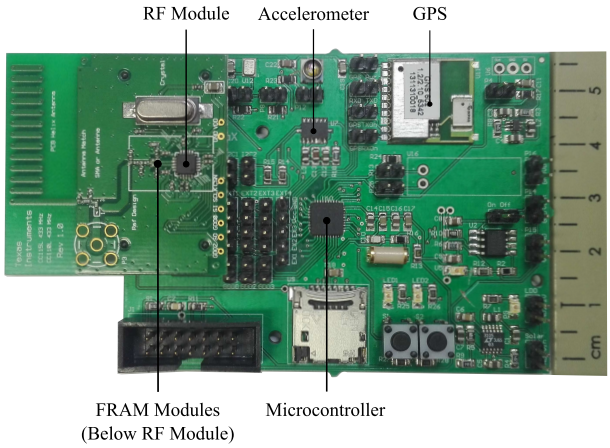


Figure 4.1: Assembled biotelemetry tag. The assembled printed circuit board (PCB) measures 100 mm x 60 mm x 12 mm and weighs 32 g.

Stellenbosch University https://scholar.sun.ac.za

The tri-axial accelerometer can be considered as the most important sensor of the device as it provides detailed information on the acceleration profiles of different animal behaviours. A GPS module provides accurate spatial- and temporal information which aid in determining exactly where the animals are located and how they move about over time. Ferro-electric non-volatile random access memory (FRAM) modules provide fast memory access and were used during data sampling, whereas, a micro-SD card was used to store larger amounts of information. The wireless radio frequency transceiver enables real-time data delivery by means of wireless communication to a base station. At the heart of the system a low-power microcontroller unit (MCU) coordinates the data sampling, processing, storage and transmission. The hardware block diagram is shown in Figure 4.2 and depicts the inter connectivity between the various hardware components.

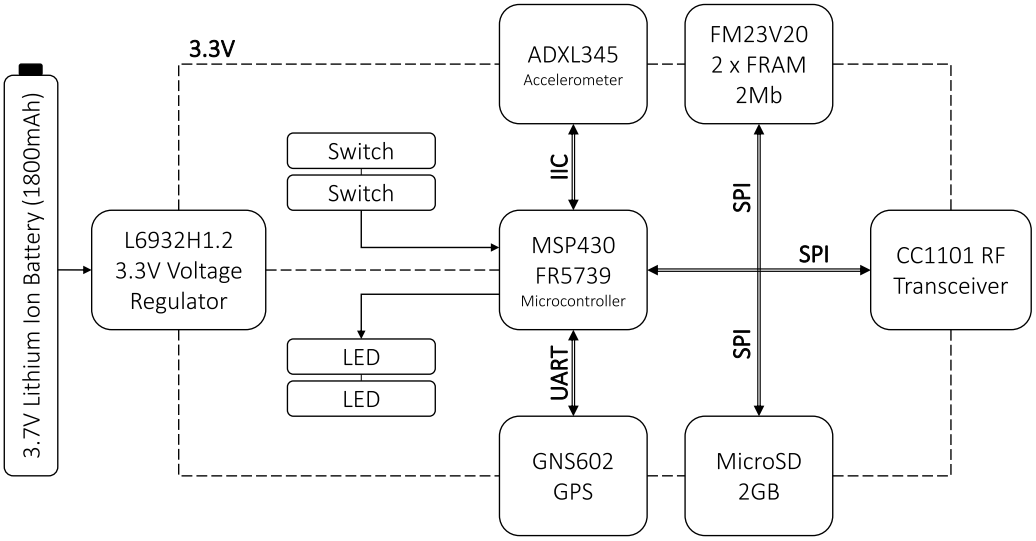


Figure 4.2: Block diagram of the hardware design of the biotelemetry tags. A MSP430 ultra-low power microcontroller communicates with the accelerometer using IIC, the GPS using UART and the FRAM, Micro-SD card and radio frequency transceiver using SPI. Two tactile switches and two light emitting diodes were included for software functionality selection and indication purposes, respectively.

More specifically, the design utilised a L6932H1.2 ultra low drop-out voltage regulator (STMicroelectronics, Geneva, Switzerland), a MSP430FR5739 low-power mixed signal microcontroller (Texas Instruments, Dallas, Texas, US), a GNS602 GPS receiver (Global Navigation Systems, North Shields, UK), two FM25V20 FRAM storage modules of 2 Mb each (Cypress Semiconductor, San Jose, California, US), a 2 GB Micro-SD card (Kingston, München, Germany) and an ADXL345 tri-axial accelerometer (Analog Devices, Norwood, Massachusetts, US) with high resolution (13-bit) measurement at up to $\pm 16g$ ($1g = 9.81\,\mathrm{m/s^2}$). A low-power CC1101 sub-1 GHz radio frequency transceiver (Texas Instruments, Dallas, Texas, US) allows wireless data communication at 433 MHz. The tags are powered by 3.7 V 1800 mAh lithium-ion batteries.

Stellenbosch University https://scholar.sun.ac.za

The tags were packaged using two different collar designs for sheep and rhinoceros respectively. The sheep collars had a total weight of 281 g, which included an 146 mm x 80 mm x 65 mm polycarbonate casing (126 g), the physical electronics (32 g), a battery (33 g) and a belt (90 g) used to fit the collar around the necks of sheep. The rhinoceros collar had a total weight of 371 g which included a 117 mm x 77 mm x 38 mm thick durable and waterproof casing (165 g) made of polyoxymethylene (acetal), the physical electronics (32 g), two batteries (66 g) and two belts (108 g) used to fit the collar around the legs of rhinoceros.

The base stations utilised Raspberri Pi B+ units and CC1101 radio frequency shields as explained in our earlier work [87]. Figure 4.3 shows the base station design. Communication from the tags to the base station was implemented using a star network typology. The current implementation used a basic carrier sense multiple access with collision avoidance (CSMA/CA) communication protocol with exponential back-off time and without message acknowledgements from the base station. Field tests indicated that a communication range of roughly 1 km can be achieved using this configuration [87]. The communication protocol optimisation and base station hardware optimisation, are not within the scope of this dissertation.

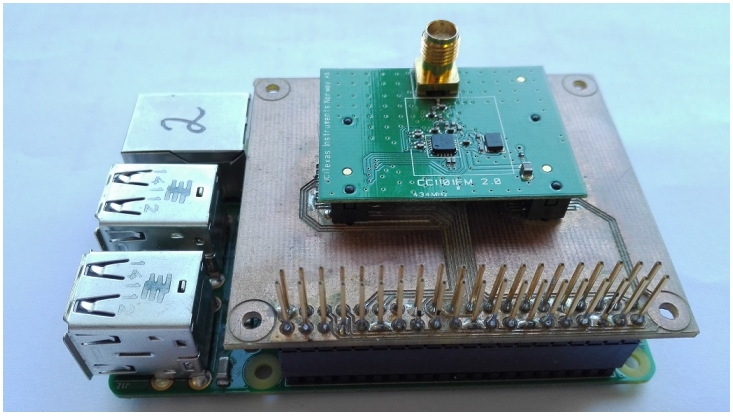


Figure 4.3: Raspberri Pi with CC1101 RF Shield.

4.2 Data collection

Datasets for both sheep and rhinoceros were complied as part of this study. The collected data were manually labelled with the observed behaviour, to provide ground truth. The resulting annotated corpus was used to develop and evaluate statistical classifiers.

Stellenbosch University https://scholar.sun.ac.za

4.2.1 Sheep dataset

For the sheep, raw data were collected on a farm in Carnarvon, Northern Cape, South Africa. The tags were fitted around the necks of the sheep using collars as shown in Figure 4.4. The x-, y- and z-axes were associated with left-right, up-down and forward-backward movement of the sheep, respectively.

(a) (b)

Figure 4.4: Biotelemetry collars fitted around the necks of sheep. The accelerometer x-, y- and z-axes are associated with left-right, up-down and forward-backward movement of the sheep, respectively.

The data collection was performed during three separate days. On each of these three mornings, collars were attached to five individual sheep (Dohne Merino) randomly selected from a group of approximately twenty animals. The sheep were then led to a larger camp and left undisturbed for the duration of the day, during which the tags continuously logged the acceleration. Five common types of behaviour were identified: lying down, standing, walking, running and grazing. The behaviour of the sheep was manually documented with timestamps to serve as ground truth. At the end of each day, the sheep were collected, the collars removed and the recorded acceleration signals downloaded.

Figure 4.5 shows examples of the raw measurements for the different recorded sheep behaviours. Differences in the acceleration waveforms are apparent between the five behaviours. For both walking and running we note significant signal movement with an extent of periodicity in the measurements. The three more-stationary behaviours, standing, grazing and lying down result in different dc-components for each of the accelerometer axes and show very little movement.

Stellenbosch University https://scholar.sun.ac.za

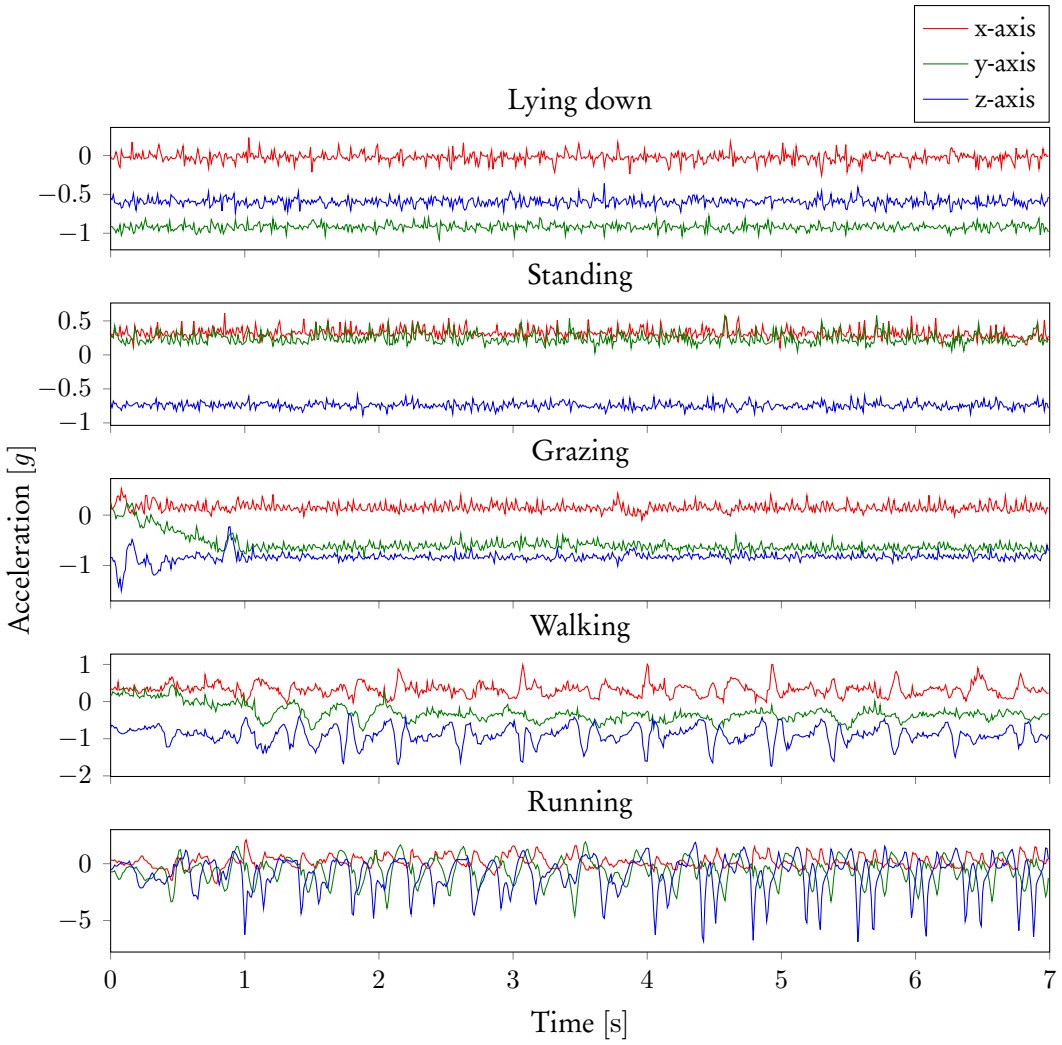


Figure 4.5: Raw acceleration measurements from the sheep dataset. Typical acceleration measurements for the five identified sheep behaviours sampled at 100 Hz. The x (red), y (green) and z (blue) accelerometer axes are shown. Differences in the acceleration waveforms are apparent between the five behaviours. However, lying down and grazing behaviour have similar acceleration signals.

4.2.2 Rhinoceros dataset

The rhinoceros dataset was collected at a private sanctuary. Three rhinoceros were available for data collection, one male and one female White Rhinoceros (Ceratotherium simum) and one male Black Rhinoceros (Diceros bicornis). The animals were held in a camp approximately 100 ha in area and moved around freely. The collars were fitted around the left back leg of the rhinoceros as shown in Figure 4.6. The x-axis was associated with up-down movement, the y-axis with forward-backward movement and the z-axis with left-right movement.

Data collection took place over a period of one week. The rhinoceros were fed daily and this provided the opportunity to fit the collars. Once the collars were fitted, the rhinoceros

Stellenbosch University https://scholar.sun.ac.za

(a) (b)

Figure 4.6: Biotelemetry collar fitted around the back leg of a rhinoceros. The accelerometer x, y and z axes are associated with the up-down, forward-backward and left-right movement, respectively.

were left undisturbed to roam freely in their camp. The collars were removed during later feeding sessions and the data downloaded, after which the collars were redeployed. The rhinoceros were monitored each day at the feeding sessions during which timestamped video recordings were made. Four common behaviours were identified: lying down, standing, walking and running. The video recordings allowed accurate labelling of the gathered data. Figure 4.7 shows examples of the raw measurements for the different behaviours.

4.3 Data labelling

All collected data for both sheep and rhinoceros were segmented and labelled manually with the corresponding behaviour. A summary of the raw sheep and rhinoceros datasets is given in Table 4.1. Data collection from rhinoceros was logistically and practically challenging and as a result the data corpus gathered was smaller than that for sheep. Running behaviour was not often observed for rhinoceros. Grazing behaviour could not be detected because the tags were attached to the back leg for the rhinoceros, and not to the neck as for sheep.

Table 4.1: Raw unbalanced datasets collected for each behaviour class for sheep and for rhinoceros (hours:minutes).

	Lying down	Standing	Walking	Running	Grazing	Total
Sheep	1:29	0:55	6:08	1:12	6:27	16:11
Rhinoceros	2:05	3:46	0:48	0:07	–	6:46

Stellenbosch University https://scholar.sun.ac.za

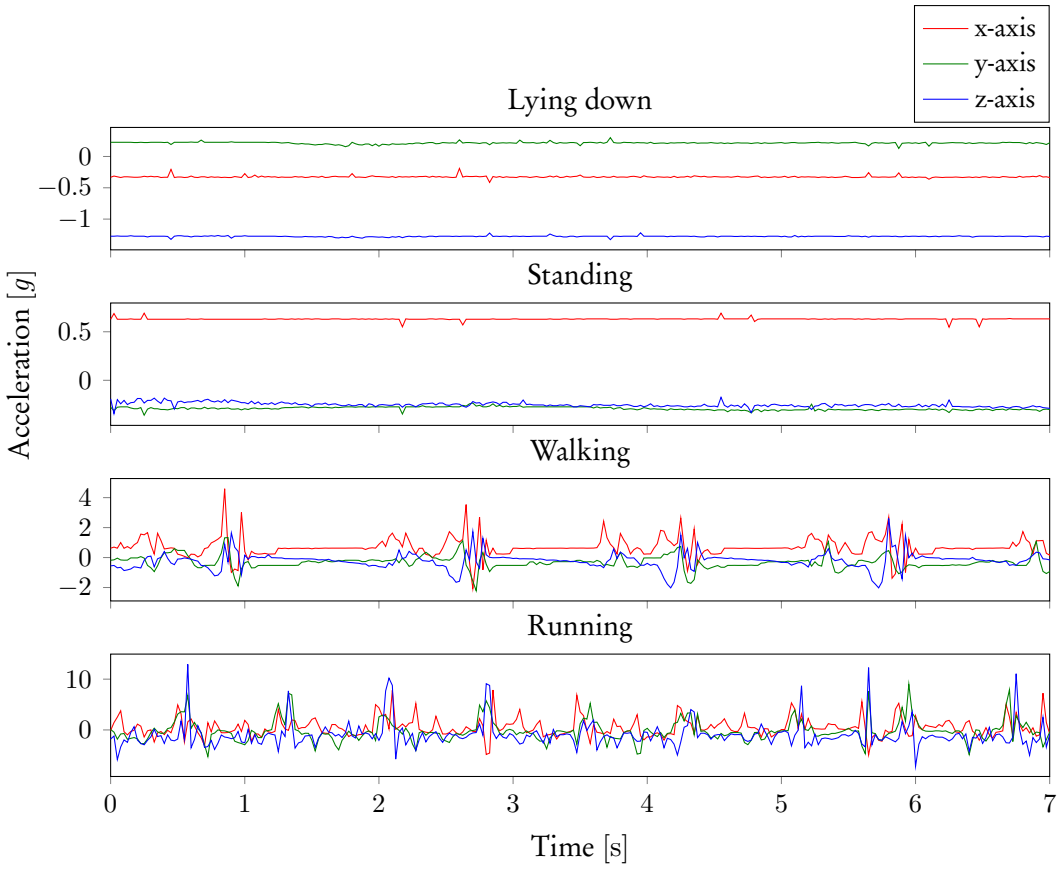


Figure 4.7: Raw acceleration measurements from the rhinoceros dataset. Typical acceleration measurements for the four identified rhinoceros behaviours sampled at 40 Hz. The x (red), y (green) and z (blue) accelerometer axes are shown. Differences in the acceleration waveforms are apparent between the four behaviours.

4.4 Data pre-processing

Data collection was performed at different sampling frequencies of 100 Hz for sheep and 40 Hz for rhinoceros, respectively. Spectral analysis for both the sheep and rhinoceros datasets showed very little spectral information above 10 Hz as shown in Figure 4.8. The figure shows the Fast Fourier transform (FFT) of all running data for the y-axis in the sheep dataset. The raw measurements were therefore low-pass filtered using an 8th-order Butterworth filter with a cut-off frequency of 10 Hz before down-sampling the signal to 20 Hz. Resampling the data allows the system to function at a lower sampling frequency (20 Hz) in future deployments. Frames spanning 128 measurements were subsequently extracted for sheep and rhinoceros. The extracted frames had a duration of 6.4 s and were allowed to overlap by 50 %. These frames were then used in the subsequent feature extraction process.

Stellenbosch University https://scholar.sun.ac.za

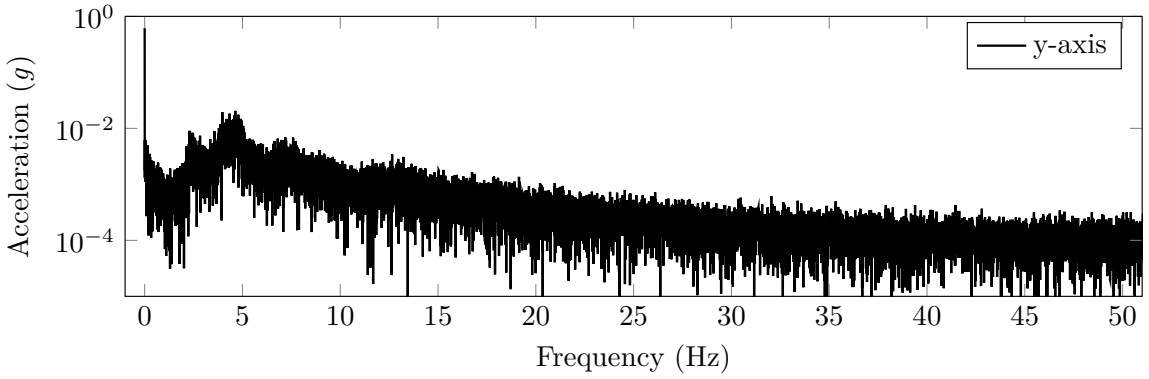


Figure 4.8: The Fast Fourier transform of all running data for the y-axis in the sheep dataset.

4.5 Feature extraction

The 13 features extracted are listed in Table 4.2. For each feature, except average signal magnitude, three values were computed, one for each accelerometer axis, leading to a total of 64 features per frame. The same features were extracted for both the sheep and the rhinoceros datasets.

Table 4.2: Features extracted from the compiled datasets. Each frame consists of N sequential samples, and here $\mathbf{x}$ denotes a vector of these samples for each accelerometer axis, x, y and z. The FFT of $\mathbf{x}$ is denoted by $\hat{\mathbf{x}}$ and the normalised power spectrum of $\mathbf{x}$ by $P_n(\mathbf{x})$. Cross-correlation and the mean distance are calculated for each axis pair (x,y), (x,z), and (y,z). N_b is the number of samples in each bin and B denotes the number of bins. All features except average signal magnitude provide three values: one per axis.

Feature	**Notation**	**Equation**
Average signal magnitude	asm	$\frac{1}{N}\sum_{i=1}^{N}\sqrt{x_i^2+y_i^2+z_i^2}$
Maximum value	max()	$\max(\mathbf{x})$
Minimum value	min()	$\min(\mathbf{x})$
Mean ($\bar{x}$)	mean()	$\frac{1}{N}\sum_{i=1}^{N}x_i$
Standard deviation (σ_x)	std()	$\sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i-\bar{x})^2}$
Variance	var()	σ_x^2
Skewness	skew()	$\frac{\frac{1}{N}\sum_{i=1}^{N}(x_i-\bar{x})^3}{\sigma_x^3}$
Kurtosis	kurt()	$\frac{\frac{1}{N}\sum_{i=1}^{N}(x_i-\bar{x})^4}{\sigma_x^4}$
Energy	en()	$\frac{1}{N}\sum_{i=1}^{N}\lvert\hat{\mathbf{x}}_\mathbf{i}\rvert^2$
Spectral entropy	pse()	$\sum_{i=1}^{N}P_n(x_i)\log\frac{1}{P_n(x_i)}$
Pairwise correlation between the axes	corr()	$\frac{cov(\mathbf{x},\mathbf{y})}{\sigma_x\sigma_y}$
Mean distance between the axes	dist()	$\frac{1}{N}\sum_{i=1}^{N}(\mathbf{x_i}-\mathbf{y_i})$
Energy in 1 Hz bins	$\text{bin}_B()$	$\frac{1}{N_b}\sum_{i=BN_b+1}^{BN_b+N_b}\lvert\hat{\mathbf{x}}_\mathbf{i}\rvert^2$ with $B=\{0,1,..,9\}$

Stellenbosch University https://scholar.sun.ac.za

The two datasets were balanced after feature extraction, in order to avoid possible bias between classes and animals. The sheep dataset was collected from 15 animals by randomly selecting five sheep from a heard of 20 sheep over three days, hence the identity of each animal can not be guaranteed. However, for this study we assume 15 individual animals. In the case of the rhinoceros dataset, collection took place from three distinct and identifiable animals over a period of one week. The two datasets were therefore balanced over behavioural classes as well as animals. This was achieved by limiting the data available for each class from each animal to that available for the least frequent class among all animals. The resulting dataset had an equal number of samples for each class and for each animal. We decided to omit the small amount of running data from the rhinoceros dataset for cross-animal testing purposes, since this behaviour was observed for only one animal. The total number of feature vectors available after balancing the datasets is given in Table 4.3.

Table 4.3: Balanced datasets, indicating the number of feature vectors extracted for each behaviour class for the sheep and rhinoceros data presented in Table 4.1.

	Lying	**Standing**	**Walking**	**Running**	**Grazing**	**Total**
Sheep	1008	1008	1008	1008	1008	5040
Rhinoceros	894	894	894	-	-	2682

We express each of the balanced datasets as a 5040 by 64 dimensional matrix $\mathbf{D}_S$ and a 2682 by 64 dimensional matrix $\mathbf{D}_R$, for sheep and rhinoceros, respectively. Each row in the matrices depicts a single training example of 64 features and each column depicts all the training examples of a single feature. The balanced datasets $\mathbf{D}_S$ and $\mathbf{D}_R$ were subsequently used in-order to train and validate various statistical classification models.

4.6 Classification model training and evaluation procedure

The model training and evaluating process is broadly described in Figure 4.9. From the figure we see that, as a first step, a balanced dataset needs to be selected for evaluation. We can select from either the sheep dataset $\mathbf{D}_S$ or the rhinoceros dataset $\mathbf{D}_R$. Both datasets have 64 available features but a different number of training examples. There are 5040 training examples in the sheep dataset and 2682 training examples in the rhinoceros dataset, as explained in Chapter 4.5. The next step involves selecting a specific classification model to evaluate. The available classifiers are listed in Chapter 3. Next, we employ a K-fold cross-validation procedure to coordinate the feature selection-, model training- and final model evaluation-processes. This ensures optimal feature selection and robust model performance. The details of each step are explained in the following sections.

Stellenbosch University https://scholar.sun.ac.za

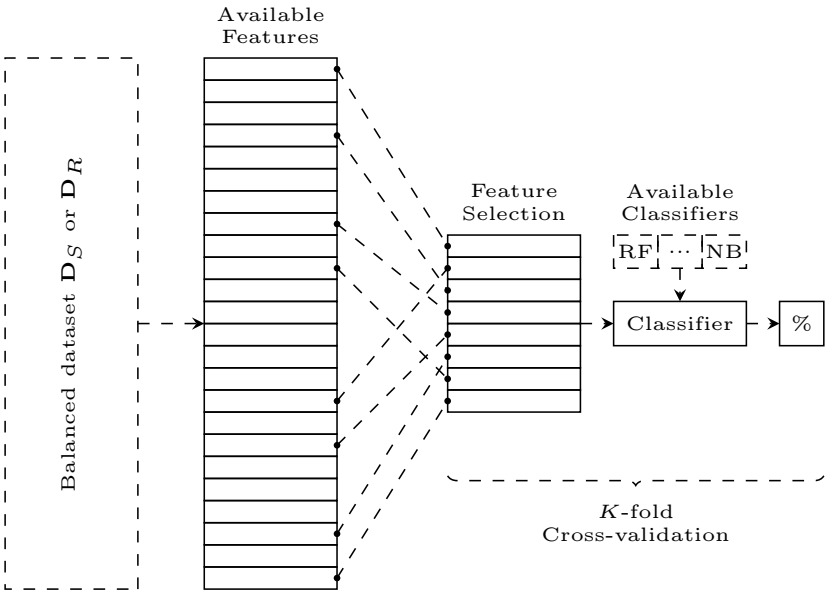


Figure 4.9: Overview of the statistical classification strategy employed.

4.7 Sequential forward feature selection

Feature selection is the proses of evaluating the effectiveness of various combinations of features to accurately predict class labels when used with statistical classifiers. Using all the available features does not necessarily result in the highest model performance and quite often result in sub-optimal performance. It is therefore important to evaluate different combinations of features to determine which combination results in the highest classification performance. One way to achieve this goal is to follow a brute-force approach and evaluate every single combination that can be achieved with the available features. However, the long execution times involved in evaluation these large numbers of feature combinations, makes this approach unattractive and therefore researchers employ different feature selection techniques to find optimal feature combinations suited for their applications. One such a feature selection approach is called sequential feature selection (SFS). SFS is a greedy feature selection algorithm which selects and adds features one by one with the goal of minimising the classification error with each addition [88]. SFS, in turn, selects a single feature from the total number of available features $d = 64$ whereafter a classifier is trained and evaluated using only the selected feature. Each of the remaining features are evaluated one by one, in the same manner. The feature that results in the highest classification accuracy is selected and kept as the best feature of the first round. In the next iteration, a single feature is selected from the 63 remaining features. However, this time the model training and evaluation it done using the combination of the best feature kept in the previous round and the newly selected feature. After considering all 63 features, the feature that results in the highest performance gain is kept in this round. This process is repeated d times until all 64 available feature are

Stellenbosch University https://scholar.sun.ac.za

arranged in order of importance. SFS therefore only considers 2080 different combinations when selecting from 64 features. Although SFS typically achieves great model performance, it is important to note that the global optimum can not be guaranteed, since SFS does not evaluate all possible feature combinations.

4.7.1 Grouped, nested and repeated K-fold cross-validation

Feature selection and model evaluation were carried out within a grouped, nested and repeated K-fold cross-validation framework to ensure model robustness [89]. The cross-validation procedure is illustrated in Figure 4.10 for $K = 5$ (5-fold cross-validation). The dataset $\mathbf{D}$, representing either $\mathbf{D}_S$ or $\mathbf{D}_R$, is initially split into K even-sized folds $\mathcal{T} = \{\mathbf{T}_1, \mathbf{T}_2, ..., \mathbf{T}_{K=5}\}$, based on the data collected from individual animals. Each fold $\mathbf{T}_i$ is a sub-matrix of the dataset $\mathbf{D}$ that includes all features (columns) but only a fraction of the training examples (rows). For each iteration of cross-validation, a single fold $\mathbf{T}_i$ is selected as a held-out set, while the remaining $K-1$ folds, $\mathbf{O}_{train}$, are used for model training and validation. For example, in Figure 4.10, $\mathbf{T}_1$ becomes the first held-out set while $\mathbf{T}_2$ through $\mathbf{T}_5$ are used for model training and validation. In subsequent iterations, $\mathbf{T}_2$, $\mathbf{T}_3$, $\mathbf{T}_4$ and $\mathbf{T}_5$ will in turn be used as held-out sets. The training and validation partition, $\mathbf{O}_{train}$, is further divided into $K-1$ folds by randomly selecting an equal number of samples (rows) for each fold. We denote this set of $K-1$ sub-matrices of $\mathbf{O}_{train}$ as $\mathcal{V} = \{\mathbf{V}_1, \mathbf{V}_2, ..., \mathbf{V}_{K-1}\}$. One of these folds $\mathbf{V}_j$ is chosen as a validation set, while the remainder form the training set, $\mathbf{I}_{train}$. In the example in Figure 4.10, $\mathbf{V}_1$ becomes the first validation-set while $\mathbf{V}_2$ to $\mathbf{V}_4$ are used for training. In subsequent iterations, $\mathbf{V}_2$, followed by $\mathbf{V}_3$, and finally, $\mathbf{V}_4$ will be used as validation-sets. Model hyper-parameter optimisation and feature selection is performed on each of these training- and validation-set combinations. By repeating this optimisation N_{in} times, we can select models based on the average validation-set performance, thus avoiding over-optimistic results [89]. After completing all inner-fold iterations, a merged list of optimal features is obtained and used to determine the test-set error $\mathbf{e}_{rri}$ associated with the held-out set $\mathbf{T}_i$. The average error over all held-out test-sets $\mathbf{T}_i$, is computed to obtain the final cross-validated model performance $\mathbf{e}_{rr}$. Each element in the vector $\mathbf{e}_{rr}$ represents the average cross-validated classification performance that can be achieved using a specific number of optimally arranged features. We define e_{rr} as the peak performance in $\mathbf{e}_{rr}$ which is achieved at the optimal number of features F_N, as shown in Equation 4.1. Since model training and validation is performed on data obtained from one subset of animals while the held-out sets $\mathbf{T}_i$ consists of data obtained from a different set of animals, overfitting individual animal characteristics is avoided and the evaluation of cross-animal performance is achieved.

$$e_{rr} = \mathbf{e}_{rr}^{(F_N)} \tag{4.1}$$

Stellenbosch University https://scholar.sun.ac.za

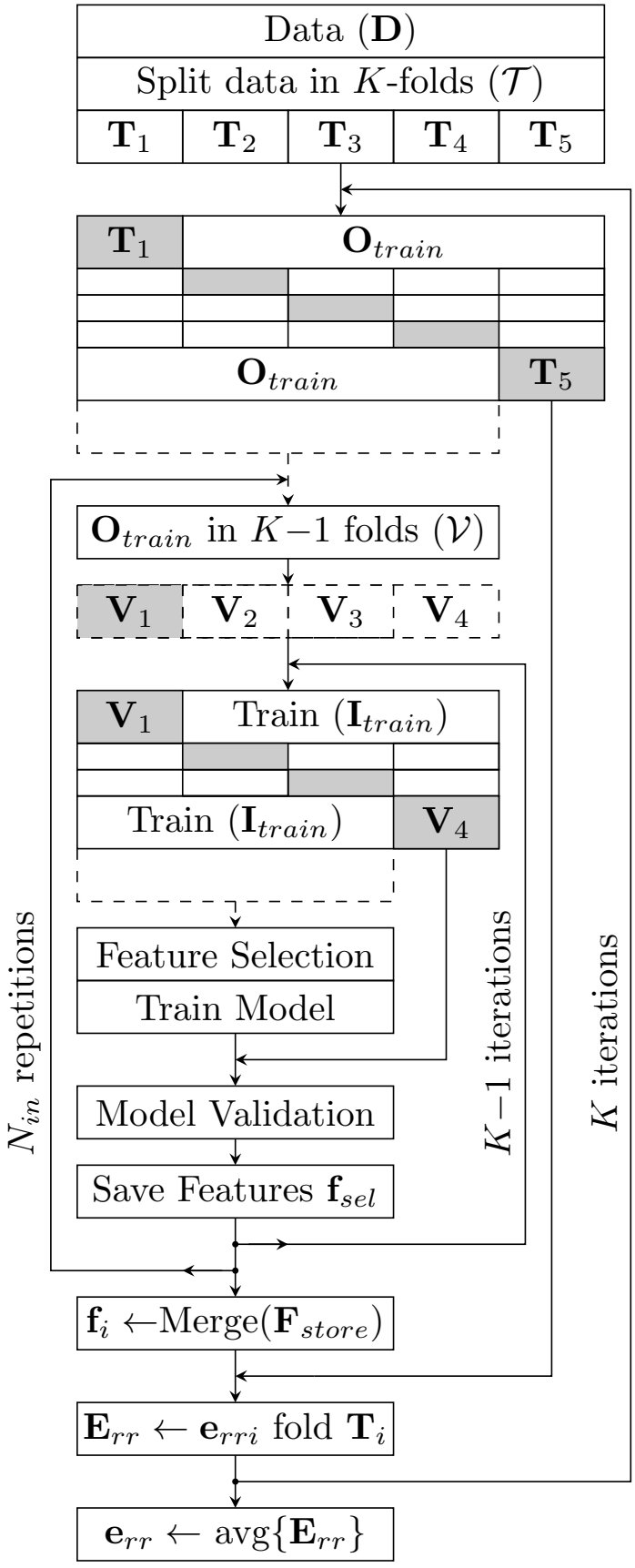


Figure 4.10: Grouped, nested and repeated K-fold cross-validation framework, illustrated for the case $K = 5$.

Stellenbosch University https://scholar.sun.ac.za

4.8 Conclusion

This chapter considered the hardware design of the animal-attached sensor tags used in this study for both raw data collection and for the real-time embedded behaviour classification system. Furthermore, it described the raw data collection, data pre-processing and feature extraction steps followed for both sheep and rhinoceros datasets. A description of the classification procedure, including the feature selection and cross-validation steps, were also described. In the next chapter the classification results achieved for various statistical classification models are presented.

Stellenbosch University https://scholar.sun.ac.za

Chapter 5

Automatic behaviour classification

This chapter considers the training and evaluation of various statistical classifiers for both the rhinoceros and sheep datasets. First, the details of the cross-validation algorithm employed and subsequently the classification accuracies achieved for both the rhinoceros and sheep datasets using the different classification models are provided. For each classification technique, four different performance metrics are presented. The best performing models are identified and further explored with their most informative features highlighted. This chapter is based on our earlier work as presented in [2].

5.0.1 K-fold cross-validation algorithm

The details of the algorithm used to perform feature selection, model validation and test-set evaluation within a grouped nested and repeated K-fold cross-validation framework is presented in Algorithm 1. The algorithm is executed for both datasets $\mathbf{D}_S$ and $\mathbf{D}_R$ described in Chapter 4 and for all the classification models $C(\alpha)$[1] described in Chapter 3. The sheep dataset is split into 15 outer-folds, since data were collected from 15 individual sheep. We express these folds as a set of sub-matrices of the dataset $\mathbf{D}_S$ denoted as $\mathcal{T} = \{\mathbf{T}_1, \mathbf{T}_2, ..., \mathbf{T}_{K=15}\}$. Similarly the rhinoceros data is split into three outer-folds with $\mathcal{T} = \{\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_{K=3}\}$. Since there are K-folds in $\mathcal{T}$ the outer loop of the cross-validation procedure will have K-iterations. For each of these iterations a single fold $\mathbf{T}_i$ is selected from $\mathcal{T}$, as shown in line four. The outer-fold training-set $\mathbf{O}_{train}$ is then defined as all the data samples in $\mathbf{D}$ excluding the samples in $\mathbf{T}_i$. The outer-fold test-set $\mathbf{O}_{test}$ is subsequently defined as only the data samples in fold $\mathbf{T}_i$. The test-set $\mathbf{O}_{test}$ is first excluded from all model training and validation procedures and later exclusively

[1]This work collectively refers to the hyper-parameters α of a specific classifier as $C(\alpha)$. For both random forest and decision tree classifiers the maximum tree depth is a hyper-parameter that can be set to limit the tree growth and therefore avoid very deep trees. This work considers tree depths of one to 10 and also evaluate models with no tree depth restrictions ($\alpha = \infty$). For the k-nearest neighbour classifier, the number of nearest points to consider is a hyper-parameter of the model and this work considers between one and 10 closest neighbours. For example, $C(\alpha) =$ DT(10) refers to a decision tree classifier with a maximum tree depth of 10 and k-NN(3) refers to a k-NN classifier with three nearest neighbours.

Stellenbosch University https://scholar.sun.ac.za

used to evaluate the test-set performance. The $\mathbf{O}_{train}$ data partition is subsequently used in the feature selection and model training steps, shown in lines to 34. For a total of $N_{in} = 10$ repetitions the data partition $\mathbf{O}_{train}$ is further divided into K-1 folds $\mathbf{V}_1, \mathbf{V}_2, ..., \mathbf{V}_{K-1}$ by randomly selecting an equal number of samples from $\mathbf{O}_{train}$ for each fold. We denote this set of sub-matrices of $\mathbf{O}_{train}$ as $\mathcal{V} = \{\mathbf{V}_1, \mathbf{V}_2, ..., \mathbf{V}_{K-1}\}$. For each of the K-1 iterations of the inner-loop, shown in line 10, a single fold $\mathbf{V}_j$ is selected from $\mathcal{V}$. The inner-fold training-set $\mathbf{I}_{train}$ is then defined as all the data samples in $\mathbf{O}_{train}$ excluding the samples in $\mathbf{V}_j$. The inner-fold validation-set $\mathbf{I}_{val}$ is subsequently defined as only the data samples in fold $\mathbf{V}_j$. Lines 13 to 32 describe the SFS process and shows how the classification error is evaluated to determine the best order of features. Line 21 specifically shows how a statistical model is trained using $\mathbf{I}_{train}$ with only a number of features selected. This trained model m' is subsequently used to validate the model's performance on the validation-set $\mathbf{I}_{val}$ with only a number of features selected, as shown in line 22. The validation-set performance is tracked and utilised to select the features with the aim of maximising the model performance at every step, as shown in lines 23 to 29. Subsequently, a feature vector $\mathbf{f}_{sel}$ is produced for each inner fold $\mathbf{V}_j$ and for every repetition N_{in}. These feature vectors are merged, using the average rank of each feature in each vector, to produce a single feature vector $\mathbf{f}_i$ for each test-set fold $\mathbf{T}_i$, as shown in line 36. The test-set error is subsequently determined by evaluating the trained model on the held-out data partition $\mathbf{O}_{test}$, as shown in lines 37 to 46. As a final step, the mean cross-validation error $\mathbf{e}_{rr}$ is calculated by averaging over all the individual results obtained from each test-set partition. We report the average rank of the top performing features, the optimal number of selected features F_N and the minimum cross-validation error $e_{rr} = \mathbf{e}_{rr}^{(F_N)}$ evaluated at the optimal number of features for each dataset and classification model evaluated.

Stellenbosch University https://scholar.sun.ac.za

ALGORITHM 1: Grouped, Nested and Repeated K-fold Cross-Validation

procedure CROSS VALIDATION($\mathbf{D}$, $C(\alpha)$)
    Divide the dataset $\mathbf{D}$ into K-folds: $\mathcal{T} = \{\mathbf{T}_1, \mathbf{T}_2, ..., \mathbf{T}_K\}$
    $\mathbf{E}_{rr} \leftarrow \varnothing$ # place holder for test-set classification error
    for $\mathbf{T}_i$ in $\mathcal{T}$ **do**
        Define $\mathbf{O}_{train}$ as $\mathbf{D}$ without the fold $\mathbf{T}_i$
        Define $\mathbf{O}_{test}$ as the fold $\mathbf{T}_i$
        $\mathbf{F}_{store} \leftarrow \varnothing$ # place holder for selected feature vectors
        for $N_{in} \leftarrow 1$ to 10 **do**
            Divide $\mathbf{O}_{train}$ randomly into (K-1)-folds: $\mathcal{V} = \{\mathbf{V}_1, \mathbf{V}_2, ..., \mathbf{V}_{K-1}\}$
            for $\mathbf{V}_j$ in $\mathcal{V}$ **do**
                Define $\mathbf{I}_{train}$ as $\mathbf{O}_{train}$ without the fold $\mathbf{V}_j$
                Define $\mathbf{I}_{val}$ as the fold $\mathbf{V}_j$
                # Sequential feature selection starts here
                $\mathbf{f}_{sel} \leftarrow \varnothing$ # place holder for selected features
                $\mathbf{f}_{eval} \leftarrow \mathbf{f}_{all} = \{f_1, f_2, ..., f_d\}$ # All available features
                for $n \leftarrow 1$ to d **do**
                    $e_{min} \leftarrow 100\,\%$ # some large value
                    for f_{eval} in $\mathbf{f}_{eval}$ **do** # consider each feature in turn
                        Define $\mathbf{I}'_{train}$ as $\mathbf{I}_{train}$ selecting only the features, $\mathbf{f}_{sel}$ + f_{eval}
                        Define $\mathbf{I}'_{val}$ as $\mathbf{I}_{val}$ selecting only the features, $\mathbf{f}_{sel}$ + f_{eval}
                        Build a statistical model $m' = m(\mathbf{I}'_{train}, C(\alpha))$
                        Apply m' to $\mathbf{I}'_{val}$ and determine the classification error: e_{val}
                        if $e_{val} < e_{min}$ **then**
                            $e_{min} \leftarrow e_{val}$
                            $f_{best} \leftarrow f_{eval}$
                        end if
                    end for
                    Append f_{best} to $\mathbf{f}_{sel}$
                    Remove f_{best} from $\mathbf{f}_{eval}$
                end for
                Append $\mathbf{f}_{sel}$ to $\mathbf{F}_{store}$ # store optimal set of features for iteration $\mathbf{V}_j$
                # Sequential feature selection ends here
            end for
        end for
        # This results in $(N_{in}) * (K-1)$ stored feature vectors, hence the matrix $\mathbf{F}_{store}$ has dimensions of $(N_{in})(K-1) \times 64$

Stellenbosch University https://scholar.sun.ac.za

Merge $\mathbf{F}_{store}$ to produce a single optimal feature vector $\mathbf{f}_i = \{\hat{f}_1, \hat{f}_2, ..., \hat{f}_64\}$ for each fold $\mathbf{T}_i$
$\mathbf{f}_{test} \leftarrow \varnothing$
$\mathbf{e}_{rri} \leftarrow \varnothing$
for f_i in $\mathbf{f}_i$ **do**
    Append f_i to $\mathbf{f}_{test}$
    Define $\mathbf{O}'_{train}$ as $\mathbf{O}_{train}$ selecting only the features $\mathbf{f}_{test}$
    Define $\mathbf{O}'_{test}$ as $\mathbf{O}_{test}$ selecting only the features $\mathbf{f}_{\mathbf{test}}$
    Build a statistical model $m' = m(\mathbf{O}'_{train}, C(\alpha))$
    Apply m' to $\mathbf{O}'_{test}$ and append the error to $\mathbf{e}_{rri}$
end for
Append $\mathbf{e}_{rri}$ to $\mathbf{E}_{rr}$ # store the test-set results of fold $\mathbf{T}_i$
end for
Average the values in $\mathbf{E}_{rr}$ to produce the overall average cross-validated error $\mathbf{e}_{rr}$
$\mathbf{e}_{rr}$ represents the average cross-validated classification error achieved using different number of features.
$F_N \leftarrow \text{argmin}_i(\mathbf{e}_{rr}^{(i)})$ # $i = 1, 2, ..., d$
$e_{rr} = \mathbf{e}_{rr}^{(F_N)}$
e_{rr} is the lowest cross-validated classification error achieved.
end procedure

Stellenbosch University https://scholar.sun.ac.za

5.1 Results for the rhinoceros dataset

A summary of the results obtained for the rhinoceros dataset is provided in Table 5.1. The table shows the accuracy A_{CC} achieved for each classification model employed $C(\alpha)$ at the optimal hyper-parameter setting α and the optimal number of features F_N.

Table 5.1: Mean cross-validated accuracies achieved at the optimal number of features and optimal hyper-parameter setting for each classification technique evaluated on the rhinoceros dataset.

Classifier $C(\alpha)$	F_N	A_{CC} (%)
LSVM	11	99.61
LDA	6	99.51
LR	10	99.51
k-NN (8)	7	99.41
RF (3)	47	96.91
DT (2)	38	96.83
NB	1	92.78
QDA	1	92.78

From the table we see that all the classification techniques achieved high classification accuracies. The LSVM, with 11 selected features, achieved the highest classification accuracy of $99.61\,\%$. The other models achieved accuracies between $92.78\,\%$ and $99.51\,\%$ with the number of features ranging from one to 47. Further details for the best performing model is provided in Table 5.2. Table 5.2a shows the precision, recall and $F1$ scores achieved with the LSVM model for the individual classes. Table 5.2b shows the confusion matrix for the LSVM model. From the confusion matrix we see that a very small degree of confusion exists between classes. Table 5.2c lists the 10 most informative features selected, starting with the most important feature, the maximum value of the x-axis. Furthermore, Figure 5.1 shows the results obtained from the sequential feature selection process. In the figure the solid line indicates the mean cross-validated accuracy and the error bars depict the range of test-set accuracies achieved for a specific number of features. We see how the SFS process adds features in such a manner to maximise the classification accuracy afforded, by additional features at every step. The performance metrics of all the evaluated classification models are presented in a similar manner, in Tables 5.3 through Table 5.9.

Stellenbosch University https://scholar.sun.ac.za

Table 5.2: The performance of a linear support vector machine, utilising 11 optimal features, evaluated on the rhinoceros dataset.

(a) The per-class precision, recall and $F1$ score.

P_r	R_e	$F1$	Class	F_N
0.994	0.997	0.996	Lie	11
0.999	0.996	0.997	Stand	11
0.997	0.998	0.997	Walk	11

(b) The per-class confusion-matrix. The rows depict true labels and the columns depict predicted labels.

	Lie	Stand	Walk
Lie	888	0	3
Stand	4	887	0
Walk	1	1	889

(c) Top 10 selected features.

max(x), mean(x), asm(), min(x), kurt(x), var(x), std(x), en(x), pse(x), max(y)

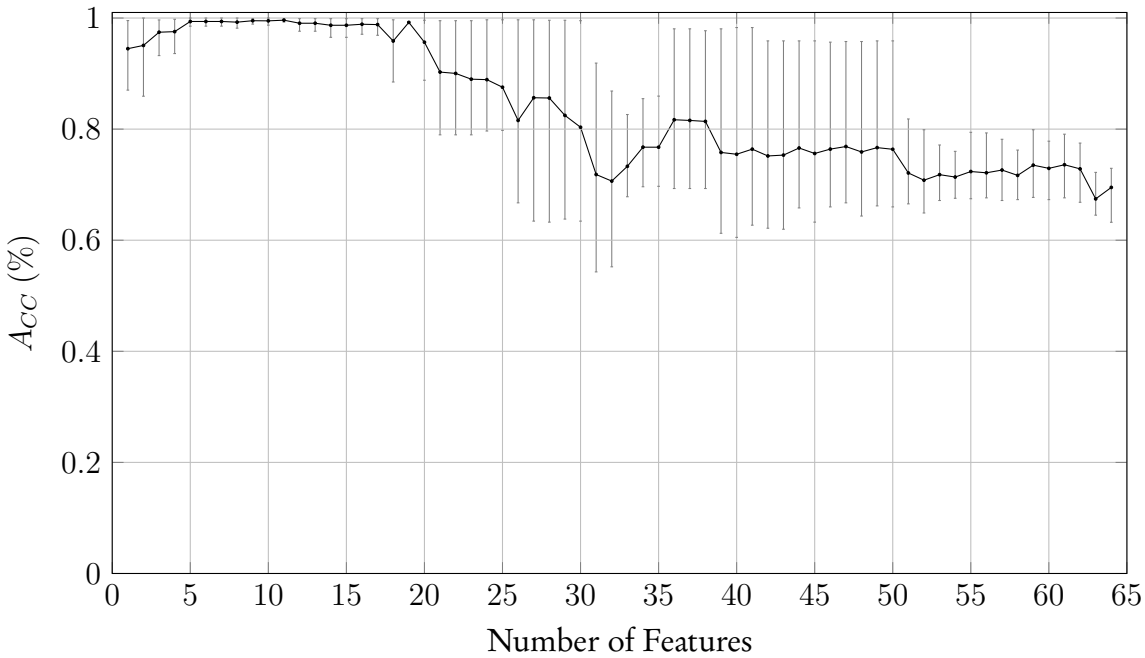


Figure 5.1: The mean cross-validated test-set accuracies achieved utilising optimal features and a linear support vector machine evaluated on the rhinoceros dataset. The error-bars indicate the maximum and minimum test-set accuracies achieved for the specific number of features.

Stellenbosch University https://scholar.sun.ac.za

Table 5.3: The performance of linear discriminant analysis, utilising six optimal features, evaluated on the rhinoceros dataset.

(a) The per-class precision, recall and $F1$ score.

P_r	R_e	$F1$	Class	F_N
0.998	0.994	0.996	Lie	6
0.988	0.997	0.992	Stand	6
0.998	0.992	0.995	Walk	6

(b) The per-class confusion-matrix. The rows depict true labels and the columns depict predicted labels.

	Lie	Stand	Walk
Lie	886	4	1
Stand	2	888	1
Walk	0	7	884

(c) Top 10 selected features.

max(x), kurt(x), var(x), min(y), min(x), pse(y), mean(x), max(y), en(x), mean(y)

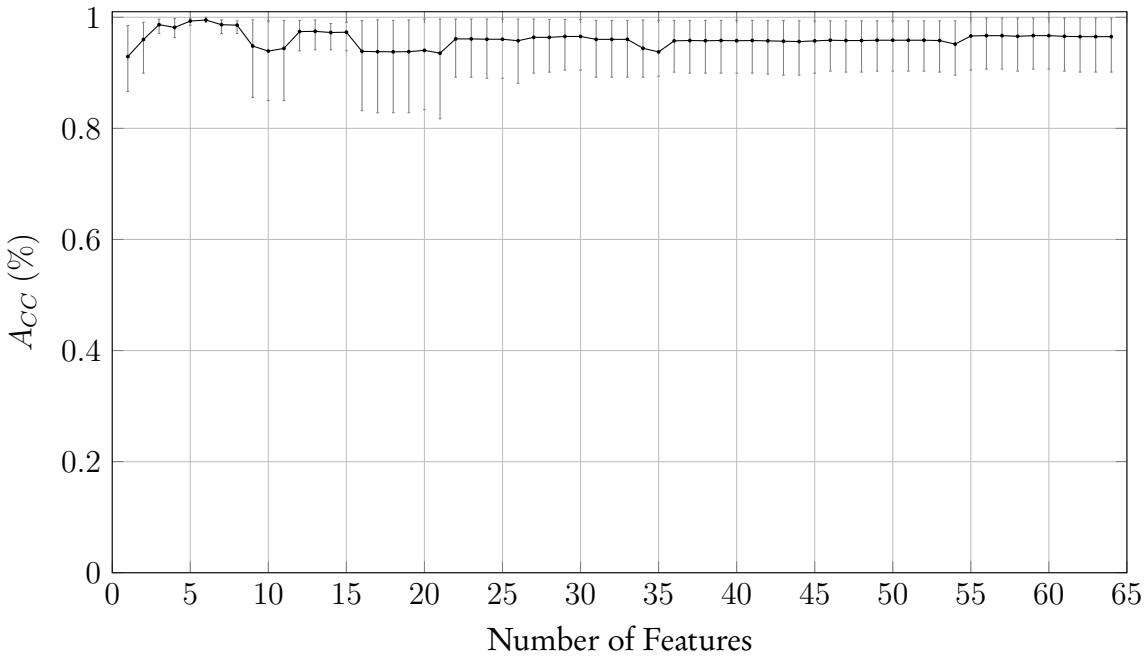


Figure 5.2: The mean cross-validated test-set accuracies achieved utilising optimal features and linear discriminant analysis evaluated on the rhinoceros dataset. The error-bars indicate the maximum and minimum test-set accuracies achieved for the specific number of features.

Table 5.3 shows the cross-validated results of LDA using six optimal features. From the table we see that LDA is able to discriminate between the three behaviours with high- precision, recall and $F1$ score. The confusion-matrix shows very little confusion between classes. A peak classification accuracy of $99.51\,\%$ was achieved using the following 6 features: max(x), kurt(x), var(x), min(y), min(x) and pse(y). However, Figure 5.2 shows that by using only the first three features will also result in good model performance.

Stellenbosch University https://scholar.sun.ac.za

Table 5.4: The performance of logistic regression, utilising 10 optimal features, evaluated on the rhinoceros dataset.

(a) The per-class precision, recall and $F1$ score.

P_r	R_e	$F1$	Class	F_N
0.993	0.997	0.995	Lie	10
0.993	0.994	0.994	Stand	10
0.998	0.993	0.996	Walk	10

(b) The per-class confusion-matrix. The rows depict true labels and the columns depict predicted labels.

	Lie	Stand	Walk
Lie	888	1	2
Stand	5	886	0
Walk	1	5	885

(c) Top 10 selected features.

max(x), mean(x), min(x), kurt(x), asm(), var(x), std(x), pse(x), max(y), min(y)

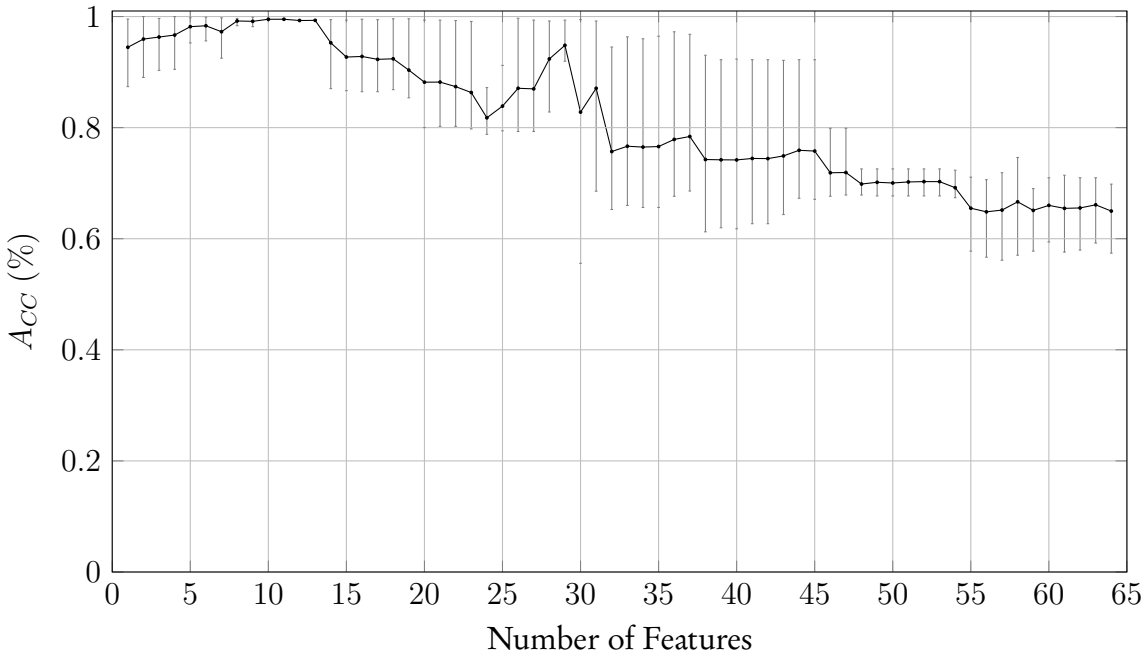


Figure 5.3: The mean cross-validated test-set accuracies achieved utilising optimal features and logistic regression evaluated on the rhinoceros dataset. The error-bars indicate the maximum and minimum test-set accuracies achieved for the specific number of features.

Table 5.4 shows the cross-validated results of LR using 10 optimal features. From the table we see that a LR classifier is able to distinguish between the three behaviours with high-precision, recall and $F1$ score. The confusion-matrix shows very little confusion between classes. A peak classification accuracy of $99.51\,\%$ was achieved using the 10 features listed in Table 5.4c. Figure 5.3 depicts that by using only the first five features will also result in good model performance.

Stellenbosch University https://scholar.sun.ac.za

Table 5.5: The performance of a k-nearest neighbour classifier (with 8-nearest neighbours), utilising seven optimal features, evaluated on the rhinoceros dataset.

(a) The per-class precision, recall and $F1$ score.

P_r	R_e	$F1$	Class	N_F
0.998	0.981	0.989	Lie	7
0.987	0.998	0.992	Stand	7
0.994	1.0	0.997	Walk	7

(b) The per-class confusion-matrix. The rows depict true labels and the columns depict predicted labels.

	Lie	Stand	Walk
Lie	874	12	5
Stand	2	889	0
Walk	0	0	891

(c) Top 10 selected features.

max(x), mean(x), asm(), var(x), std(x), max(y), min(x), var(y), mean(y), std(y)

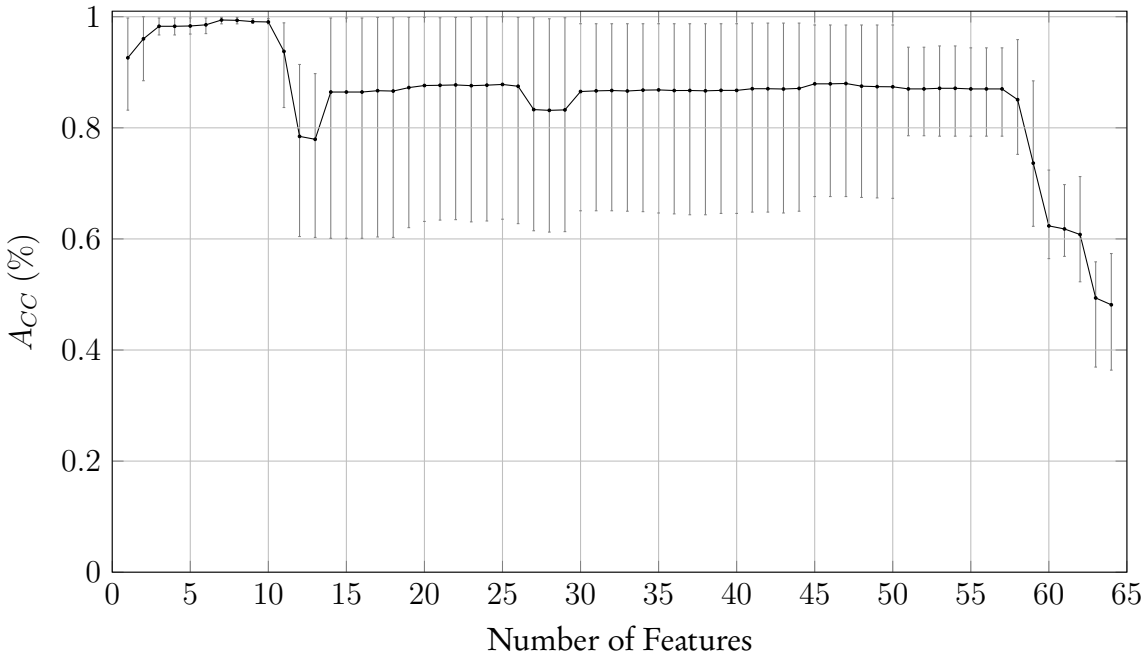


Figure 5.4: The mean cross-validated test-set accuracies achieved utilising optimal features for a k-nearest neighbour classifier with 8-nearest neighbours evaluated on the rhinoceros dataset. The error-bars indicate the maximum and minimum test-set accuracies achieved for the specific number of features.

Table 5.5 shows the cross-validated results of a k-NN classifier with 8-nearest neighbours using seven optimal features. From the table we see that the k-NN classifier is able to discriminate between the three behaviours with high- precision, recall and $F1$ score. The confusion-matrix shows very little confusion between classes and in particular, no confusion exists for walking behaviour. A peak classification accuracy of $99.41\,\%$ was achieved using the following 7 features: max(x), mean(x), asm(), var(x), std(x), max(y) and min(x). Figure 5.4 shows that by using only the first three features will also result in good model performance.

Stellenbosch University https://scholar.sun.ac.za

Table 5.6: The performance of a random forest classifier (with a maximum tree depth of three), utilising 47 optimal features, evaluated on the rhinoceros dataset.

(a) The per-class precision, recall and $F1$ score.

P_r	R_e	$F1$	Class	N_F
0.949	0.996	0.972	Lie	47
0.997	0.999	0.998	Stand	47
0.998	0.946	0.971	Walk	47

(b) The per-class confusion-matrix. The rows depict true labels and the columns depict predicted labels.

	Lie	Stand	Walk
Lie	887	2	2
Stand	1	890	0
Walk	47	1	843

(c) Top 10 selected features.

max(x), mean(x), asm(), var(x), std(x), skew(x), kurt(x), en(x), pse(x), min(y)

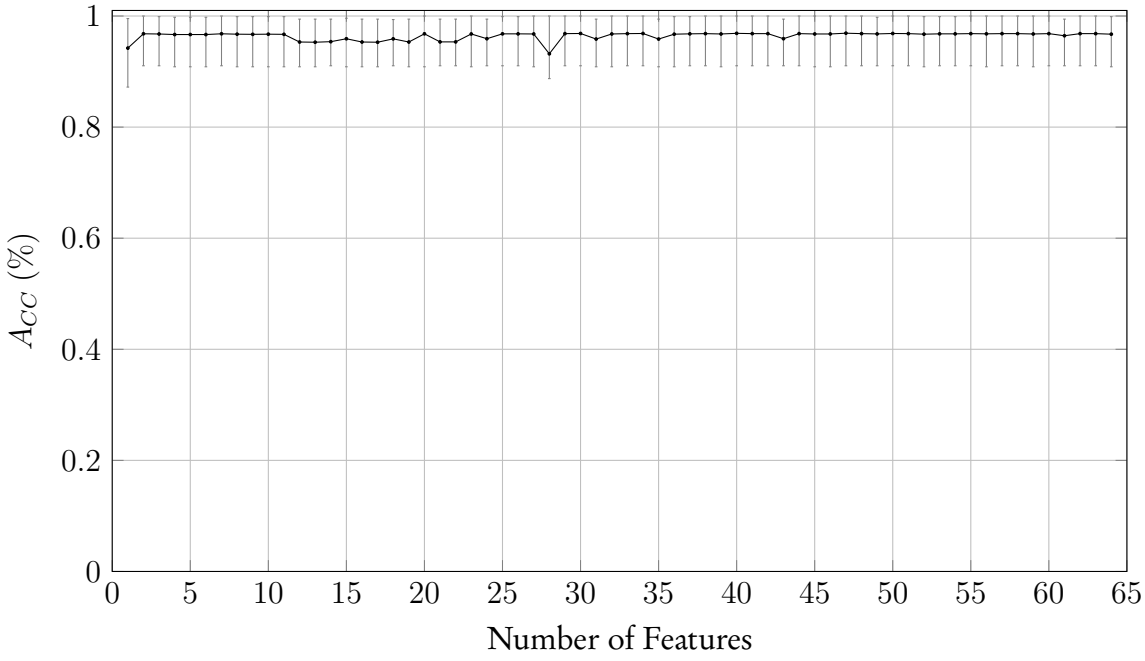


Figure 5.5: The mean cross-validated test-set accuracies achieved utilising optimal features and a random forest with a maximum tree-depth of three evaluated on the rhinoceros dataset. The error-bars indicate the maximum and minimum test-set accuracies achieved for the specific number of features.

Table 5.6 shows the cross-validated results of a RF classifier with a maximum tree-depth of three using 47 optimal features. From the table we see that the RF classifier is able to discriminate between the three behaviours with high- precision, recall and $F1$ score. The confusion-matrix shows very little confusion between classes. A peak classification accuracy of $96.91\,\%$ was achieved using 47 features, of which the first 10 is shown in Table 5.6c. However, from Figure 5.5 it is clear that similar results can be obtained with two and more features and specifically by using only the max(x) and mean(x) features.

Stellenbosch University https://scholar.sun.ac.za

Table 5.7: The performance of a decision tree classifier (with a maximum tree depth of two), utilising 38 optimal features, evaluated on the rhinoceros dataset.

(a) The per-class precision, recall and $F1$ score.

P_r	R_e	$F1$	Class	N_F
0.949	0.996	0.972	Lie	38
0.994	0.998	0.996	Stand	38
0.998	0.945	0.971	Walk	38

(b) The per-class confusion-matrix. The rows depict true labels and the columns depict predicted labels.

	Lie	Stand	Walk
Lie	887	3	1
Stand	1	889	1
Walk	47	2	842

(c) Top 10 selected features.

max(x), mean(x), asm(), var(x), std(x), skew(x), kurt(x), en(x), en(x), min(y)

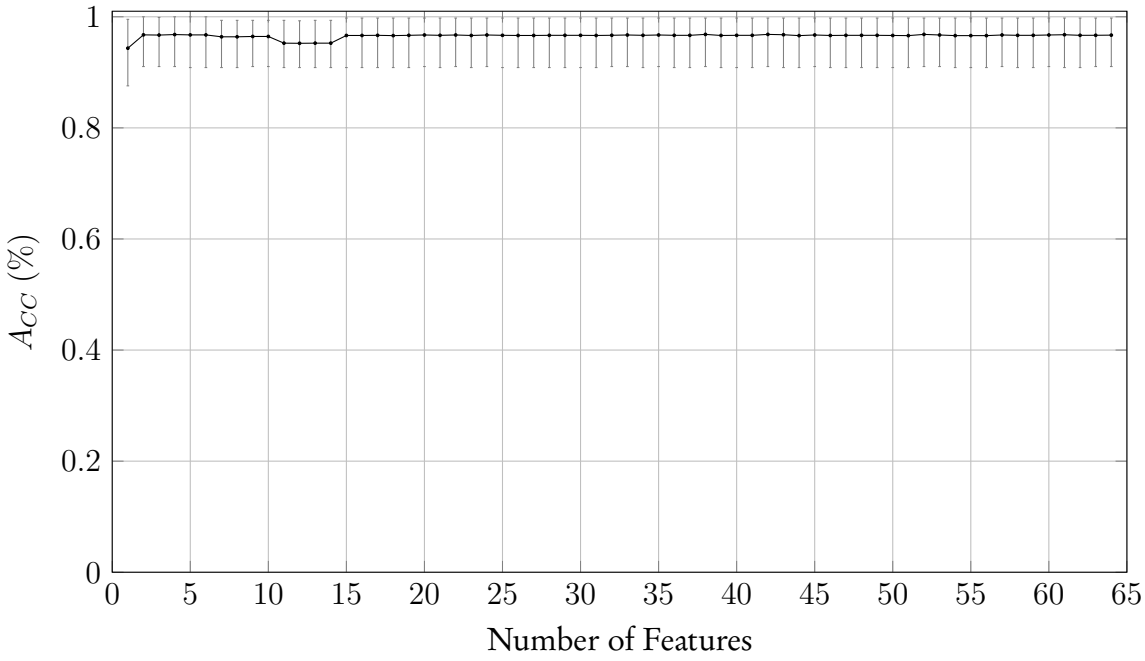


Figure 5.6: The mean cross-validated test-set accuracies achieved utilising optimal features and a decision tree with a maximum tree depth of two evaluated on the rhinoceros dataset. The error-bars indicate the maximum and minimum test-set accuracies achieved for the specific number of features.

Table 5.7 shows the cross-validated results of a DT classifier with a maximum tree depth of two using 38 optimal features. From the table we see that the DT classifier is able to discriminate between the three behaviours with high- precision, recall and $F1$ score. The confusion-matrix shows very little confusion between classes, however, there is a $5.27\,\%$ confusion between walking and lying down. A peak classification accuracy of $96.83\,\%$ was achieved using 38 features, of which the first 10 is shown in Table 5.7c. Figure 5.6 depicts that similar results can be obtained with two and more features and specifically by using the max(x) and mean(x) features.

Stellenbosch University https://scholar.sun.ac.za

Table 5.8: The performance of a naive Bayes classifier, utilising one feature, evaluated on the rhinoceros dataset.

(a) The per-class precision, recall and $F1$ score.

P_r	R_e	$F1$	Class	F_N
0.901	0.954	0.927	Lie	1
0.963	0.888	0.924	Stand	1
0.980	1.0	0.99	Walk	1

(b) The per-class confusion-matrix. The rows depict true labels and the columns depict predicted labels.

	Lie	Stand	Walk
Lie	850	30	11
Stand	93	791	7
Walk	0	0	891

(c) Top 10 selected features.

max(x), mean(x), min(y), mean(y), var(y), std(y), en(y), var(x), std(x), max(y)

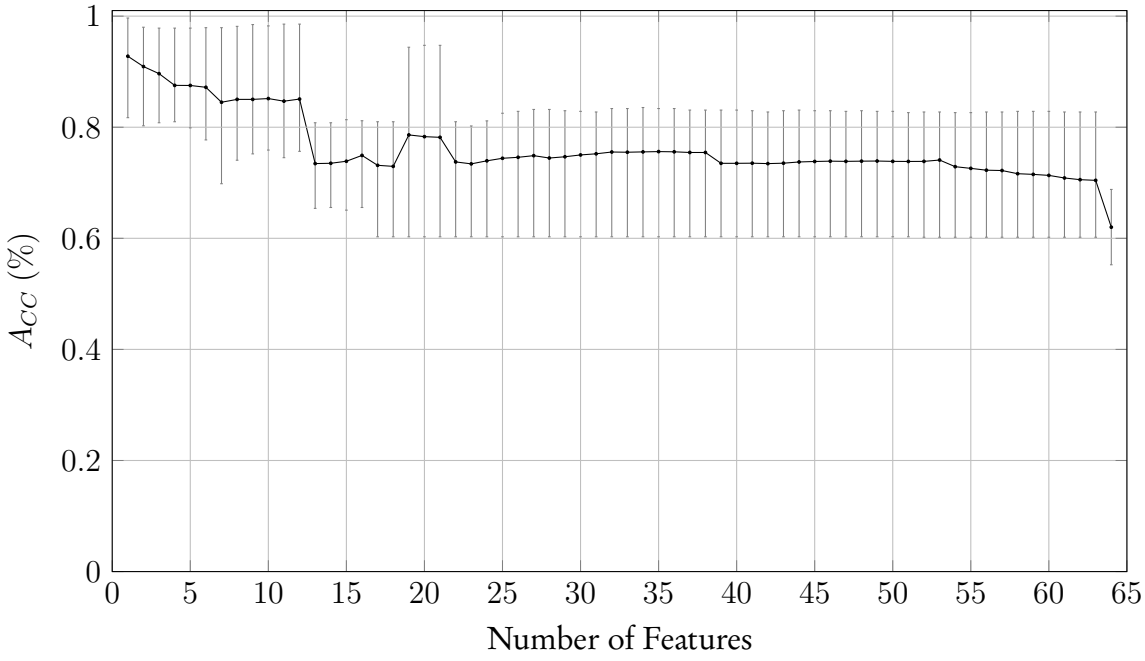


Figure 5.7: The mean cross-validated test-set accuracies achieved utilising optimal features and a naive Bayes classifier evaluated on the rhinoceros dataset. The error-bars indicate the maximum and minimum test-set accuracies achieved for the specific number of features.

Table 5.8 shows the cross-validated results of a NB classifier using max(x) as the only feature. From the table we see that the NB classifier is able to discriminate between the three behaviours with a reasonable- precision, recall and $F1$ score. The confusion-matrix shows that some extent of confusion exists between classes. When the actual class label is lying down $3.37\,\%$ of labels are wrongly classified as standing and similarly when predicting standing behaviour $10.44\,\%$ of samples are classified as lying down. A peak classification accuracy of $92.78\,\%$ was achieved using the single feature. Figure 5.7 shows that no other combination of features achieved good results for this classifier.

Stellenbosch University https://scholar.sun.ac.za

Table 5.9: The performance of quadratic discriminant analysis, utilising one feature, evaluated on the rhinoceros dataset.

(a) The per-class precision, recall and $F1$ score.

P_r	R_e	$F1$	Class	N_F
0.901	0.954	0.927	Lie	1
0.963	0.888	0.924	Stand	1
0.980	1.0	0.990	Walk	1

(b) The per-class confusion-matrix. The rows depict true labels and the columns depict predicted labels.

	Lie	Stand	Walk
Lie	850	30	11
Stand	93	791	7
Walk	0	0	891

(c) Top 10 selected features.

max(x), asm(), min(y), corr(x,z), kurt(x), mean(x), skew(y), corr(y,z), mean(y), en(y)

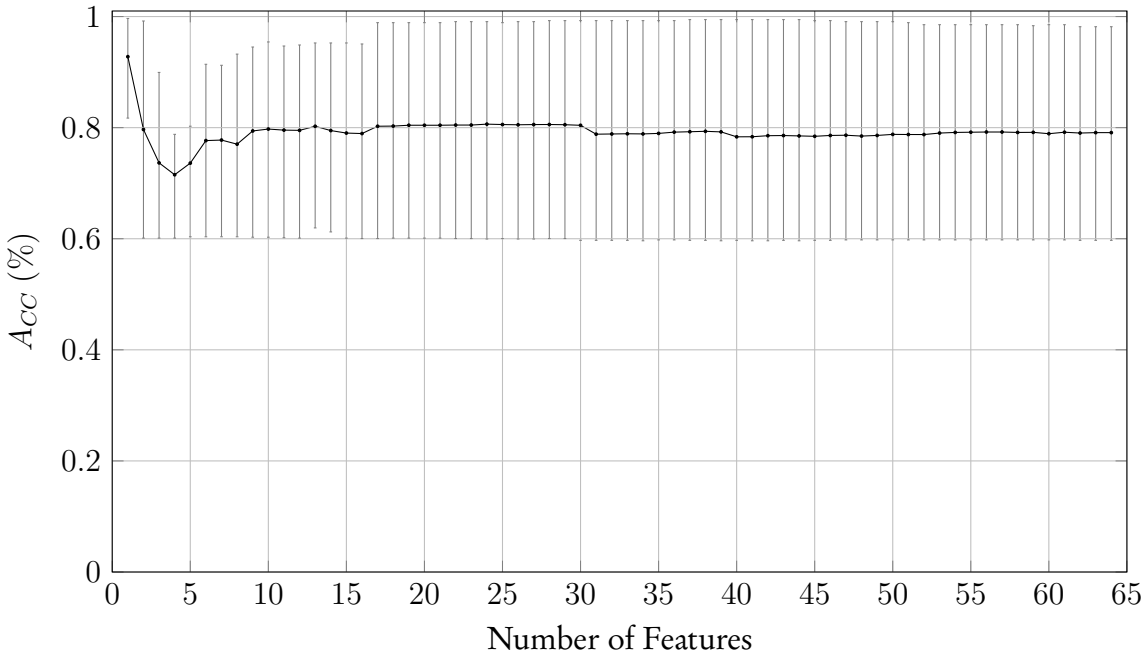


Figure 5.8: The mean cross-validated test-set accuracies achieved utilising optimal features and quadratic discriminant analysis evaluated on the rhinoceros dataset. The error-bars indicate the maximum and minimum test-set accuracies achieved for the specific number of features.

Table 5.9 shows the cross-validated results of QDA using max(x) as the only feature. From the table it is clear that the QDA classifier is able to discriminate between the three behaviours with a reasonable- precision, recall and $F1$ score. The confusion-matrix shows that some extent of confusion exists between classes. When the actual class label is lying down $3.37\,\%$ of labels are wrongly classified as standing and similarly when predicting standing behaviour $10.44\,\%$ of samples are classified as lying down. A peak classification accuracy of $92.78\,\%$ was achieved using the single feature. Figure 5.8 shows that no other combination of features achieved good results for this classifier. We note that the results presented in Tables 5.8 and 5.9 are exactly the same. This is no coincidence, in fact, if we consider the mathematical models of QDA and NB provided in Chapter 3, we see that if only one feature is present, QDA and NB are exactly the same when assuming the class-conditional distributions are Gaussian.

Stellenbosch University https://scholar.sun.ac.za

5.2 Results for the sheep dataset

A summary of the results obtained for the sheep dataset is provided in Table 5.10. The table shows the accuracy A_{CC} achieved for each classification model employed $C(\alpha)$ at the optimal hyper-parameter setting α and the optimal number of features F_N.

Table 5.10: Mean cross-validated accuracies achieved at the optimal number of features and optimal hyper-parameter setting for each classification technique evaluated on the sheep dataset.

Classifier $C(\alpha)$	F_N	A_{CC} (%)
LR	34	89.59
LSVM	45	88.23
QDA	48	88.04
k-NN (4)	49	88.01
DT (6)	54	87.83
RF (10)	53	87.38
NB	27	87.33
LDA	28	85.02

From the table it is evident that all the classification techniques achieved good classification accuracies. The logistic regression classifier, with 34 selected features, achieved the highest classification accuracy of $89.59\,\%$. The other models achieved accuracies between $85.02\,\%$ and $88.23\,\%$ with the number of features ranging from 27 to 54. Further details for the best performing model is provided in Table 5.11. Table 5.11a shows the precision, recall and $F1$ scores achieved with the LR model for the individual classes. Table 5.11b shows the confusion matrix for the LR model. The confusion matrix demonstrates some degree of confusion between classes. The main confusion exists between grazing behaviour and lying down. Table 5.11c lists the 10 most informative features, starting with the most important feature, the skewness of the y-axis. Furthermore, Figure 5.9 shows the results obtained from the sequential feature selection process. From the figure it is clear how the model progressively achieved higher accuracies as more features are added. From these results we also note that discriminating between the five behaviours in the sheep dataset is a harder problem than that of the rhinoceros. Tables 5.12 through 5.18 show detailed results for the remaining classification models evaluated.

Stellenbosch University https://scholar.sun.ac.za

Table 5.11: The performance of logistic regression, utilising 34 optimal features, evaluated on the sheep dataset.

(a) The per-class precision, recall and $F1$ score.

P_r	R_e	$F1$	Class	N_F
0.802	0.844	0.822	Lie	34
0.958	0.974	0.966	Stand	34
0.925	0.923	0.924	Walk	34
0.994	0.986	0.990	Run	34
0.793	0.746	0.769	Graze	34

(b) The per-class confusion-matrix. The rows depict true labels and the columns depict predicted labels.

	Lie	Stand	Walk	Run	Graze
Lie	842	0	1	0	155
Stand	0	967	22	0	4
Walk	0	36	917	6	34
Run	0	0	13	979	1
Graze	208	6	38	0	741

(c) Top 10 selected features.

skew(y), bin_3(x), var(x), min(z), min(y), var(z), var(y), mean(y), std(y), std(x)

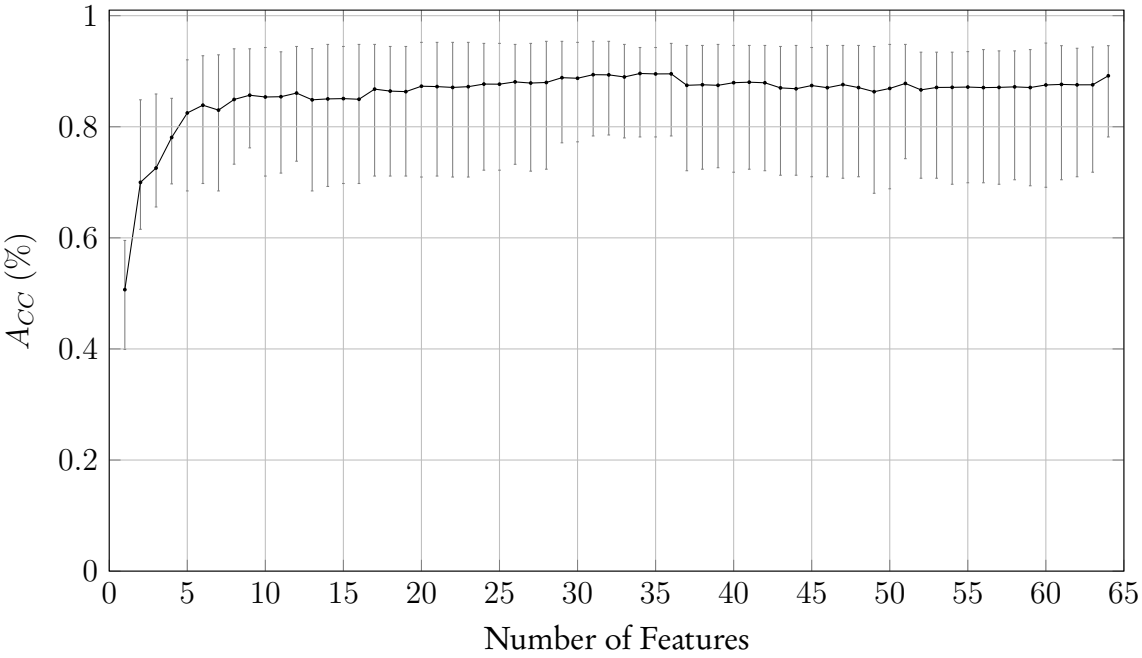


Figure 5.9: The mean cross-validated test-set accuracies achieved utilising optimal features and logistic regression evaluated on the sheep dataset. The error-bars indicate the maximum and minimum test-set accuracies achieved for the specific number of features.

Stellenbosch University https://scholar.sun.ac.za

Table 5.12: The performance of a linear support vector machine, utilising 45 optimal features, evaluated on the sheep dataset.

(a) The per-class precision, recall and $F1$ score.

P_r	R_e	$F1$	Class	N_F
0.800	0.772	0.786	Lie	45
0.967	0.966	0.966	Stand	45
0.930	0.930	0.930	Walk	45
0.987	0.982	0.984	Run	45
0.748	0.779	0.763	Graze	45

(b) The per-class confusion-matrix. The rows depict true labels and the columns depict predicted labels.

	Lie	Stand	Walk	Run	Graze
Lie	770	1	1	0	226
Stand	0	959	28	0	6
Walk	1	28	923	13	28
Run	3	0	14	975	1
Graze	188	4	27	0	774

(c) Top 10 selected features.

bin_3(x), skew(y), bin_4(z), min(y), var(z), mean(y), min(z), skew(z), var(y), min(x)

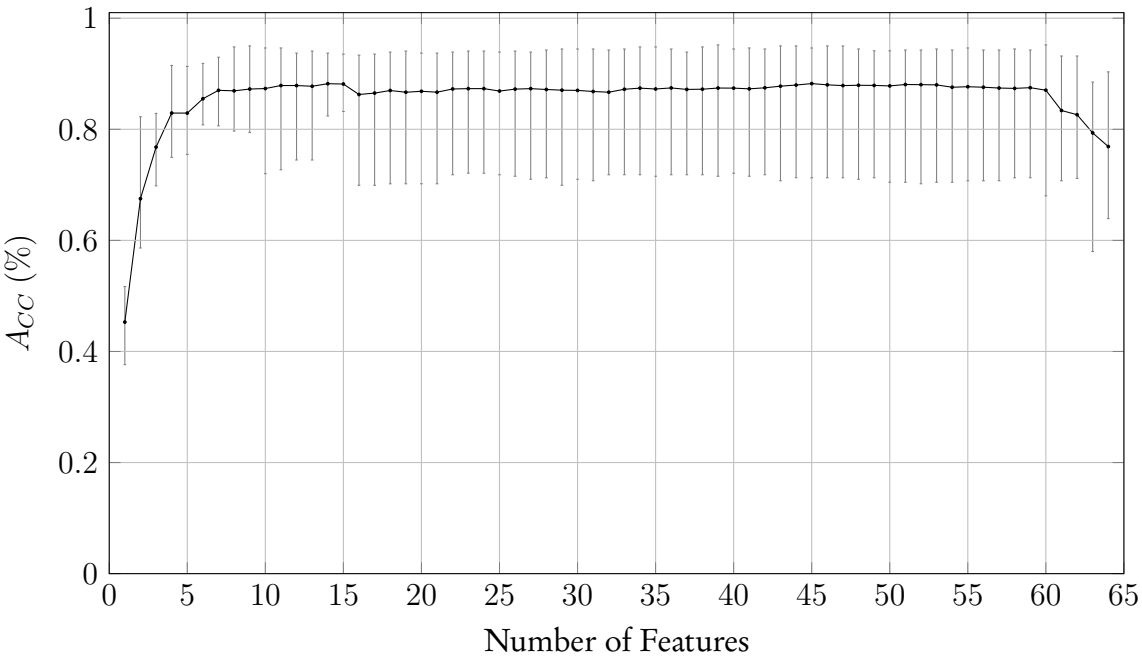


Figure 5.10: The mean cross-validated test-set accuracies achieved utilising optimal features and a linear support vector machine evaluated on the sheep dataset. The error-bars indicate the maximum and minimum test-set accuracies achieved for the specific number of features.

Table 5.12 shows the cross-validated results of a LSVM using 45 optimal features. The table shows that the LSVM is able to discriminate between the five behaviours with a good precision, recall and $F1$ score. The confusion-matrix shows little confusion between classes other than lying down and grazing. A peak classification accuracy of $88.23\,\%$ was achieved using 45 features, of which the first 10 is shown in Table 5.12c. From Figure 5.10 it is clear that only using the first seven features will also result in good model performance.

Stellenbosch University https://scholar.sun.ac.za

Table 5.13: The performance of quadratic discriminant analysis, utilising 48 optimal features, evaluated on the sheep dataset.

(a) The per-class precision, recall and $F1$ score.

P_r	R_e	$F1$	Class	N_F
0.871	0.737	0.799	Lie	48
0.964	0.912	0.937	Stand	48
0.869	0.927	0.897	Walk	48
0.974	0.997	0.986	Run	48
0.742	0.829	0.783	Graze	48

(b) The per-class confusion-matrix. The rows depict true labels and the columns depict predicted labels.

	Lie	Stand	Walk	Run	Graze
Lie	736	0	0	6	256
Stand	0	906	79	1	7
Walk	0	32	921	17	23
Run	0	0	3	990	0
Graze	109	2	57	2	823

(c) Top 10 selected features.

min(z), corr(x,z), en(x), min(x), min(y), skew(x), skew(y), mean(y), asm(), mean(z)

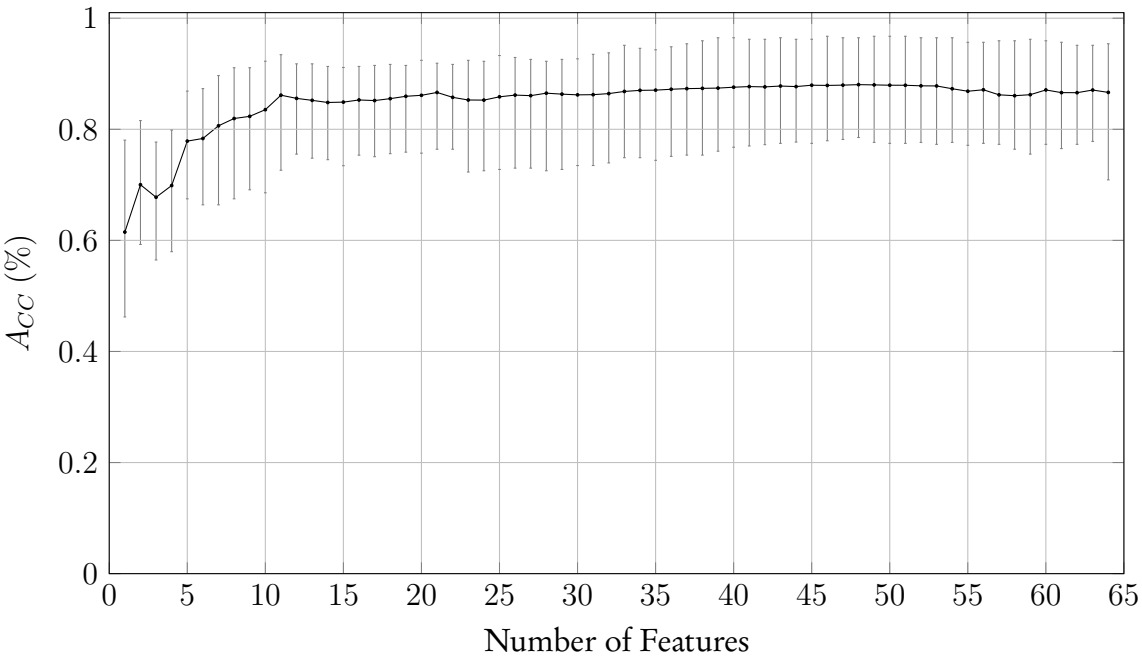


Figure 5.11: The mean cross-validated test-set accuracies achieved utilising optimal features and quadratic discriminant analysis evaluated on the sheep dataset. The error-bars indicate the maximum and minimum test-set accuracies achieved for the specific number of features.

Table 5.13 shows the cross-validated results of QDA using 48 optimal features. From the table it is clear that QDA is able to discriminate between the five behaviours with a good precision, recall and $F1$ score. The confusion-matrix shows little confusion between classes other than lying down and grazing. A peak classification accuracy of $88.04\,\%$ was achieved using 48 features, of which the first 10 is shown in Table 5.13c. Figure 5.11 shows that only using the first 11 features will also result in good model performance.

Stellenbosch University https://scholar.sun.ac.za

Table 5.14: The performance of a k-nearest neighbour classifier (with 4-nearest neighbours), utilising 49 optimal features, evaluated on the sheep dataset.

(a) The per-class precision, recall and $F1$ score.

P_r	R_e	$F1$	Class	N_F
0.839	0.763	0.799	Lie	49
0.913	0.962	0.937	Stand	49
0.940	0.876	0.907	Walk	49
0.998	0.981	0.989	Run	49
0.724	0.814	0.767	Graze	49

(b) The per-class confusion-matrix. The rows depict true labels and the columns depict predicted labels.

	Lie	Stand	Walk	Run	Graze
Lie	761	1	0	0	236
Stand	1	955	11	0	26
Walk	1	75	870	2	45
Run	0	0	19	974	0
Graze	144	15	26	0	808

(c) Top 10 selected features.

min(y), min(z), var(x), std(x), var(y), en(z), max(z), asm(), std(y), var(z)

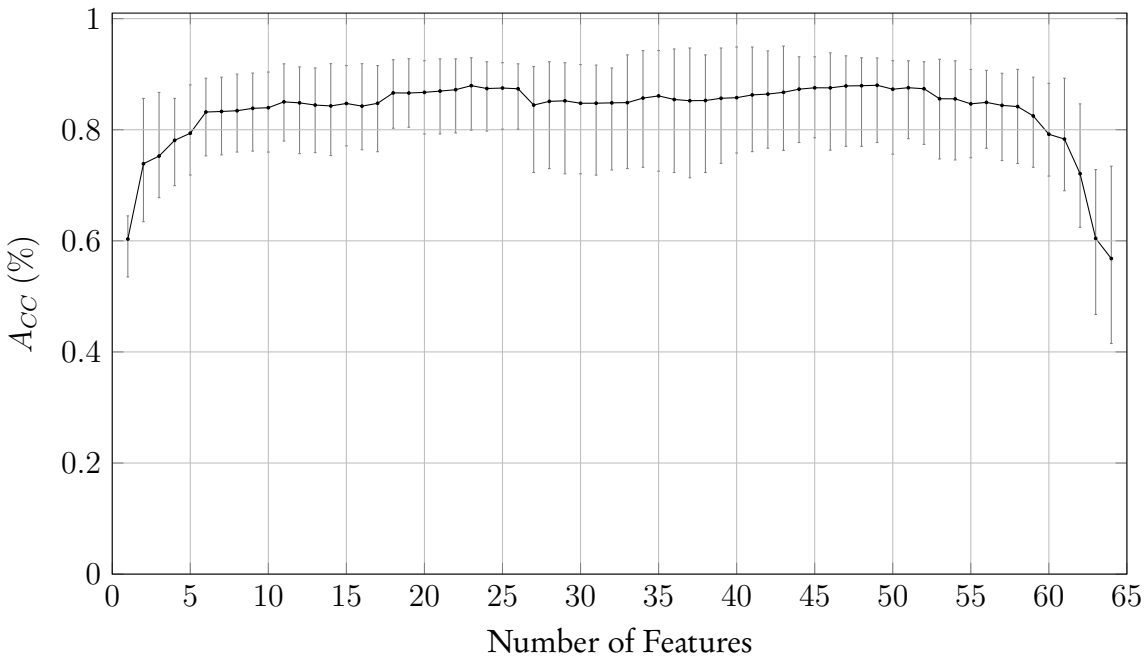


Figure 5.12: The mean cross-validated test-set accuracies achieved utilising optimal features and a k-nearest neighbours classifier (with 4-nearest neighbours) evaluated on the sheep dataset. The error-bars indicate the maximum and minimum test-set accuracies achieved for the specific number of features.

Table 5.14 shows the cross-validated results of a k-NN classifier with 4-nearest neighbours using 49 optimal features. The table shows that the k-NN classifier is able to distinguish between the five behaviours with a good precision, recall and $F1$ score. The confusion-matrix shows little confusion between classes other than lying down and grazing. A peak classification accuracy of $88.01\,\%$ was achieved using 49 features, of which the first 10 is shown in Table 5.14c. Figure 5.12 depicts that only using the first 23 features will also result in good model performance.

Stellenbosch University https://scholar.sun.ac.za

Table 5.15: The performance of a decision tree classifier (with a maximum tree depth of six), utilising 54 optimal features, evaluated on the sheep dataset.

(a) The per-class precision, recall and $F1$ score.

P_r	R_e	$F1$	Class	N_F
0.884	0.768	0.821	Lie	54
0.949	0.906	0.927	Stand	54
0.872	0.875	0.874	Walk	54
0.978	0.983	0.980	Run	54
0.722	0.844	0.778	Graze	54

(b) The per-class confusion-matrix. The rows depict true labels and the columns depict predicted labels.

	Lie	Stand	Walk	Run	Graze
Lie	766	0	0	0	232
Stand	0	898	61	6	28
Walk	2	53	864	14	60
Run	0	0	18	975	0
Graze	101	3	50	0	839

(c) Top 10 selected features.

dist(x,y), asm(), min(y), var(y), min(x), en(y), skew(y), corr(x,z), var(x), mean(z)

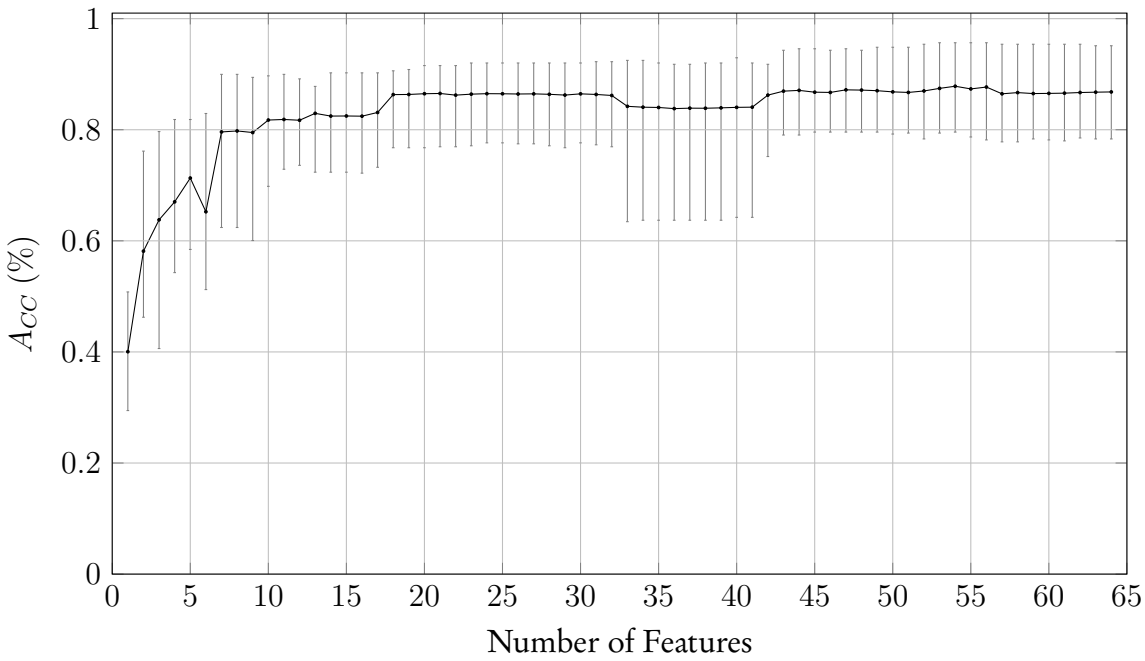


Figure 5.13: The mean cross-validated test-set accuracies achieved utilising optimal features and a decision tree classifier (with a maximum tree depth of six) evaluated on the sheep dataset. The error-bars indicate the maximum and minimum test-set accuracies achieved for the specific number of features.

Table 5.15 shows the cross-validated results of a DT classifier with a maximum tree depth of 6 using 54 optimal features. From the table it is clear that the DT classifier is able to distinguish between the five behaviours with a good precision, recall and $F1$ score. The confusion-matrix shows little confusion between classes other than lying down and grazing. A peak classification accuracy of $87.83\,\%$ was achieved using 54 features, of which the first 10 is shown in Table 5.15c. Figure 5.13 depicts that only using the first 18 features will also result in good model performance.

Stellenbosch University https://scholar.sun.ac.za

Table 5.16: The performance of a random forest classifier (with a maximum tree depth of 10), utilising 53 optimal features, evaluated on the sheep dataset.

(a) The per-class precision, recall and $F1$ score.

P_r	R_e	$F1$	Class	N_F
0.852	0.734	0.789	Lie	53
0.945	0.921	0.933	Stand	53
0.890	0.883	0.887	Walk	53
0.986	0.993	0.989	Run	53
0.709	0.826	0.763	Graze	53

(b) The per-class confusion-matrix. The rows depict true labels and the columns depict predicted labels.

	Lie	Stand	Walk	Run	Graze
Lie	733	0	0	0	265
Stand	0	915	59	8	11
Walk	0	49	877	6	61
Run	0	0	7	986	0
Graze	127	4	42	0	820

(c) Top 10 selected features.

min(y), asm(), en(x), var(y), mean(x), var(x), max(x), max(z), min(z), mean(z)

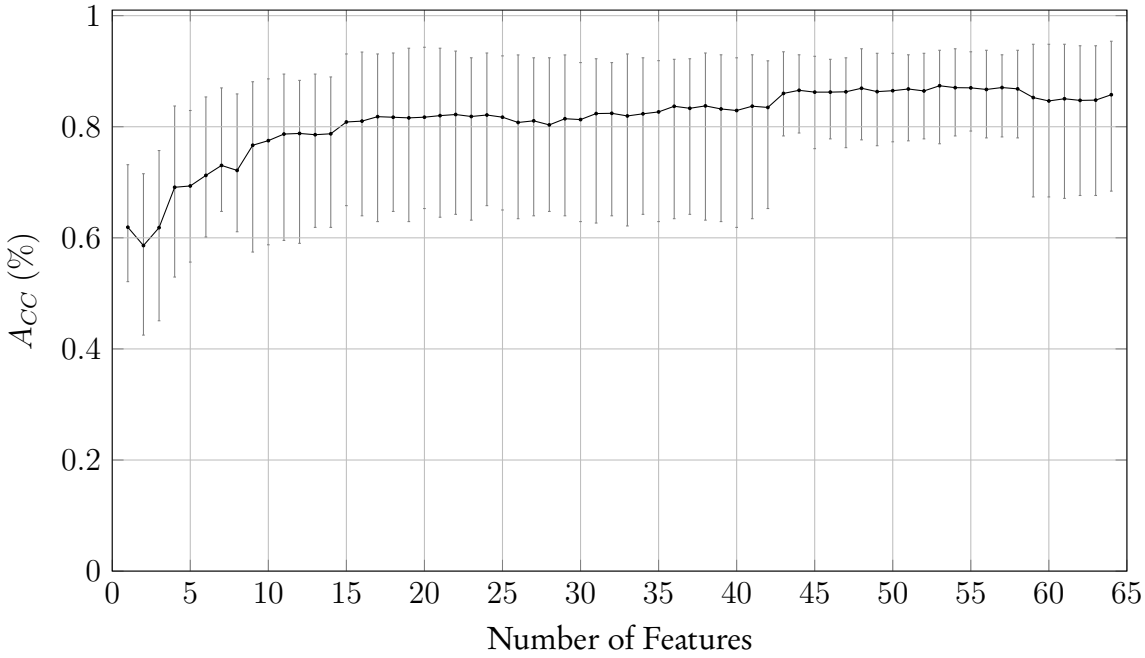


Figure 5.14: The mean cross-validated test-set accuracies achieved utilising optimal features and a random forest classifier (with a maximum tree depth of 10) evaluated on the sheep dataset. The error-bars indicate the maximum and minimum test-set accuracies achieved for the specific number of features.

Table 5.16 shows the cross-validated results of a RF classifier with a maximum tree depth of 10 using 53 optimal features. The table demonstrates that the RF classifier is able to distinguish between the five behaviours with a good precision, recall and $F1$ score. The confusion-matrix shows little confusion between classes other than lying down and grazing. A peak classification accuracy of $87.38\,\%$ was achieved using 53 features, of which the first 10 is shown in Table 5.16c. From Figure 5.14 we see that only using the first 43 features will also result in good model performance.

Stellenbosch University https://scholar.sun.ac.za

Table 5.17: The performance of a naive Bayes classifier, utilising 27 optimal features, evaluated on the sheep dataset.

(a) The per-class precision, recall and $F1$ score.

P_r	R_e	$F1$	Class	N_F
0.784	0.8170	0.802	Lie	27
0.920	0.937	0.928	Stand	27
0.886	0.880	0.883	Walk	27
0.984	0.998	0.991	Run	27
0.771	0.716	0.743	Graze	27

(b) The per-class confusion-matrix. The rows depict true labels and the columns depict predicted labels.

	Lie	Stand	Walk	Run	Graze
Lie	818	2	1	4	173
Stand	0	930	58	0	5
Walk	0	76	874	10	33
Run	0	0	2	991	0
Graze	225	3	52	2	711

(c) Top 10 selected features.

skew(y), min(z), min(y), en(x), max(z), mean(x), asm(), skew(x), max(x), std(y)

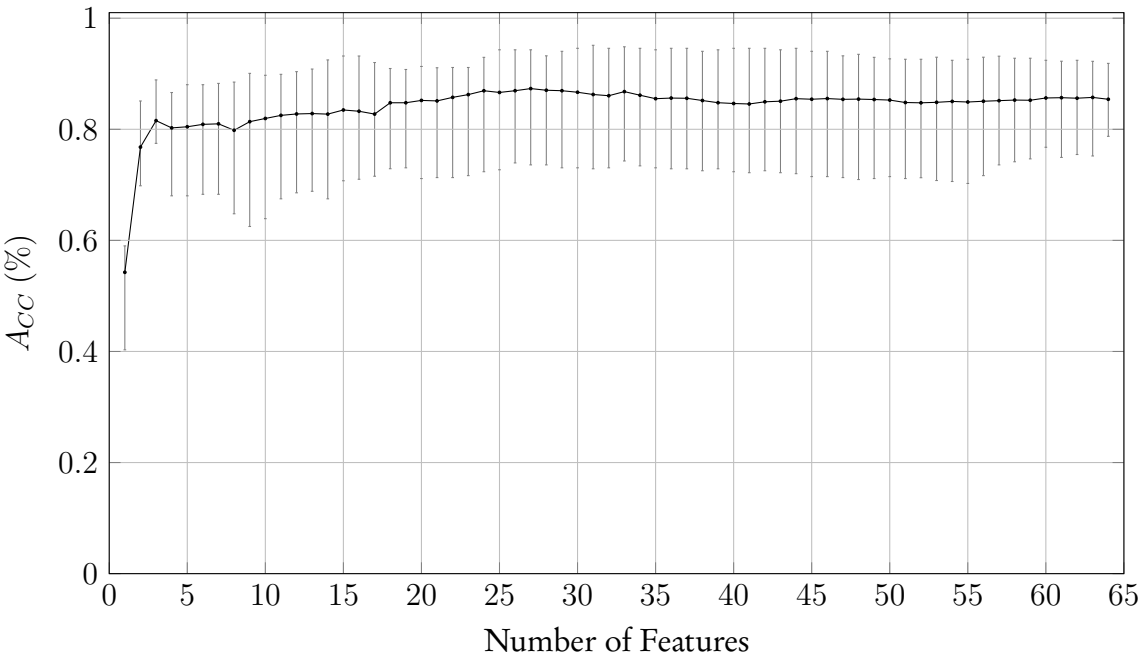


Figure 5.15: The mean cross-validated test-set accuracies achieved utilising optimal features and a naive Bayes classifier evaluated on the sheep dataset. The error-bars indicate the maximum and minimum test-set accuracies achieved for the specific number of features.

Table 5.17 shows the cross-validated results of a NB classifier using 27 optimal features. The table shows that the NB classifier is able to distinguish between the five behaviours with a good precision, recall and $F1$ score. The confusion-matrix shows little confusion between classes other than lying down and grazing. A peak classification accuracy of $87.33\,\%$ was achieved using 27 features, of which the first 10 is shown in Table 5.17c. Figure 5.15 shows that only using the first 24 features will also result in good model performance.

Stellenbosch University https://scholar.sun.ac.za

Table 5.18: The performance of linear discriminant analysis, utilising 28 optimal features, evaluated on the sheep dataset.

(a) The per-class precision, recall and $F1$ score.

P_r	R_e	$F1$	Class	N_F
0.748	0.973	0.846	Lie	28
0.869	0.898	0.883	Stand	28
0.867	0.779	0.821	Walk	28
0.997	0.987	0.992	Run	28
0.779	0.603	0.680	Graze	28

(b) The per-class confusion-matrix. The rows depict true labels and the columns depict predicted labels.

	Lie	Stand	Walk	Run	Graze
Lie	971	2	0	0	25
Stand	18	892	28	0	55
Walk	3	124	774	3	89
Run	0	0	12	980	1
Graze	306	9	79	0	599

(c) Top 10 selected features.

min(z), en(z), skew(y), skew(z), kurt(z), bin_3(x), kurt(x), bin_6(x), var(z), min(x)

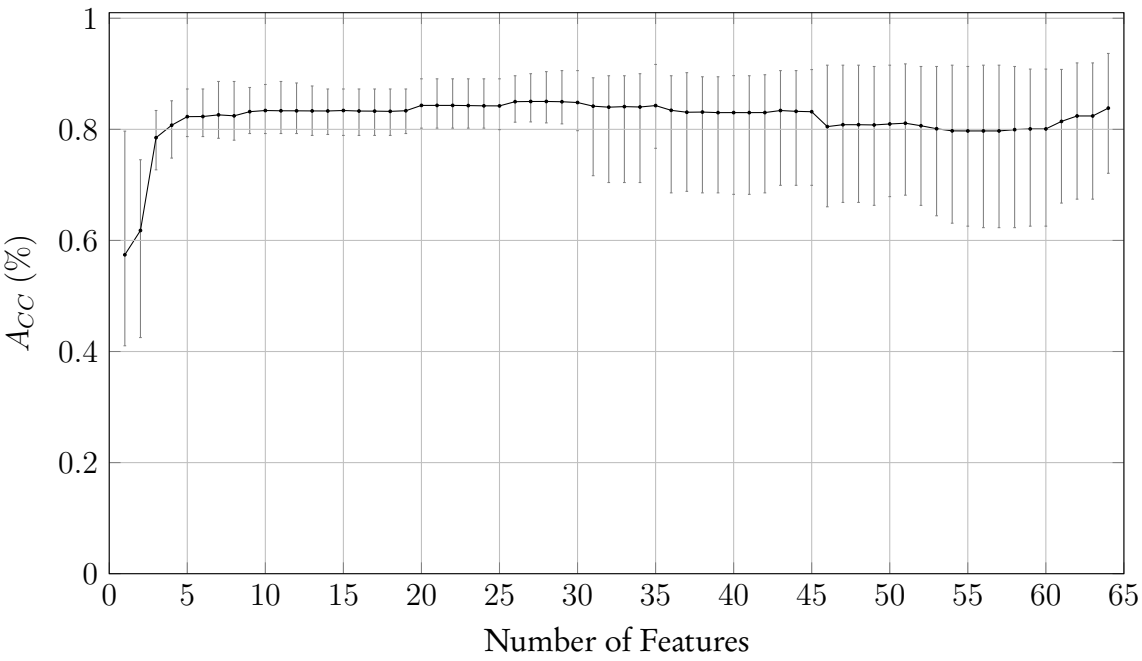


Figure 5.16: The mean cross-validated test-set accuracies achieved utilising optimal features and linear discriminant analysis evaluated on the sheep dataset. The error-bars indicate the maximum and minimum test-set accuracies achieved for the specific number of features.

Table 5.18 shows the cross-validated results of LDA using 28 optimal features. From the table we see that LDA is able to discriminate between the five behaviours with a moderate precision, recall and $F1$ score. The confusion-matrix exhibits some extent of confusion between the previously noted classes, lying down and grazing, and also between walking and standing. A peak classification accuracy of $85.02\,\%$ was achieved using 28 features, of which the first 10 is shown in Table 5.18c.

Stellenbosch University https://scholar.sun.ac.za

5.3 Conclusion

This chapter presented the mean cross-validated results for the different offline automatic behaviour classification techniques considered for both the sheep and rhinoceros datasets. For the rhinoceros dataset a linear support vector machine with 11 selected features, achieved the highest classification accuracy of $99.61\,\%$. For the sheep dataset a logistic regression classifier, with 34 selected features, achieved the highest classification accuracy of $89.59\,\%$. Furthermore, the precision, recall and $F1$-scores were provided for these top-preforming models and also for the other models considered. For each classification model, the top 10 selected features were presented and the sequential addition of features affecting the model performance demonstrated. The top performing models for both the sheep and rhinoceros datasets will subsequently be used in the next chapter to achieve real-time embedded behaviour classification.

Stellenbosch University https://scholar.sun.ac.za

Chapter 6

Real-time embedded behaviour classification

This chapter explains how real-time automatic behaviour classification is achieved by implementing the optimal classification models, for both sheep and rhinoceros, on the tags itself. This allows real-time behavioural updates to be calculated and transmitted to a receiver station where it is available for further analysis. This chapter is based on our earlier work as presented in [2].

6.1 Procedure overview

In short, real-time embedded behaviour classification is achieved by implementing a trained statistical classifier on the biotelemetry tag itself to enable immediate feature calculation and prediction of different behaviours. This is in contrast with the classical approach where the data is first collected or transmitted and later processed at a remote station. For this purpose, the tag periodically fills a data buffer with tri-axial accelerometer measurements sampled at a fixed frequency. This buffer of measurements is subsequently used for feature extraction, where a number of different informative metrics are calculated. The extracted features are then used as the input to a trained statistical classifier which discriminates between a number of behavioural classes. The resulting predicted behaviour is subsequently transmitted to a receiver station. The GPS coordinates are also acquired and transmitted at a slower interval. This provides real-time insights on what the animals are doing and also where they show particular behavioural patterns.

Stellenbosch University https://scholar.sun.ac.za

6.2 Embedded rhinoceros behaviour classification model

For the rhinoceros dataset the best classification model is a linear support vector machine with 11 features. From Chapter 5 we saw that this model achieves a classification accuracy of $99.61\,\%$ among three behavioural classes. The 11 features extracted from each frame of raw measurements are max(x), mean(x), asm(), min(x), kurt(x), var(x), std(x), en(x), pse(x), max(y) and min(y). The embedded implementation requires data sampling, feature extraction and the subsequent classification. Classification is achieved by evaluating the implemented decision boundary of the LSVM model. A linear support vector machine assigns a d-dimensional input feature vector $\mathbf{x} = \{x_1, ..., x_d\}$ to one of k classes $\omega_1, \omega_2, ..., \omega_k$ using a linear decision boundary as shown in Chapter 3. Classification is achieved by means of a linear machine which divides the feature space into k decision regions $\mathcal{R}_1, \mathcal{R}_2, ..., \mathcal{R}_k$ using k linear discriminant functions $g_i(\mathbf{x})$ as shown in Equation 6.1. The vectors $\mathbf{v}_i$ and the intercept terms v_{i0} indicate the decision boundaries between the k classes and the maximum discriminant indicates the region within which $\mathbf{x}$ falls. Accordingly, we assign $\mathbf{x}$ to ω_i when $g_i(\mathbf{x}) > g_j(\mathbf{x})$ for all $i \neq j$.

$$g_i(\mathbf{x}) = \mathbf{v}_i^{\mathrm{T}}\mathbf{x} + v_{i0} \qquad i = 1, ..., k \tag{6.1}$$

For the presented model we have a feature vector with dimension $d = 11$ and since the model is discriminating between three classes, $k = 3$. The decision boundaries are learned during training which produce the weight values of $\mathbf{v}_i$ and v_{i0}. We can combine the latter into a weight matrix $\mathbf{V}$ and intercept vector $\mathbf{v}_0$ and conveniently write Equation 6.1 as,

$$\mathbf{g}(\mathbf{x}) = \mathbf{x}\mathbf{V}^{\mathrm{T}} + \mathbf{v}_0. \tag{6.2}$$

In Equation 6.2, the dimensions of $\mathbf{x}$ and $\mathbf{V}^{\mathrm{T}}$ is 1×11 and 11×3, respectively. Their product results in a 1×3 dimensional vector. And similarly a 1×3 dimensional vector is produced after adding the bias terms. The index of the maximum discriminant in the resulting vector corresponds to the class label. Algorithm 2 provides the details of our firmware implementation of the classifier. In the algorithm $\mathbf{v}$ represents an array (or row vector) of the elements of $\mathbf{V}^{\mathrm{T}}$ where each vector $\mathbf{v}_i^{\mathrm{T}}$ is followed by the other such that $\mathbf{v} = [\mathbf{v}_1^{\mathrm{T}}\ \mathbf{v}_2^{\mathrm{T}}\ \mathbf{v}_3^{\mathrm{T}}]$ and similarly the intercept terms are combined to provide $\mathbf{v}_0 = [v_{10}\ v_{20}\ v_{30}]$.

Stellenbosch University https://scholar.sun.ac.za

ALGORITHM 2: Support vector machine embedded classification implementation.

```
1:  procedure SUPPORT VECTOR MACHINE CLASSIFICATION(x)
2:      v ← [1 × 33] # floating point weight values expressed as a vector
3:      v_0 ← [1 × 3] # floating point bias values
4:      g ← {0, 0, 0}
5:      l ← 0
6:      for i ← 0; i < 3 do
7:          for j ← 0; j < 11 do
8:              g[i] ← g[i] + x[j] · v[l++]
9:          end for
10:         g[i] ← g[i] + v_0[i]
11:     end for
12:     max ← g[0]
13:     for i ← 0; i < 3 do
14:         if g[i] ≥ max then
15:             max ← g[i]
16:             class ← i
17:         end if
18:     end for
19: return class
20: end procedure
```

6.3 Embedded sheep behaviour classification model

For the sheep dataset the best classification model is a logistic regression classifier, with 34 features. From Chapter 5 we saw that this model achieves a classification accuracy of $89.59\,\%$. The 34 features extracted from each frame of raw measurements are skew(y), bin_3(x), var(x), min(z), min(y), var(z), var(y), mean(y), std(y), std(x), max(z), mean(z), asm(), bin_5(x), bin_7(x), dist(y,z), min(x), dist(x,y), bin_9(x), max(y), mean(x), bin_8(x), bin_4(z), kurt(y), std(z), skew(z), max(x), bin_4(x), corr(x,z), bin_6(x), bin_5(y), dist(x,z), kurt(x) and pse(z). The embedded implementation similarly requires data sampling, feature extraction and the subsequent classification. The classification is achieved by evaluating the implemented decision boundary of the logistic regression model. Logistic regression combines the input values linearly whereafter a sigmoid function is applied to produce a binary output as shown in

Stellenbosch University https://scholar.sun.ac.za

Chapter 3. Logistic regression utilises k discriminant functions as shown in Equation 6.3. Once again, the maximum discriminant indicates the region, $\mathcal{R}_1, \mathcal{R}_2, ..., \mathcal{R}_k$, within which $\mathbf{x}$ falls and we assign $\mathbf{x}$ to ω_i when $g_i(\mathbf{x}) > g_j(\mathbf{x})$ for all $i \neq j$.

$$g_i(\mathbf{x}) = \frac{e^{\mathbf{v}_i^{\mathrm{T}}\mathbf{x}+v_{i0}}}{1 + e^{\mathbf{v}_i^{\mathrm{T}}\mathbf{x}+v_{i0}}} \qquad i = 1, ..., k \tag{6.3}$$

For the presented model we have a feature vector with dimension $d = 34$ and since the model is discriminating between five classes $k = 5$. The decision boundaries are learned during training which produce the weight values of $\mathbf{v}_i$ and v_{i0}. The weight vectors and intercept terms can be combined and expressed as $\mathbf{V}^{\mathrm{T}}$ and $\mathbf{v}_0$. The dimensions of $\mathbf{x}$ and $\mathbf{V}^{\mathrm{T}}$ is 1×34 and 34×5, respectively. Their product results in a 1×5 dimensional vector. And similarly, a 1×5 dimensional vector is produced after adding the bias terms. After applying a sigmoid function the index of the maximum discriminant in the resulting vector corresponds to the class label. Algorithm 3 provides the details of our firmware implementation of the classifier. In the algorithm $\mathbf{v}$ represents an array (or row vector) of the elements of $\mathbf{V}^{\mathrm{T}}$ where each vector $\mathbf{v}_i^{\mathrm{T}}$ is followed by the other such that $\mathbf{v} = [\mathbf{v}_1^{\mathrm{T}}\ \mathbf{v}_2^{\mathrm{T}}\ \mathbf{v}_3^{\mathrm{T}}\ \mathbf{v}_4^{\mathrm{T}}\ \mathbf{v}_5^{\mathrm{T}}]$ and similarly the intercept terms are combined to provide $\mathbf{v}_0 = [v_{10}\ v_{20}\ v_{30}\ v_{40}\ v_{50}]$.

ALGORITHM 3: Logistic regression embedded classification implementation.

```
1:  procedure LOGISTIC REGRESSION CLASSIFICATION(x)
2:      v ← [1 × 170] # floating point weight values expressed as a vector
3:      v_0 ← [1 × 5] # floating point bias values
4:      g ← {0, 0, 0, 0, 0}
5:      l ← 0
6:      for i ← 0; i < 5 do
7:          for j ← 0; j < 34 do
8:              g[i] ← g[i] + x[j] · v[l++]
9:          end for
10:         g[i] ← sigmoid(g[i] + v_0[i])
11:     end for
12:     max ← g[0]
13:     for i ← 0; i < 5 do
14:         if g[i] ≥ max then
15:             max ← g[i]
16:             class ← i
17:         end if
```

Stellenbosch University https://scholar.sun.ac.za

18: **end for**
19: **return** $class$
20: **end procedure**

6.4 Firmware flow diagram

The details of the embedded firmware implementation is shown in Figure 6.1. The figure shows a flow diagram of the firmware routine implemented on the biotelemetry tags. From the flow diagram we see that the routine begins by initialising the system and setting all variables to appropriate values. Next, two timers are started. The first corresponds to the desired accelerometer sampling frequency of 20 Hz for both sheep and rhinoceros. The second corresponds to the desired GPS sampling interval. The routine then enters the main loop, in which it waits for the read-accelerometer timer to trigger an interrupt. The corresponding interrupt service routine samples the accelerometer and stores the raw x, y, and z acceleration values in a data buffer. This process is repeated until 128 samples have been obtained. Feature extraction is subsequently performed on the frame stored in the data buffer. The feature vector is subsequently passed to the onboard classifier which performs classification as explained in Algorithms 2 and 3 for rhinoceros and sheep datasets, respectively. The classification result is subsequently transmitted and optionally also stored onboard. At this stage, if the GPS timer was triggered, the GPS location can be obtained, transmitted and stored. The loop completes by clearing the data buffer. An additional delay can be introduced at this point during which the system enters a low-power state for a set duration. This is useful when behavioural updates are required less frequently. However, the system was configured for continuous behavioural updates, hence no delay was introduced. With this software implementation behavioural updates are available approximately every 6.5 s for sheep and rhinoceros, since the sampling is done at constant interval of 20 Hz and 128 samples are stored in the data buffer before classification and transmission.

Stellenbosch University https://scholar.sun.ac.za

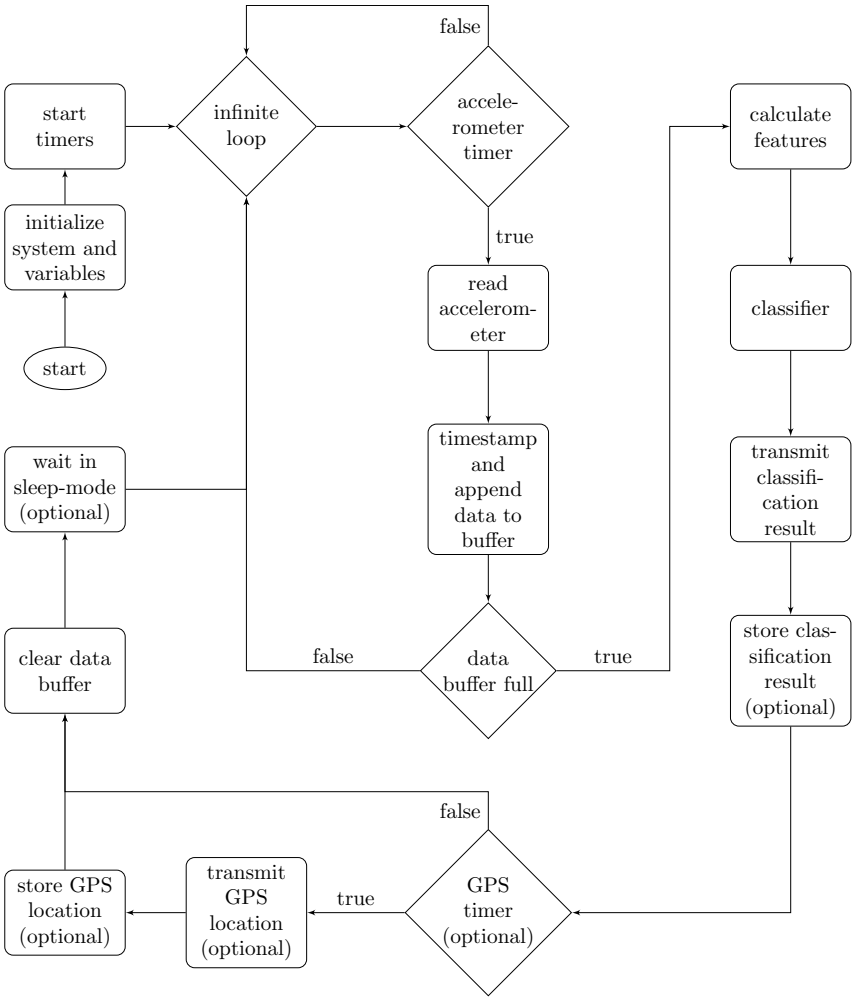


Figure 6.1: Software flow diagram of the real-time embedded behaviour classification system. Accelerometer sampling is controlled by a timer to ensure a constant sampling frequency of 20 Hz for both sheep and rhinoceros. The GPS acquisition interval is determined by a second timer.

6.5 Real-time behavioural updates

The embedded behaviour classification provides real-time behavioural updates and when combined with other information such as GPS locations it provides enhanced levels of information previously not achievable in real-time with conventional techniques. For example, when the real-time embedded behaviour classification system was deployed on rhinoceros, behavioural updates and GPS locations were transmitted to a the receiver station approximately every 3 min for a 24 h period [90]. Figure 6.2 shows the collected information. For security purposes the GPS coordinates and terrain data are not provided in the figure. We see that both the behaviour and location of the rhinoceros are available in real-time. The figure depicts where the rhinoceros was lying down (blue), where it stood (red) and where it walked (black). The analysis of the spatial and temporal relationship of animal behaviour holds great potential in fields such as behavioural ecology, nature conservation and precision farming.

Stellenbosch University https://scholar.sun.ac.za

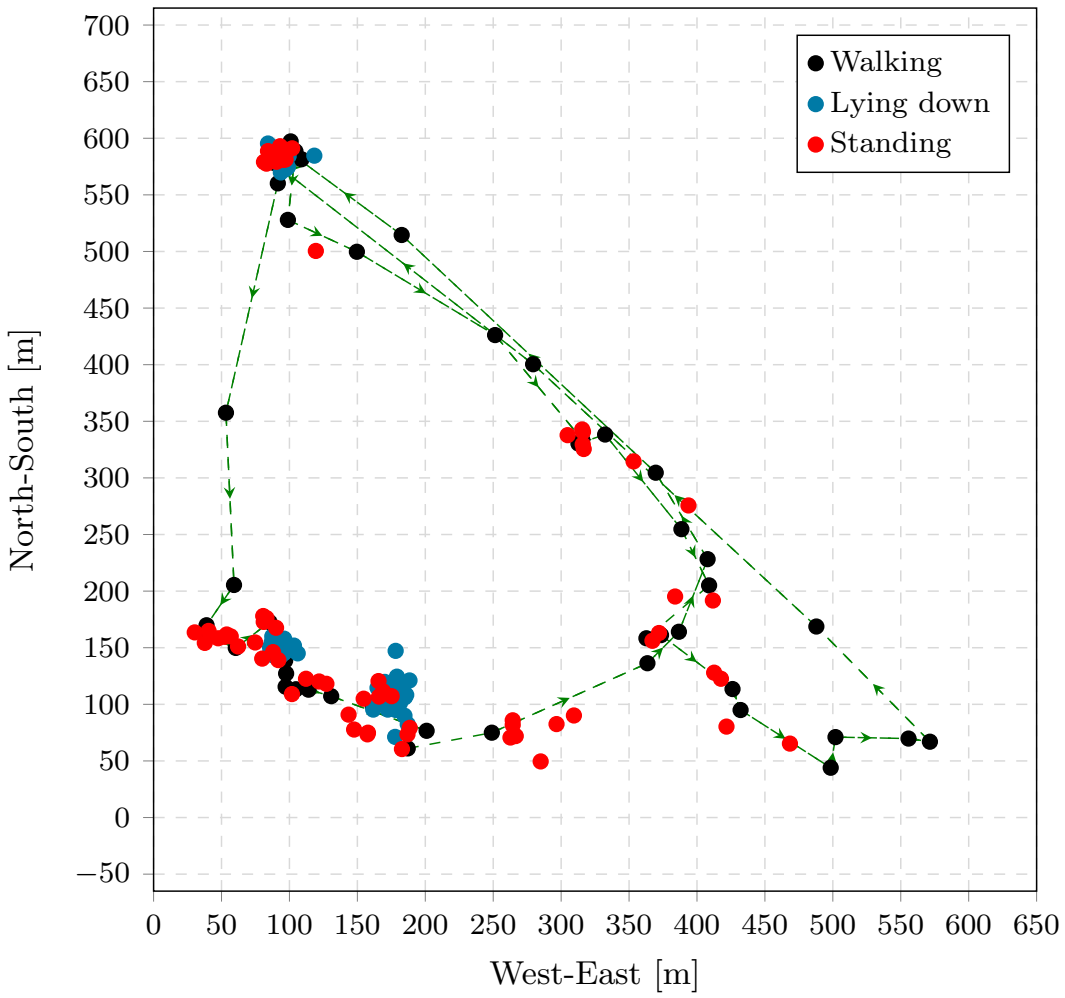


Figure 6.2: Rhinoceros behaviour and movement collected in real-time. Since the behaviour is classified on the biotelemetry tag it can be combined with GPS location to produce information on what, where and when the animal is exhibiting specific behaviours. This figure shows where the rhinoceros was lying down (blue), where it stood (red) and where it walked (black).

Stellenbosch University https://scholar.sun.ac.za

6.6 Conclusion

This chapter explained the embedded firmware implementation of the real-time behaviour classification system. Detailed algorithms were provided for a linear support vector machine model implemented for rhinoceros behaviour classification. In the case of sheep a logistic regression model was implemented for the behaviour classification. A firmware flow diagram details the specific implementations. The implemented models were optimised for classification accuracy as shown in Chapter 5, however, the energy cost of feature extraction and classification did not form part of the optimisation. The next chapter shows how the classification models can be optimised both in terms of classification accuracy and energy consumption.

Stellenbosch University https://scholar.sun.ac.za

Chapter 7

Energy-aware feature and model optimisation

This chapter presents an energy-aware feature- and model selection technique for low-power embedded behaviour classification systems. A greedy sequential feature selection algorithm was utilised to minimise a cost function. This function incorporates a linear weighting of the energy expense of adding specific features and the change in classification error afforded by the added features. In addition, the energy expense of different classification algorithms was considered in selecting the optimal models.

7.1 Hardware

For this specific optimisation problem a measurement board was designed to enable the accurate measurement of the power consumed while calculating individual features and performing classification. The measurement board only includes a MSP430FR5739 low-power mixed signal microcontroller, a 24 MHz crystal oscillator and some passive components. The hardware layout is shown in Figure 7.1. The measurement board is based on the same microcontroller utilised in the biotelemetry tags as described in Chapter 4.1. However, this minimalistic design eliminates all energy consumption not related to the feature extraction and classification steps. All power measurements were carried out using a benchtop Textronic Digital Multimeter DMM 4020. First, the selection of the supply voltage and clock frequency that will result in the lowest energy consumption is described. Followed by the presentation of the optimisation technique.

Stellenbosch University https://scholar.sun.ac.za

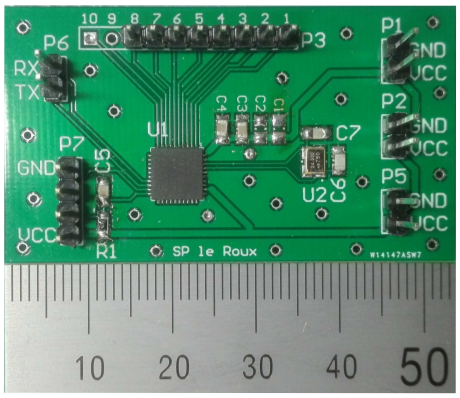


Figure 7.1: The MSP430FR5739 microcontroller board used for energy consumption measurements.

The power requirements of a CMOS processor are given by the following equation [91].

$$P = C_e V_{cc}^2 f + I_L V_{cc} \tag{7.1}$$

In Equation 7.1, C_e is the effective switching capacitance, which can be considered constant. V_{cc} is the MCU supply voltage, f is the clock frequency and I_L denotes the leakage current. From the equation we see that reducing V_{cc} to the minimum allowable voltage while keeping f constant significantly reduces P. V_{cc} is therefore selected to be $2\,\mathrm{V}$, which is the minimum operational voltage of the MSP430FR5739. At any particular supply voltage the power consumption increases with the clock frequency, but instructions are also executed more quickly. The energy consumed to complete a specific task is therefore considered. Since $E = \frac{P}{f}$, from Equation 7.1 we obtain,

$$E = C_e V_{cc}^2 + \frac{I_L V_{cc}}{f}. \tag{7.2}$$

The power consumption of the MSP430FR5739 was measured at different clock frequencies while keeping V_{cc} at $2\,\mathrm{V}$. Figure 7.2 shows the observed energy consumption of the MSP430FR5739 per clock cycle for different values of f, determined by dividing the measured power consumption by the clock frequency. For example, with $f = 20\,\mathrm{MHz}$ the power consumption P is $2.35\,\mathrm{mW}$ and the clock period $\frac{1}{f} = 50\,\mathrm{ns}$, resulting in the consumption of $0.118\,\mathrm{nJ}$ per clock cycle. Figure 7.2 exhibits the inverse proportionality predicted by Equation 7.2, and it is clear that selecting the highest clock frequency will result in the lowest energy consumption for a given task. Note that selecting the highest clock frequency does not always result in the lowest energy consumption since on some MCU architectures not all clock frequencies are available at the lowest supply voltage. In such a case, the reader is referred to the work of Dudacek et al. which suggests selecting the highest available clock frequency associated with the lowest operating voltage [91]. All measurements in the remainder

Stellenbosch University https://scholar.sun.ac.za

of this chapter were considered at $V_{cc} = 2\,\mathrm{V}$ and $f = 23.722\,\mathrm{MHz}$.

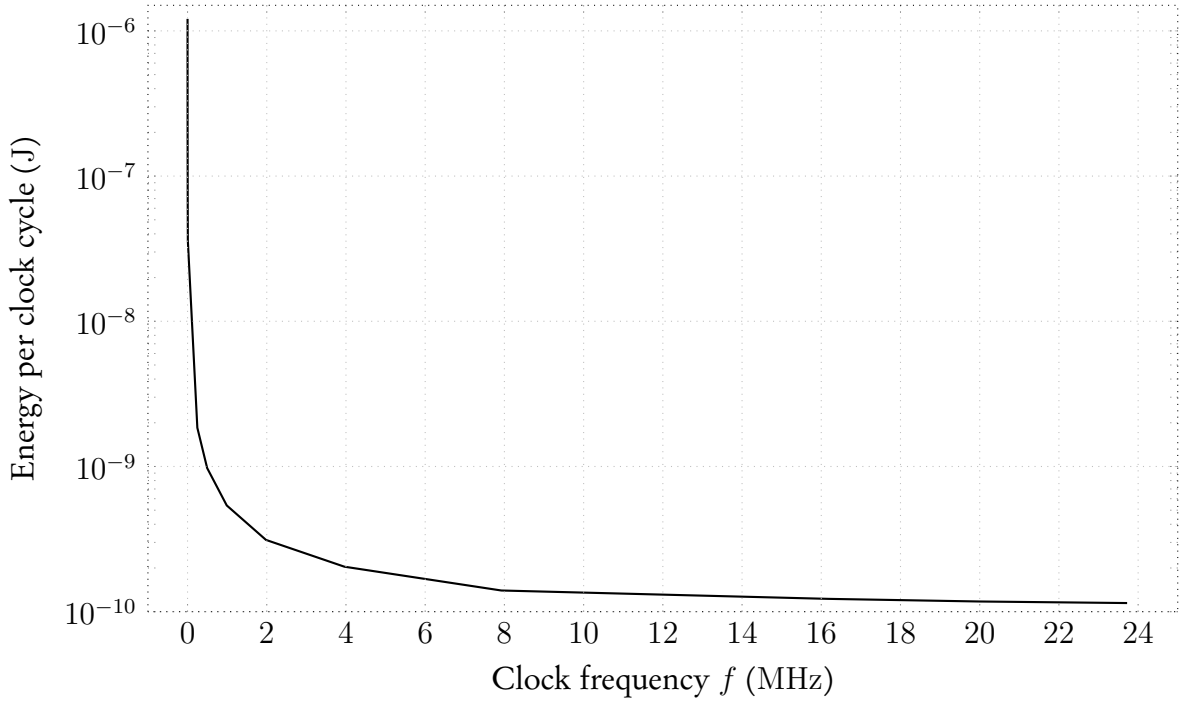


Figure 7.2: The energy consumed during each clock cycle of the MSP430FR5739 for various clock frequencies with V_{cc} set to 2 V.

7.2 Optimisation method

The aim is to explicitly and systematically balance the classification accuracy and energy consumption of different feature extraction- and classification techniques. The desired result is the optimal classification model and associated feature set for the rhinoceros and sheep datasets. In the following, the steps followed to achieve this goal are described. First we measure the energy expense of calculating each feature listed in Table 4.2 individually on the physical hardware. Subsequently, for each classification technique, a greedy sequential forward selection procedure, as described in Chapter 4.7, is employed to consider many different combinations of features while minimising a cost function given by Equation 7.3.

$$J(\lambda, E_f, E_{rr}) = \lambda E_f + (1 - \lambda) E_{rr} \tag{7.3}$$

Here, E_f is the energy expense of calculating a specific set of features, E_{rr} is the classification error achieved with that set of features and λ is a constant weight balancing the contributions of E_f and E_{rr} to the cost J. The cost function minimisation is performed for each classification technique and hyper-parameter combination listed in Chapter 3 and for all considered values of $\lambda \in \{0.0, 0.2, 0.5, 0.8, 1.0\}$. In each case, the feature combination

Stellenbosch University https://scholar.sun.ac.za

that result in the lowest cost J for each considered value of λ is recorded. A single baseline is selected by identifying the model with the lowest classification error for $\lambda = 0.0$. This baseline represents the best classification model that can be achieved using the conventional SFS approach. A shortlist is subsequently compiled by selecting those models with the lowest energy expense E_f for each classification technique whose classification error falls within a predefined margin of the baseline. Finally, the energy expense of classification, E_c, is measured for all models on the shortlist. The model with the lowest total energy expenditure, given by Equation 7.4, is selected as the final result.

$$E_t = E_f + E_c \tag{7.4}$$

7.3 Step 1: Per-feature energy measurement

Each of the features listed in Table 4.2 was individually programmed onto the MSP430FR5739 using the C programming language. For each feature, the computation time, the number of clock cycles and the current consumption were measured and the energy consumption calculated. Table 7.1 shows these measured values. This table is used in the subsequent optimisation step to calculate the energy expense of extracting different combinations of features E_f. For example, extracting the max(x), max(z), var(y) and en(x) would result in $E_f = 0.606 + 0.606 + 66.274 + 105.175 = 172.661\,\mu\text{J}$.

Although other metrics could be used to quantify the computational complexity of each feature, such as the number of clock cycles required during computation, the measured energy consumption will be used since it is independent of hardware specific characteristics. A more theoretical approach based on computational complexity of each feature expressed as the number of operations required for its calculation, would be affected by how differences in MCU architectures and their inherent executional routines. Direct energy measurements were found to be both quicker and guaranteed to reflect true energy expenditure.

Stellenbosch University https://scholar.sun.ac.za

Table 7.1: The energy consumed during the calculation of individual features on the MSP430FR5739 with $V_{cc} = 2\,\mathrm{V}$ and $f = 23.722\,\mathrm{MHz}$.

Feature	Time (ms)	Clock Cycles	Current (mA)	E_f (µJ)
Max	0.142	3377	2.129	0.606
Min	0.147	3496	2.068	0.609
Mean	1.322	31368	2.143	5.666
Mean distance between axes	1.960	46501	2.077	8.141
Standard deviation	8.254	195792	2.005	33.093
Variance	16.518	391844	2.006	66.274
Skewness	19.365	459381	1.938	75.063
Kurtosis	20.927	496435	1.840	77.011
Energy in 1 Hz bins	28.118	667021	1.793	100.818
Energy in the whole signal	29.170	691977	1.803	105.175
Pairwise correlation between axes	26.086	618818	2.082	108.617
Average signal magnitude	34.989	830016	2.094	146.534
Spectral entropy	58.287	1382696	1.958	228.287

7.4 Step 2: Optimisation algorithm

The optimisation algorithm is presented in Algorithm 4. The algorithm is similar to the one presented in Algorithm 1, however, to achieve energy-aware feature selection, a cost function is employed which weighs the energy expense of adding a feature against the change in error afforded by the additional feature, as given by Equation 7.3. Model optimisation and evaluation is done within a grouped, nested and repeated cross-validation framework, as explained in Chapter 4.7.1.

ALGORITHM 4: Optimisation Algorithm within a Grouped, Nested and Repeated Cross-Validation Procedure

procedure OPTIMISATION ALGORITHM($\mathbf{D}$, $C(\alpha)$)
  Divide the dataset $\mathbf{D}$ into K-folds: $\mathcal{T} = \{\mathbf{T}_1, \mathbf{T}_2, ..., \mathbf{T}_K\}$
  $\mathbf{E}_{rr} \leftarrow \varnothing$ # place-holder for test-set classification error
  for λ in {0.0, 0.2, 0.5, 0.8, 1.0} **do** # the set of λ to be considered
    for $\mathbf{T}_i$ in $\mathcal{T}$ **do**
      Define $\mathbf{O}_{train}$ as $\mathbf{D}$ without the fold $\mathbf{T}_i$
      Define $\mathbf{O}_{test}$ as the fold $\mathbf{T}_i$
      $\mathbf{F}_{store} \leftarrow \varnothing$ # place-holder for selected feature vectors
      for $N_{in} \leftarrow 1$ to 10 **do**
        Divide $\mathbf{O}_{train}$ randomly into (K-1)-folds: $\mathcal{V} = \{\mathbf{V}_1, \mathbf{V}_2, ..., \mathbf{V}_{K-1}\}$
        for $\mathbf{V}_j$ in $\mathcal{V}$ **do**

Stellenbosch University https://scholar.sun.ac.za

Define $\mathbf{I}_{train}$ as $\mathbf{O}_{train}$ without the fold $\mathbf{V}_j$
Define $\mathbf{I}_{val}$ as the fold $\mathbf{V}_j$
Sequential feature selection starts here
$\mathbf{f}_{sel} \leftarrow \varnothing$ # place-holder for features selected for iteration $\mathbf{V}_j$
$\mathbf{f}_{eval} \leftarrow \mathbf{f}_{all} = \{f_1, f_2, ..., f_d\}$ # all available features
for $n \leftarrow 1$ to d **do**
    $J_{min} \leftarrow 1000$ # some large value
    for f_{eval} in $\mathbf{f}_{eval}$ **do** # consider each feature in turn
        Define $\mathbf{I}'_{train}$ as $\mathbf{I}_{train}$ selecting only the features, $\mathbf{f}_{sel} + f_{eval}$
        Define $\mathbf{I}'_{val}$ as $\mathbf{I}_{val}$ selecting only the features, $\mathbf{f}_{sel} + f_{eval}$
        Build a statistical model $m' = m(\mathbf{I}'_{train}, C(\alpha))$
        Apply m' to $\mathbf{I}'_{val}$ and determine the classification error E_{rr}
        Calculate E_f, from Table 7.1, for the selected features, $\mathbf{f}_{sel} + f_{eval}$
        Evaluate the cost function $J = \lambda E_f + (1 - \lambda)E_{rr}$
        if $J < J_{min}$ **then**
            $J_{min} \leftarrow J$
            $f_{best} \leftarrow f_{eval}$
        end if
    end for
    Append f_{best} to $\mathbf{f}_{sel}$
    Remove f_{best} from $\mathbf{f}_{eval}$
end for
Append $\mathbf{f}_{sel}$ to $\mathbf{F}_{store}$ # store optimal set of features for iteration $\mathbf{V}_j$
# Sequential feature selection ends here
end for
end for
# This results in $(N_{in}) * (K - 1)$ stored sets of features in the matrix $\mathbf{F}_{store}$. # Hence $\mathbf{F}_{store}$ has the dimension $((N_{in}) * (K - 1)) \times d$
Merge $\mathbf{F}_{store}$ to produce a single optimal set of features $\mathbf{f}_i = \{\hat{f}_1, \hat{f}_2, ..., \hat{f}_d\}$ for fold $\mathbf{T}_i$
$\mathbf{f}_{test} \leftarrow \varnothing$
$\mathbf{e}_{rri} \leftarrow \varnothing$
# Determine classification error for fold $\mathbf{T}_i$
for f_i in $\mathbf{f}_i$ **do**
    Append f_i to $\mathbf{f}_{test}$
    Define $\mathbf{O}'_{train}$ as $\mathbf{O}_{train}$ with only the features $\mathbf{f}_{test}$ selected
    Define $\mathbf{O}'_{test}$ as the set $\mathbf{O}_{test}$ with only the features $\mathbf{f}_{test}$ selected
    Build a statistical model $m' = m(\mathbf{O}'_{train}, C(\alpha))$
    Apply m' to $\mathbf{O}'_{test}$ and append the error to $\mathbf{e}_{rri}$
end for

Stellenbosch University https://scholar.sun.ac.za

50: Append $\mathbf{e}_{rri}$ to $\mathbf{E}_{rr}$ # store the test-set results of fold $\mathbf{T}_i$
51: **end for**
52: Average the values in $\mathbf{E}_{rr}$ to produce the overall average cross-validated error $\mathbf{e}_{rr}$
53: # $\mathbf{e}_{rr}$ represents the average cross-validated classification error achieved using different number of features.
54: $F_N \leftarrow \text{argmin}_i(\mathbf{e}_{rr}^{(i)})$ # $i = 1, 2, ..., d$
55: $e_{rr} = \mathbf{e}_{rr}^{(F_N)}$
56: # e_{rr} is the lowest cross-validated classification error achieved with the optimal number of features F_N.
57: **end for**
58: **end procedure**

Optimisation is performed as follows. For each dataset $\mathbf{D}_S$ and $\mathbf{D}_R$ the algorithm is executed for all the classification models $C(\alpha)$ and for each value of λ. The sheep dataset is split into 15 outer-folds, since data were collected from 15 individual sheep, resulting in $\mathcal{T} = \{\mathbf{T}_1, \mathbf{T}_2, ..., \mathbf{T}_{K=15}\}$. Similarly, the rhinoceros data is split into three outer-folds with $\mathcal{T} = \{\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_{K=3}\}$. Lines 14 to 35 describe the SFS process and shows how the cost function J is used to sort the features into an order of diminishing utility. Line 22 shows how a statistical model is trained using $\mathbf{I}_{train}$ which consists of a subset of the full set of features. This trained model m' is subsequently used to calculate performance on the validation-set $\mathbf{I}_{val}$ using the same subset of features, as shown in line 23. The cost function J is used to order the features with the aim of maximising the model performance at every step, as shown in lines 25 to 28. Subsequently, a feature vector $\mathbf{f}_{sel}$ is produced for each inner fold $\mathbf{V}_j$ and for every inner-loop iteration N_{in}. These feature vectors are merged, using the average rank of each feature, to produce a single feature vector $\mathbf{f}_i$ for fold $\mathbf{T}_i$, as shown in line 39. The test-set error is subsequently determined by evaluating the trained model on the held-out data partition $\mathbf{O}_{test}$, as shown in lines 42 to 50. As a final step, the mean cross-validated error $\mathbf{e}_{rr}$ is calculated by averaging over all the individual results obtained from each test-set partition. We report the average rank of the top performing features, the optimal number of selected features F_N and the lowest cross-validated error e_{rr} achieved with the optimal number of features F_N for each dataset and classification model evaluated.

To illustrate the optimisation process and the trade-off between classification error and energy consumption let us consider a logistic regression classifier applied to the sheep dataset, as an example. Figure 7.3 shows the dependence of the cost function given in Equation 7.3 on the value of λ when features are added sequentially using SFS. The greedy nature of SFS adds features in a way that minimises the cost of each additional feature. Initially, the cost is high due to the high error associated with a small number of features. The cost decreases to

Stellenbosch University https://scholar.sun.ac.za

a minimum (indicated with a dot on the graph) as more features are added and subsequently increases again as a result of the higher energy consumption. Different values of λ lead to a different compromise between accuracy and energy expenditure and also in general lead to different selections of features. The effect of λ is further illustrated in Figure 7.4, in which E_f and E_{rr} are plotted separately for the minimum cost J in Figure 7.3. Low values of λ lead to classifiers with low classification error but high energy consumption, whereas high values of λ lead to classifiers with low energy consumption but poor classification performance. Figure 7.3 also shows that for a specific classifier we obtain an optimal point for each value of λ. The results obtained at these optimal points of J are reported for each classification technique and each value of λ.

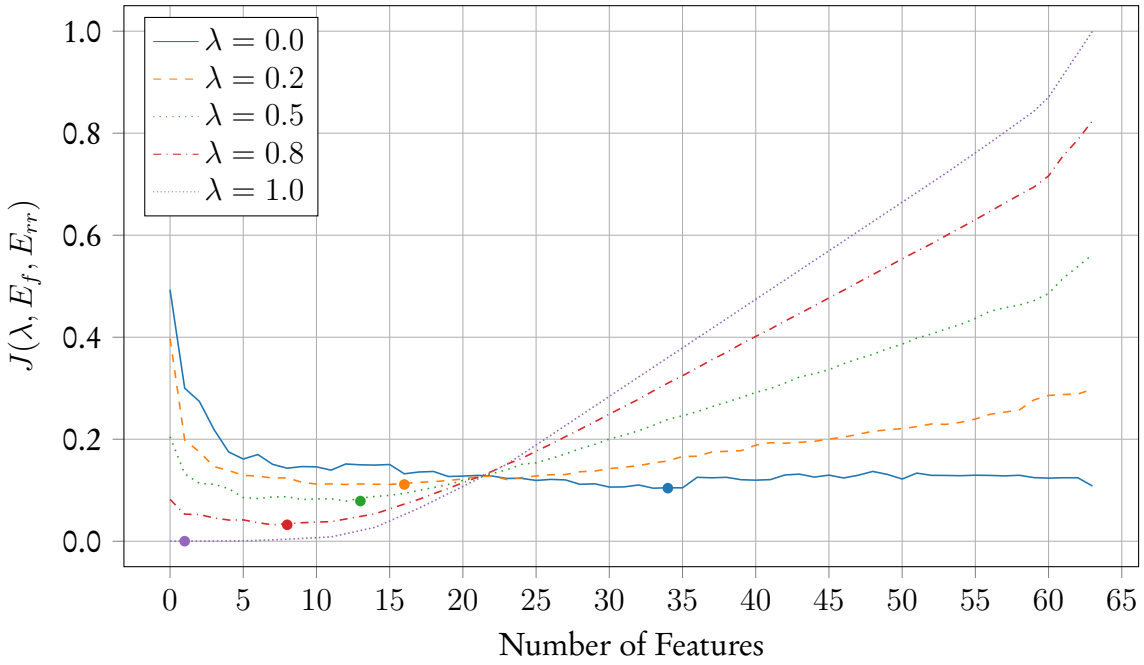


Figure 7.3: The cost function values achieved using a logistic regression classifier applied to the sheep dataset for different number of features and each value of λ.

Stellenbosch University https://scholar.sun.ac.za

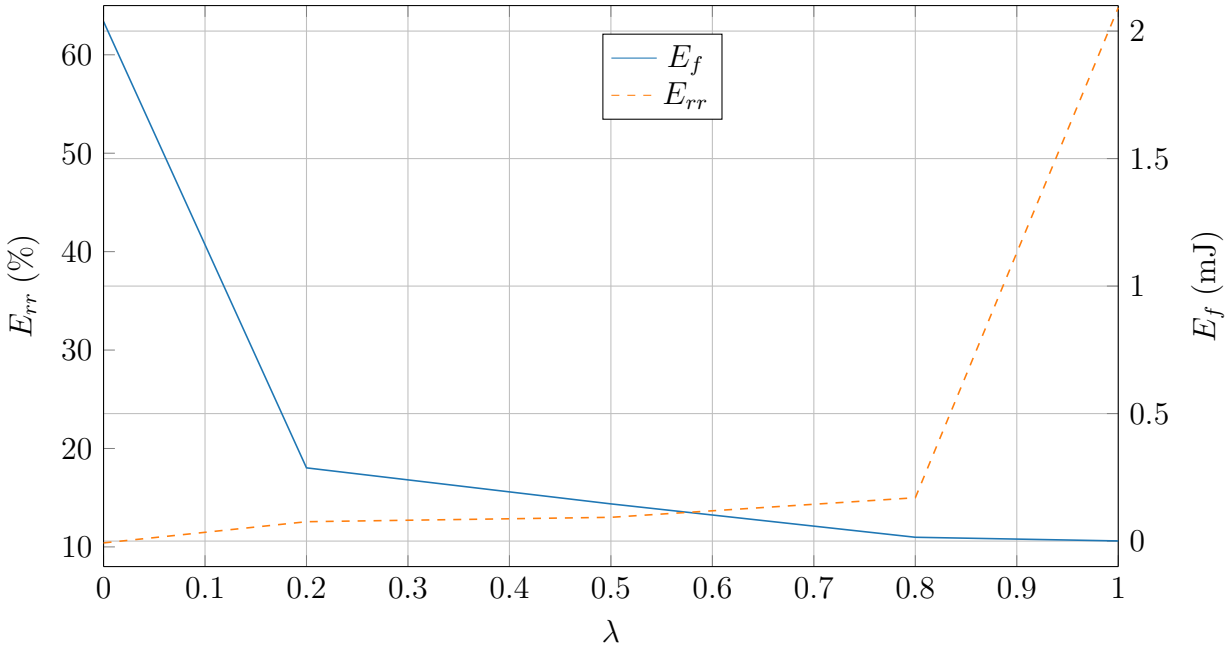


Figure 7.4: The relationship between E_f, E_{rr} and λ for a logistic regression classifier trained on the sheep dataset.

The results obtained for the rhinoceros and sheep datasets are shown in Tables 7.2 and 7.3, respectively. The tables list the different classification models and associated model hyper-parameters (in parentheses), the optimal-number of selected features F_N, the lowest cross-validated error e_{rr} achieved at the optimal number of features and the energy consumed E_f when calculating these features.

7.5 Step 3: Energy-aware classifier selection

The procedure described in Chapter 7.4 results in a set of features that is optimal in terms of Equation 7.3 for a each classification technique, hyper-parameter combination and value of λ. However, this only considers the energy expense of feature extraction and does not include the energy expense of the classification algorithm employed. To take the energy expense of the classification algorithm into account, we proceed as follows. First, using the results obtained for $\lambda = 0.0$, the classification technique that achieves the lowest overall classification error is identified and highlighted (in dark grey) in Tables 7.2 and 7.3. This represents the minimum error that can be achieved by classical SFS with $J = E_{rr}$, when the energy cost of computing features is not considered. Next, an acceptable margin of deviation from this baseline is specified. A shortlist is then compiled for further consideration by selecting models that achieve classification accuracies within this margin while also achieving the lowest

Stellenbosch University https://scholar.sun.ac.za

Table 7.2: Energy-aware feature selection results for the rhinoceros dataset. The rows depict the result of optimising J by SFS for each considered classifier and hyper-parameter combination, as identified in the first column. The remaining columns indicate the number of features at this optimum, the classification error achieved using the selected features and the energy cost of computing the features for the considered values of λ.

Classifier $C(\alpha)$	$\lambda = 0.0$ F_N	$\lambda = 0.0$ e_{rr} (%)	$\lambda = 0.0$ E_f (uJ)	$\lambda = 0.2$ F_N	$\lambda = 0.2$ e_{rr} (%)	$\lambda = 0.2$ E_f (uJ)	$\lambda = 0.5$ F_N	$\lambda = 0.5$ e_{rr} (%)	$\lambda = 0.5$ E_f (uJ)	$\lambda = 0.8$ F_N	$\lambda = 0.8$ e_{rr} (%)	$\lambda = 0.8$ E_f (uJ)	$\lambda = 1.0$ F_N	$\lambda = 1.0$ e_{rr} (%)	$\lambda = 1.0$ E_f (uJ)
LSVM	**11**	**0.39**	**664.47**	**7**	**0.53**	**9.31**	7	0.53	9.31	7	0.53	9.31	1	5.52	0.61
LDA	6	0.49	373.40	8	1.03	14.98	**5**	**0.93**	**8.10**	7	0.99	9.31	1	7.08	0.61
QDA	1	7.22	0.61	1	7.22	0.61	1	7.22	0.61	2	6.10	1.22	1	7.22	0.61
DT (1)	2	35.29	152.20	7	38.12	9.31	7	38.12	9.31	1	38.51	0.61	1	38.51	0.61
DT (2)	38	3.17	2675.44	6	3.20	8.70	5	3.20	8.09	**3**	**0.83**	**1.82**	1	5.66	0.61
DT (3)	2	3.20	6.27	2	3.20	6.27	2	3.20	6.27	4	0.35	2.43	1	5.60	0.61
DT (4)	2	3.20	6.27	2	3.20	6.27	2	3.20	6.27	4	0.35	2.43	1	5.53	0.61
DT (5)	2	3.20	6.27	2	3.20	6.27	2	3.20	6.27	4	2.13	2.43	1	5.70	0.61
DT (6)	2	3.20	6.27	2	3.20	6.27	2	3.20	6.27	4	2.13	2.43	1	8.51	0.61
DT (7)	2	3.20	6.27	2	3.20	6.27	2	3.20	6.27	4	2.13	2.43	1	6.74	0.61
DT (8)	2	3.20	6.27	2	3.20	6.27	2	3.20	6.27	4	2.13	2.43	1	6.91	0.61
DT (9)	2	3.20	6.27	2	3.20	6.27	2	3.20	6.27	4	2.13	2.43	1	6.98	0.61
DT (10)	2	3.20	6.27	2	3.20	6.27	2	3.20	6.27	4	2.13	2.43	1	7.02	0.61
DT (∞)	2	3.20	6.27	2	3.20	6.27	2	3.20	6.27	4	2.13	2.43	1	6.93	0.61
k-NN (1)	2	2.07	1.22	3	0.34	1.82	3	0.34	1.82	**3**	**0.34**	**1.82**	1	6.93	0.61
k-NN (2)	14	1.07	728.65	3	0.67	1.82	3	0.67	1.82	3	0.67	1.82	1	7.98	0.61
k-NN (3)	2	2.08	6.27	3	0.49	1.82	3	0.49	1.82	3	0.49	1.82	1	6.01	0.61
k-NN (4)	7	1.24	253.39	5	0.94	8.09	5	0.94	8.09	4	1.92	2.43	1	7.85	0.61
k-NN (5)	7	1.28	253.39	4	0.41	7.49	4	0.41	7.49	4	1.80	2.43	1	6.39	0.61
k-NN (6)	7	0.67	253.39	5	0.58	8.09	5	0.58	8.09	4	1.28	2.43	1	7.82	0.61
k-NN (7)	7	0.75	253.39	5	0.64	8.09	5	0.64	8.09	4	1.31	2.43	1	6.91	0.61
k-NN (8)	7	0.59	253.39	5	0.56	8.09	5	0.56	8.09	4	0.88	2.43	1	7.39	0.61
k-NN (9)	5	1.85	252.17	5	0.61	8.09	5	0.61	8.09	4	0.88	2.43	1	7.00	0.61
k-NN (10)	9	0.91	325.33	5	0.58	8.09	5	0.58	8.09	4	0.86	2.43	1	7.45	0.61
RF (1)	33	13.92	2373.83	9	4.48	38.19	9	4.48	38.19	8	4.48	30.05	1	28.70	0.61
RF (2)	6	3.17	371.15	5	3.12	8.09	5	3.12	8.09	**2**	**0.67**	**1.22**	1	5.66	0.61
RF (3)	47	3.09	3582.08	5	3.12	8.09	5	3.12	8.09	4	0.34	2.43	1	5.78	0.61
RF (4)	40	3.12	2877.07	5	3.12	8.09	5	3.12	8.09	4	0.34	2.43	1	5.90	0.61
RF (5)	30	3.15	2139.12	5	3.12	8.09	5	3.12	8.09	4	0.38	2.43	1	5.74	0.61
RF (6)	30	3.15	2139.12	5	3.12	8.09	5	3.12	8.09	4	0.50	2.43	1	6.10	0.61
RF (7)	30	3.15	2139.12	5	3.12	8.09	5	3.12	8.09	4	0.50	2.43	1	6.30	0.61
RF (8)	30	3.15	2139.12	5	3.12	8.09	5	3.12	8.09	4	0.50	2.43	1	6.25	0.61
RF (9)	30	3.15	2139.12	5	3.12	8.09	5	3.12	8.09	4	0.50	2.43	1	6.36	0.61
RF (10)	30	3.15	2139.12	5	3.12	8.09	5	3.12	8.09	4	0.50	2.43	1	6.39	0.61
RF (∞)	30	3.15	2139.12	5	3.12	8.09	5	3.12	8.09	4	0.50	2.43	1	6.37	0.61
LR	10	0.49	559.30	5	0.45	8.09	5	0.45	8.09	**5**	**1.11**	**3.04**	1	5.53	0.61
NB	1	7.22	0.61	1	7.22	0.61	1	7.22	0.61	1	7.22	0.61	1	7.22	0.61

energy expense of feature computation E_f. The selected classifiers and their feature sets are then individually implemented on the microcontroller and the energy consumption of each measured. The measured classification energy expenditure E_c is then added to E_f to determine the total energy, $E_t = E_f + E_c$, of a single classification decision for each classification technique. The model with the lowest total energy consumption E_t is the final result.

The reader might wonder why this grid-search step, of compiling a shortlist, is performed manually and why it is not included in the optimisation algorithm. This is because it is not known before-hand which feature will be selected as optimal for each classification technique. The energy consumption associated with that specific classification technique and feature set is therefore also unknown. To find these values would require energy measurements for each combination of features, for each classification technique and hyper-parameter combination. This will result in a multitude of measurements which is time consuming and not feasible when a large number of features and classifiers are considered. However, by compiling and

Stellenbosch University https://scholar.sun.ac.za

Table 7.3: Energy-aware feature selection results for the sheep dataset. The rows depict the result of optimising J by SFS for each considered classifier and hyper-parameter combination, as identified in the first column. The remaining columns indicate the number of features at this optimum, the classification error achieved using the selected features and the energy cost of computing the features for the considered values of λ.

Classifier $C(\alpha)$	$\lambda = 0.0$ F_N	$\lambda = 0.0$ e_{rr} (%)	$\lambda = 0.0$ E_f (uJ)	$\lambda = 0.2$ F_N	$\lambda = 0.2$ e_{rr} (%)	$\lambda = 0.2$ E_f (uJ)	$\lambda = 0.5$ F_N	$\lambda = 0.5$ e_{rr} (%)	$\lambda = 0.5$ E_f (uJ)	$\lambda = 0.8$ F_N	$\lambda = 0.8$ e_{rr} (%)	$\lambda = 0.8$ E_f (uJ)	$\lambda = 1.0$ F_N	$\lambda = 1.0$ e_{rr} (%)	$\lambda = 1.0$ E_f (uJ)
LDA	28	14.98	2288.20	7	16.96	457.25	7	22.47	83.77	5	27.02	8.10	1	67.09	0.61
QDA	48	11.96	3773.72	14	12.73	420.83	10	15.03	28.79	8	15.27	14.98	1	68.44	0.61
DT (1)	7	65.07	339.19	4	63.16	12.55	4	63.16	12.55	3	65.50	1.82	1	66.45	0.61
DT (2)	19	41.90	1554.49	8	41.27	30.05	8	41.27	30.05	2	42.58	1.22	1	65.29	0.61
DT (3)	24	20.05	1929.32	14	23.65	318.30	9	25.13	35.71	5	25.13	8.10	1	69.44	0.61
DT (4)	18	13.94	3582.08	16	19.14	63.14	9	20.5	25.6	5	19.28	10.57	1	68.46	0.61
DT (5)	47	13.39	1549.76	9	19.12	103.11	9	18.23	36.93	5	17.11	14.98	1	68.22	0.61
DT (6)	54	12.17	4183.44	14	17.49	143.74	5	17.10	10.58	5	17.46	10.57	1	67.16	0.61
DT (7)	61	12.74	5015.92	15	17.03	144.35	8	18.04	22.51	5	16.77	10.57	1	68.16	0.61
DT (8)	47	12.99	3478.42	**28**	**12.03**	**1258.60**	11	17.79	44.46	6	17.99	11.18	1	66.31	0.61
DT (9)	64	12.94	5294.57	13	18.85	135.60	10	18.56	36.32	4	19.25	2.43	1	66.67	0.61
DT (10)	59	14.17	4686.81	24	16.26	905.60	4	20.05	2.43	6	21.21	3.65	1	67.44	0.61
DT (∞)	64	19.35	5294.57	17	20.07	311.44	5	23.40	8.10	5	23.53	8.10	1	68.44	0.61
k-NN (1)	23	14.70	1019.73	17	16.16	311.44	9	18.66	28.18	10	18.22	28.79	1	68.43	0.61
k-NN (2)	46	12.86	3565.89	19	14.02	443.99	11	16.73	36.93	9	17.60	23.12	1	69.71	0.61
k-NN (3)	48	13.29	3767.53	19	14.44	443.99	11	16.77	36.93	7	17.26	14.37	1	67.73	0.61
k-NN (4)	49	11.99	3868.35	19	13.61	443.99	10	15.20	36.32	9	15.94	20.65	1	68.08	0.61
k-NN (5)	27	13.04	1549.76	13	15.52	103.11	11	15.56	36.93	8	15.82	14.98	1	66.76	0.61
k-NN (6)	13	12.81	748.74	18	13.95	377.71	10	15.34	28.79	8	15.19	14.98	1	67.47	0.61
k-NN (7)	27	13.04	748.74	13	15.52	377.71	11	15.56	28.79	8	15.82	14.98	1	66.76	0.61
k-NN (8)	13	12.81	1650.06	18	13.95	144.35	10	15.34	10.57	8	15.19	10.57	1	67.47	0.61
k-NN (9)	27	12.67	1549.76	10	15.60	88.70	11	15.36	36.93	7	15.69	9.31	1	67.00	0.61
k-NN (10)	27	12.77	1549.76	18	13.78	377.71	11	15.60	36.93	7	15.57	9.31	1	67.40	0.61
RF (1)	8	53.23	567.48	14	54.45	424.40	10	54.45	118.31	5	56.96	13.15	1	65.66	0.61
RF (2)	34	32.43	2660.41	19	37.32	718.09	9	39.64	88.09	2	42.51	1.22	1	64.73	0.61
RF (3)	60	16.67	4892.01	15	18.90	372.09	8	24.04	89.44	3	26.34	1.82	1	71.07	0.61
RF (4)	16	13.75	1122.36	12	17.41	171.90	6	19.35	11.18	5	18.70	10.57	1	70.90	0.61
RF (5)	58	13.58	4690.37	10	17.91	38.80	5	17.46	10.57	5	17.47	10.57	1	70.76	0.61
RF (6)	25	13.03	5294.57	15	16.51	111.26	5	16.13	10.57	5	16.09	10.57	1	71.06	0.61
RF (7)	64	14.26	3906.74	14	15.97	245.17	5	15.87	16.84	5	15.56	17.45	1	69.11	0.61
RF (8)	51	12.66	4410.11	**16**	**12.60**	**529.79**	7	15.69	44.46	8	16.16	1.82	1	67.54	0.61
RF (9)	56	13.01	4259.65	20	13.15	529.79	11	16.99	2.43	3	19.38	3.65	1	66.78	0.61
RF (10)	53	12.62	5294.57	20	13.77	386.5	4	20.29	1.82	6	19.51	2.43	1	67.03	0.61
RF (∞)	64	15.48	5294.57	18	17.56	386.50	3	21.08	1.82	4	21.03	2.43	1	68.11	0.61
LR	**34**	**10.41**	**2037.40**	**16**	**12.56**	**287.14**	13	13.01	145.89	8	14.99	14.98	1	64.78	0.61
LSVM	45	11.77	3293.83	**9**	**11.60**	**300.47**	8	15.09	17.45	9	15.20	23.12	1	66.42	0.61
NB	27	12.67	1768.54	8	15.69	116.86	6	16.35	78.10	5	20.63	3.04	1	68.40	0.61

shortlist of optimal models, our technique drastically reduces the number of physical measurements that need to be performed without compromising model performance.

7.5.1 Shortlist for the rhinoceros dataset

Table 7.2 shows that, for the rhinoceros dataset, when $\lambda = 0.0$, the lowest cross-validated classification error e_{rr} is $0.39\,\%$ and is achieved by a LSVM using 11 features. Using this as baseline, we determine the lowest energy consumption E_f that can be achieved by each classification technique while maintaining classification performance within a $0.75\,\%$ margin of this error. This identified a shortlist of seven models highlighted (in light grey) in Table 7.2 for consideration, namely: LSVM ($\lambda = 0.0$), LSVM ($\lambda = 0.2$), LDA ($\lambda = 0.5$), k-NN (1) ($\lambda = 0.8$), DT (2) ($\lambda = 0.8$), RF (2) ($\lambda = 0.8$) and LR ($\lambda = 0.8$). Table 7.4 reveals the specific features selected for each of the seven models shortlisted by the optimisation. We note that the baseline includes the largest number of selected features, and that these include both low and high energy cost features, for example, max(x) and pse(x) respectively. In contrast, RF (2)

Stellenbosch University https://scholar.sun.ac.za

($\lambda = 0.8$) selects only two low-cost features while maintaining excellent model performance, with $e_{rr} = 0.67\,\%$ which is $0.28\,\%$ of the baseline, while consuming $\frac{664.47}{1.22} = 528$ times less energy.

Table 7.4: The optimal feature set selected for each of the seven classification techniques shortlisted from Table 7.2 for the rhinoceros dataset. The rows are ordered in ascending E_f and the baseline system is highlighted.

Classifier	F_N	$e_{rr}(\%)$	E_f	Selected features in order of importance
RF (2)	2	0.67	1.22	max(x), min(x)
DT (2)	3	0.83	1.82	max(x), min(x), max(y)
K-NN (1)	3	0.34	1.82	max(x) min(x) max(y)
LR	5	1.11	3.04	max(x), min(x), min(y), max(z), max(y)
LDA	5	0.93	8.10	max(x), min(y), min(x), mean(x), min(z)
LSVM	7	0.53	9.31	max(x), max(y), mean(x), min(x), max(z), min(y), min(z)
LSVM	11	0.39	664.47	max(x), mean(x), asm(), min(x), kurt(x), var(x), std(x), en(x), pse(x), max(y), min(y)

7.5.2 Shortlist for the sheep dataset

For the sheep dataset, Table 7.3 shows that when $\lambda = 0.0$ the lowest cross-validated classification error e_{rr} is $10.41\,\%$ and is achieved by logistic regression using 34 features. With this as baseline, we determine the lowest energy consumption E_f that can be achieved while maintaining a classification accuracy within a $2.25\,\%$ margin of this error. This identified a shortlist of five models for further consideration and is highlighted in Table 7.3 (in light grey), namely: LR ($\lambda = 0.0$), LR ($\lambda = 0.2$), DT (8) ($\lambda = 0.2$), RF (8) ($\lambda = 0.2$) and LSVM ($\lambda = 0.2$). Table 7.5 shows the specific features selected for each of the shortlisted models. We again note that the baseline uses the largest number of features, and that these again include both low and high energy cost features. In contrast, LSVM ($\lambda = 0.2$) selected nine fairly low-cost features, which results in a $1.19\,\%$ drop in classification performance while consuming $\frac{2037.40}{300.47} = 6.78$ times less energy than the baseline.

Stellenbosch University https://scholar.sun.ac.za

Table 7.5: The optimal feature set selected for each of the five classification techniques shortlisted from Table 7.3 for the sheep dataset. The ordering is in ascending E_f and the baseline system is highlighted.

Classifier	F_N	$e_{rr}(\%)$	E_f	Selected features in order of importance
LR	16	12.56	287.14	min(z), min(y), max(y), min(x), max(x), mean(y), dist(x,y), dist(y,z), mean(z), max(z), dist(x,z), mean(x), bin_3(x), skew(y), std(y), std(x)
LSVM	9	11.60	300.47	mean(y), min(y), min(z), bin_4(z), bin_3(x), skew(y), max(y), dist(y,z), dist(x,z)
RF (8)	16	12.60	529.79	min(y), max(z), max(x), mean(z), mean(x), min(z), dist(x,z), std(y), dist(x,y), std(x), dist(y,z), mean(y), min(x), max(y), std(z), bin_3(x)
DT (8)	28	12.03	1258.60	min(y), dist(x,y), mean(x), mean(x), min(x), max(x), min(z) std(x), dist(y,z), mean(y), dist(x,z), max(z), std(y), std(z), max(y), var(z), var(x), skew(z), kurt(z), bin_0(x), var(y), bin_9(x), corr(x,z), kurt(x), bin_9(y), skew(y), bin_1(x), bin_3(x)
LR	34	10.41	2037.40	skew(y), bin_3(x), var(x), min(z), min(y), var(z), var(y), mean(y), std(y), std(x), max(z), mean(z), asm(), bin_5(x), bin_7(x), dist(y,z), min(x), dist(x,y), bin_9(x), max(y), mean(x), bin_8(x), bin_4(z), kurt(y), std(z), skew(z), max(x), bin_4(x), corr(x,z), bin_6(x), bin_5(y), dist(x,z), kurt(x), pse(z)

7.5.3 Final Classifier Selection

Finally, the energy required by the classification algorithm itself is included in the optimisation for the seven rhinoceros models and the five sheep models on the respective shortlists. Each of these 12 trained classifiers were implemented on the hardware described earlier in order to accurately measure the energy E_c they consume. Table 7.6 depicts E_c for each of the seven models identified for the rhinoceros dataset. From the table we see that the DT classifier has the lowest energy consumption, while the k-NN classifier consumes three orders of magnitude more energy than the other candidates. Table 7.7 shows the corresponding measured energy consumption for the sheep dataset. In this case, the DT classifier also has the lowest energy consumption.

The total energy consumption is given by the sum of the energy consumed during feature extraction and during the classification process, as given by Equation 7.4. Conventional optimisation models only consider the energy expense of feature extraction E_f and does not include the cost of the classifier itself. As a result, models that are popular and easy to implement are often selected. For example, the k-NN classifier is a popular model and is often used as shown in Chapter 2. From Table 7.4 it is clear that if the k-NN classifier is selected, it

Stellenbosch University https://scholar.sun.ac.za

Table 7.6: The energy consumed by each shortlisted classification technique for the rhinoceros dataset. Measurements were performed on the MSP430FR5739 with $V_{cc} = 2\,\mathrm{V}$ and $f = 23.722\,\mathrm{MHz}$. The rows are ordered in ascending E_c and the baseline system is highlighted.

Classifier $C(\alpha)$	F_N	e_{rr} (%)	Time (us)	Clock Cycles	I (mA)	E_c (µJ)
DT (2)	3	0.83	21.76	516	2.19	0.095
LDA	5	0.93	96.82	2297	2.24	0.434
LSVM	7	0.53	124.17	2946	2.26	0.561
RF (2)	2	0.67	140.20	3326	2.16	0.606
LSVM	11	0.39	181.82	4314	2.24	0.815
LR	5	1.11	371.03	177985	2.14	1.588
K-NN (1)	3	0.34	1033600	24519059	2.03	4196

Table 7.7: The energy consumed by each shortlisted classification technique for the sheep dataset. Measurements were performed on the MSP430FR5739 with $V_{cc} = 2\,\mathrm{V}$ and $f = 23.722\,\mathrm{MHz}$. The rows are ordered in ascending E_c and the baseline system is highlighted.

Classifier $C(\alpha)$	F_N	e_{rr} (%)	Time (us)	Clock Cycles	I (mA)	E_c (µJ)
DT (8)	28	12.03	78.62	1865	2.19	0.344
LSVM	9	11.60	251.49	5966	2.25	1.132
RF (8)	16	12.60	1040.54	24684	2.19	4.556
LR	16	12.56	1522.00	36105	2.12	6.453
LR	34	10.41	2891.40	68590	2.15	12.433

would have resulted in a significant reduction in E_f and also achieved a very low classification accuracy. However, when the energy cost of classification E_c is additionally considered, we will later see from Table 7.8 that the k-NN is actually the worst model that can be selected. It is therefore important to select the model with the lowest total energy consumption E_t as the final result.

7.5.4 Optimal model for the rhinoceros dataset

Table 7.8 lists this total energy expenditure E_t for each classification model, as well as the cross-validated error e_{rr}, the change in error Δe_{rr} and the factor by which the total energy consumption has been reduced relative to the baseline for each of the shortlisted models evaluated on the rhinoceros dataset. The model with the lowest energy consumption is selected as the final result. For the rhinoceros, a random forest classifier with a maximum tree depth of two, achieves an overall classification accuracy of $99.33\,\%$ between three behavioural classes. This must be compared with the non-energy-aware baseline, which achieves a classification accuracy of $99.61\,\%$. However, the RF (2) classifier consumes 363 times less energy than the baseline LSVM, while only sacrificing $0.28\,\%$ in terms of classification accuracy.

Stellenbosch University https://scholar.sun.ac.za

Table 7.8: Total energy consumption of both feature extraction and classification, for each classification technique shortlisted for the rhinoceros dataset. The rows are ordered in ascending E_t and the baseline system is highlighted.

$C(\alpha)$	F_N	E_t (µJ)	$e_{rr}(\%)$	$\Delta e_{rr}(\%)$	E Ratio
RF (2)	2	1.83	0.67	0.28	363.6
DT (2)	3	1.92	0.83	0.44	346.5
LR	5	4.63	1.11	0.72	143.7
LDA	5	8.53	0.93	0.54	78.0
LSVM	7	9.87	0.53	0.14	67.4
LSVM	11	665.29	0.39	0.00	1.0
KNN (1)	3	4197.82	0.34	-0.05	0.2

Table 7.9a shows the precision, recall and $F1$ scores achieved with the optimal RF (2) model for the individual classes. Table 7.9b shows the confusion matrix for the model. From the confusion matrix we see that a very small degree of confusion exists between classes. Table 7.9c lists the two features selected for this model. Furthermore, Figure 7.5 shows the results obtained from the sequential feature selection process. In the figure, the solid line indicates the mean cross-validated accuracy and the error bars depict the range of test-set accuracies achieved for a specific number of features. Selecting the first two features result in high classification accuracy and there are no other alternatives that will achieve similar results with a lower number of features, as is the case for most classifiers presented in Chapter 5, when optimising for accuracy alone.

Stellenbosch University https://scholar.sun.ac.za

Table 7.9: The performance of a random forest classifier with a maximum tree depth of two, utilising two optimal features, evaluated on the rhinoceros dataset.

(a) The per-class precision, recall and $F1$ score.

P_r	R_e	$F1$	Class	F_N
0.948	0.992	0.970	Lie	2
0.994	0.999	0.997	Stand	2
0.995	0.945	0.969	Walk	2

(b) The per-class confusion-matrix. The rows depict true labels and the columns depict predicted labels.

	Lie	Stand	Walk
Lie	884	3	4
Stand	1	890	0
Walk	47	2	842

(c) The 2 optimally selected features.

max(x), min(x)

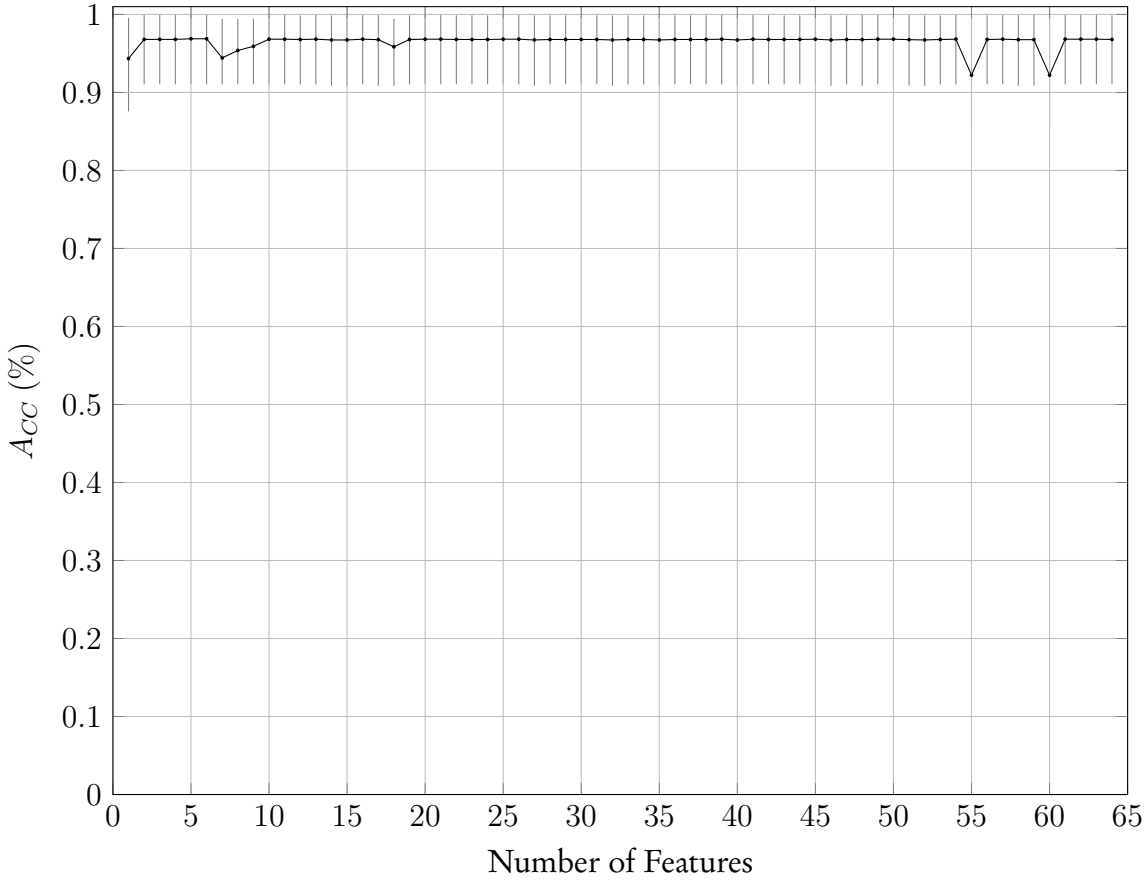


Figure 7.5: The mean cross-validated test-set accuracies achieved utilising optimal features and a random forest classifier evaluated on the rhinoceros dataset. The error-bars indicate the maximum and minimum test-set accuracies achieved for the specific number of features.

Stellenbosch University https://scholar.sun.ac.za

7.5.5 Optimal model for the sheep dataset

Table 7.10 similarly list this total energy expenditure E_t for each classification model, as well as the cross-validated error e_{rr}, the change in error Δe_{rr} and the factor by which the total energy consumption has been reduced relative to the baseline for each of the respective models evaluated on the sheep dataset. The model with the lowest energy consumption is selected as the final optimal model. In the case of sheep, a LR classifier with 16 features can result in a 6.98 times reduction in the total energy consumption with a classification accuracy of $87.44\,\%$. Although this is a $2.15\,\%$ drop from the $89.59\,\%$ achieved by the baseline, it emphasises the care that needs to be taken by the system designer to consider the particular constraints of the application. If accuracy is of overwhelmingly importance, Table 7.10 shows that the drop in classification accuracy can be narrowed to $1.19\,\%$ by choosing a LSVM, while still achieving a 6.8-fold reduction in consumed energy. For this reason the latter LSVM was chosen as the final result.

Table 7.10: Total energy consumption of both feature extraction and classification, for each classification technique shortlisted for the sheep dataset. The rows are ordered in ascending E_t and the baseline system is highlighted.

$C(\alpha)$	F_N	E_t (µJ)	$e_{rr}(\%)$	$\Delta e_{rr}(\%)$	E Ratio
LR	16	293.59	12.56	2.15	7.0
LSVM	9	301.60	11.60	1.19	6.8
RF (8)	16	534.35	12.60	2.19	3.8
DT (8)	28	1258.94	12.03	1.62	1.6
LR	34	2049.83	10.41	0.00	1.0

Table 7.11a shows the precision, recall and $F1$ scores achieved with the optimal LSVM model for the individual classes. Table 7.11b shows the confusion matrix for the LSVM model. From the confusion matrix we see that a small degree of confusion exists between classes. Table 7.11c lists the nine features selected for this model. Furthermore, Figure 7.6 contains the results obtained from the sequential feature selection process.

Stellenbosch University https://scholar.sun.ac.za

Table 7.11: The performance of a linear support vector machine, utilising nine optimal features, evaluated on the sheep dataset.

(a) The per-class precision, recall and $F1$ score.

P_r	R_e	$F1$	Class	F_N
0.748	0.903	0.818	Lie	9
0.916	0.960	0.938	Stand	9
0.817	0.808	0.813	Walk	9
0.992	0.994	0.993	Run	9
0.737	0.557	0.635	Graze	9

(b) The per-class confusion-matrix. The rows depict true labels and the columns depict predicted labels.

	Lie	Stand	Walk	Run	Graze
Lie	901	0	1	0	96
Stand	3	953	36	1	0
Walk	0	83	802	7	101
Run	0	0	6	987	0
Graze	300	4	136	0	553

(c) The 9 optimally selected features.

mean(y), min(y), min(z), bin_4(z), bin_3(x), skew(y), max(y), dist(y,z), dist(x,z)

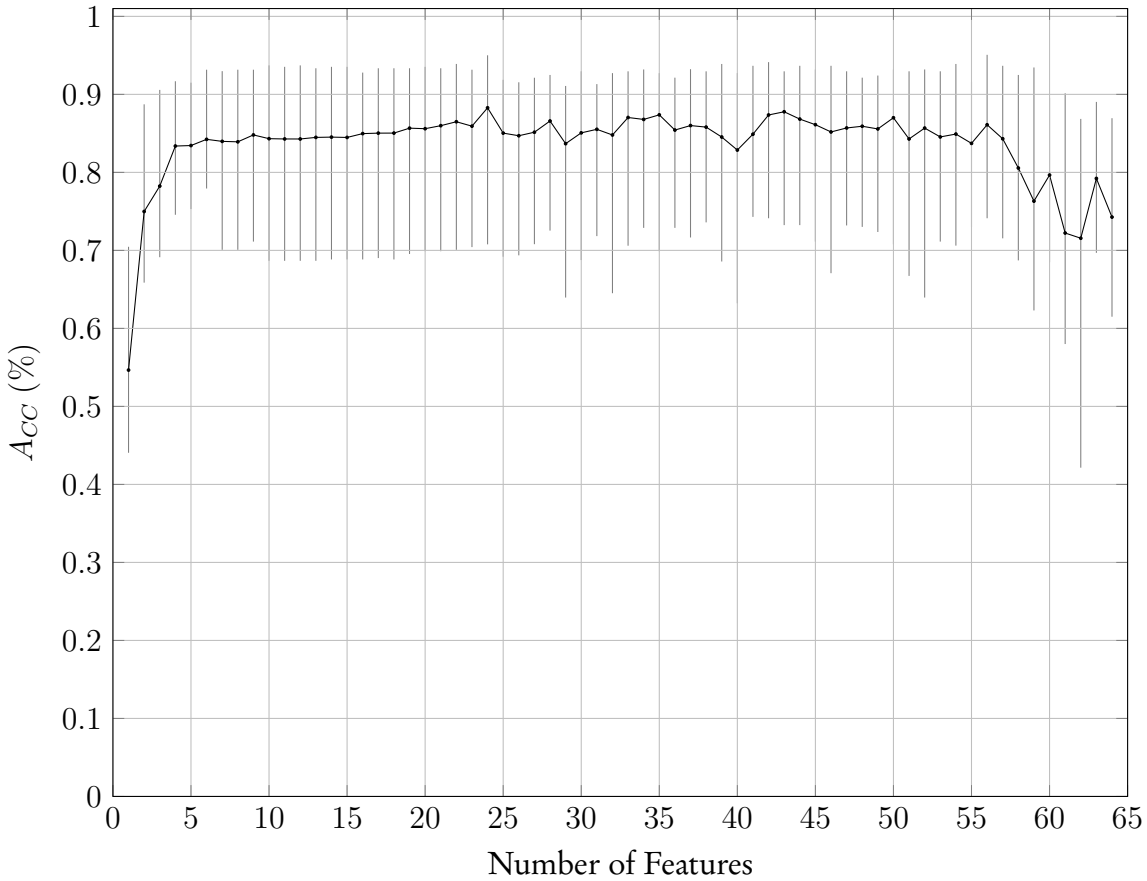


Figure 7.6: The mean cross-validated test-set accuracies achieved utilising optimal features and a linear support vector machine evaluated on the sheep dataset. The error-bars indicate the maximum and minimum test-set accuracies achieved for the specific number of features.

Stellenbosch University https://scholar.sun.ac.za

7.6 Conclusion

This chapter presented an energy-aware feature- and model selection technique for real-time embedded behaviour classification. A greedy sequential feature selection algorithm was utilised to minimise a cost function. This function incorporates a linear weighting of the energy expense of adding specific features and the change in classification error afforded by the added features. In addition, the energy expense of different classification algorithms was considered in selecting the optimal models. Optimal models were identified for both the rhinoceros and sheep datasets. A comparison of the energy and memory requirements of biotelemetry devices are presented in the next chapter which compares classical techniques to the proposed real-time embedded behaviour classification system.

Stellenbosch University https://scholar.sun.ac.za

Chapter 8

Reduced energy and memory requirements

In this chapter the energy and memory requirements of the biotelemetry tags when performing real-time embedded behaviour classification are compared to the classical technique of processing the data at a later stage. This is achieved by configuring the biotelemetry tags in four different ways and subsequently measuring the energy consumption and memory utilisation of each configuration. This chapter is based on our earlier work as presented in [3].

8.1 Device configuration

The biotelemetry tags were configured in four different manners as shown in Table 8.1. Configurations 1 and 2 involve the storage and/or transmission of the raw data, while Configurations 3 and 4 involve the storage and/or transmission of the classification result.

Table 8.1: The four hardware configurations used in power measurements.

Configuration	Description
1	Sample and store raw accelerometer data.
2	Sample and transmit raw accelerometer data.
3	Sample accelerometer data, classify behaviour and store the classification result.
4	Sample accelerometer data, classify behaviour and transmit the classification result.

Configurations 1 corresponds to the classical bio-logging case, where all raw measurements are stored on the device and later downloaded after retrieving the collar. This configuration is commonly implemented in many applications, since storing the measurements requires less energy than transmitting it, and results in longer deployment times. However, this configuration is constrained by the available onboard memory and does not provide real-time infor-

Stellenbosch University https://scholar.sun.ac.za

mation. Configuration 2 involves the transmission of raw measurements to a receiver station enabling real-time data processing and subsequent behaviour classification. This technique was only recently implemented for the first time in accordance with the growing demand for real-time information, as explained in Chapter 2. Intuitively, we can see that this approach is energy demanding as it requires the transmission of large volumes of unprocessed data to enable real-time data processing. It is suggested that embedded real-time behaviour classification can be used to reduce both the energy consumption and memory requirements of the tags, thereby increasing the operational lifetime of the tags and the immediacy of the data. For this investigation Configurations 3 and 4 are presented. In Configuration 3, raw accelerometer measurements are classified and stored onboard. This should be compared to Configuration 1 and we will later see that it significantly reduces the memory utilisation of the tags. In Configuration 4, raw accelerometer measurements are classified and subsequently transmitted to a receiver station where the behavioural updates can be analysed in real-time. This should be compared with Configuration 2 and it is subsequently proved to significantly reduce the energy consumed by the tags while providing real-time behavioural updates.

8.2 Firmware implementation

The firmware implementations for both Configurations 1 and 2 are straightforward and involves the periodic sampling of 128 accelerometer measurements whereafter these samples are either stored (Configuration 1) or transmitted to a receiver station (Configuration 2). Configurations 3 and 4 additionally include feature extraction and classification steps before storing or transmitting the result. For both Configurations 3 and 4 the optimal classification models as presented in Chapter 7 were implemented on the tags. The firmware flowdiagram for the real-time embedded behaviour classification system is shown in Figure 6.1 in Chapter 6. For the rhinoceros dataset, the optimal model is a random forest classifier with two selected features. From Chapter 7 we see that this model achieved a classification accuracy of $99.33\,\%$ among three behavioural classes. The two features are max(x), min(x). The algorithm for this classifier is shown in Algorithm 5. The algorithm indicates a number of 10 trees in the forest, comprising three identical and seven unique decision trees.

Stellenbosch University https://scholar.sun.ac.za

ALGORITHM 5: Random forest embedded classification implementation.

```
1:  procedure RF CLASSIFICATION(x)
2:      class ← -1
3:      number_classes ← 3
4:      countClass[3] ← {0}
5:      i ← 0
6:      max ← 0
7:      max_x ← x[0]
8:      min_x ← x[1]
9:      if min_x ≤ 0.1205 then
10:         countClass[0] ← countClass[0] + 3
11:     else if max_x ≤ 1.3061 then
12:         countClass[1] ← countClass[1] + 3
13:     else
14:         countClass[2] ← countClass[2] + 3
15:     end if
16:     if min_x ≤ 0.1394 then
17:         countClass[0] ← countClass[0] + 1
18:     else if max_x ≤ 1.3061 then
19:         countClass[1] ← countClass[1] + 1
20:     else
21:         countClass[2] ← countClass[2] + 1
22:     end if
23:     if min_x ≤ 0.1205 then
24:         countClass[0] ← countClass[0] + 1
25:     else if max_x ≤ 1.2517 then
26:         countClass[1] ← countClass[1] + 1
27:     else
28:         countClass[2] ← countClass[2] + 1
29:     end if
30:     if min_x ≤ 0.123 then
31:         countClass[0] ← countClass[0] + 1
32:     else if max_x ≤ 1.3061 then
33:         countClass[1] ← countClass[1] + 1
34:     else
35:         countClass[2] ← countClass[2] + 1
36:     end if
37:     if min_x ≤ 0.1229 then
38:         countClass[0] ← countClass[0] + 1
39:     else if max_x ≤ 1.2517 then
40:         countClass[1] ← countClass[1] + 1
41:     else
42:         countClass[2] ← countClass[2] + 1
43:     end if
44:     if min_x ≤ 0.1205 then
45:         countClass[0] ← countClass[0] + 1
46:     else if max_x ≤ 1.124 then
47:         countClass[1] ← countClass[1] + 1
48:     else
49:         countClass[2] ← countClass[2] + 1
50:     end if
51:     if min_x ≤ 0.123 then
52:         countClass[0] ← countClass[0] + 1
53:     else if max_x ≤ 1.1784 then
54:         countClass[1] ← countClass[1] + 1
55:     else
56:         countClass[2] ← countClass[2] + 1
57:     end if
58:     if min_x ≤ 0.123 then
59:         countClass[0] ← countClass[0] + 1
60:     else if max_x ≤ 1.2517 then
61:         countClass[1] ← countClass[1] + 1
62:     else
63:         countClass[2] ← countClass[2] + 1
64:     end if
65:     max ← countClass[0]
66:     for i ← 0; i ≤ number_classes do
67:         if countClass[i] ≥ max then
68:             class ← i
69:             max ← countClass[i]
70:         end if
71:     end for
72: return class
73: end procedure
```

Stellenbosch University https://scholar.sun.ac.za

For the sheep dataset the optimal model is a linear support vector machine with nine features. Chapter 7 recorded that this model achieved a classification accuracy of $88.40\,\%$ among five behavioural classes. The nine features are mean(y), min(y), min(z), bin_4(z), bin_3(x), skew(y), max(y), dist(y,z) and dist(x,z). The same procedure is followed to perform classification as explained in Chapter 6.2. Algorithm 6 provides the details of the firmware implementation of the classifier.

Algorithm 6: Support vector machine embedded classification implementation.

---

procedure Support Vector Machine Classification($\mathbf{x}$)
  $\mathbf{v} \leftarrow [1 \times 45]$ # floating point weight values expressed as a vector
  $\mathbf{v}_0 \leftarrow [1 \times 5]$ # floating point bias values
  $\mathbf{g} \leftarrow \{0, 0, 0\}$
  $l \leftarrow 0$
  for $i \leftarrow 0; i < 5$ **do**
    for $j \leftarrow 0; j < 9$ **do**
      $g[i] \leftarrow g[i] + \mathbf{x}[j] \cdot \mathbf{v}[l++]$
    end for
    $g[i] \leftarrow g[i] + \mathbf{v}_0[i]$
  end for
  $max \leftarrow g[0]$
  for $i \leftarrow 0; i < 5$ **do**
    if $g[i] \geq max$ **then**
      $max \leftarrow g[i]$
      $class \leftarrow i$
    end if
  end for
return $class$
end procedure

Stellenbosch University https://scholar.sun.ac.za

8.3 Power measurement

For each of the four different configurations, the power consumption was measured using an INA118 precision instrumentation amplifier circuit (Burr-Brown, Tucson, Arizona, US [92]), as shown in Figure 8.1.

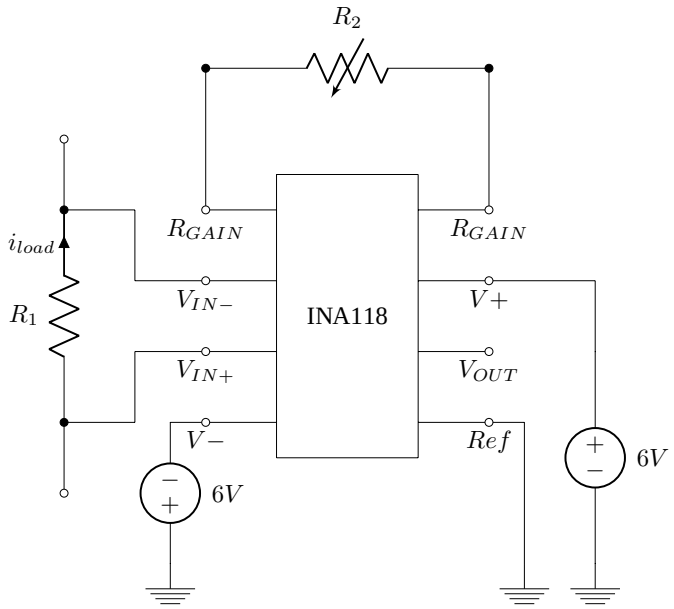


Figure 8.1: Instrumentation amplifier circuit diagram. Decoupling capacitors are not shown.

The system was configured such that the current drawn by the biotelemetry tag flows through R_1 shown in Figure 8.1. The voltage drop across R_1 is amplified using the instrumentation amplifier. Three different gains were used in order to measure the dynamic current consumption with high resolution. This was achieved by using values of $4121.0\,\Omega$, $234.25\,\Omega$ and $33.28\,\Omega$ for R_2, resulting in gains of 13.13, 214.45 and 1503.40 respectively. The output of the instrumentation amplifier V_{OUT} was recorded using an Agilent Technologies MSO-X 3014A mixed signal oscilloscope. These measurements were averaged per biotelemetry tag state, to ease the comparative analysis as presented in the remainder of this chapter.

8.4 Results

8.4.1 Configuration 1: Sample and store raw accelerometer data

At a sample rate of $20\,\mathrm{Hz}$, 128 tri-axial accelerometer measurements are captured in the space of $6.4\,\mathrm{s}$. When these 128 samples are simply stored onboard using the packet format shown in Table 8.2, the current consumption is $1.69\,\mathrm{mA}$ on average and the data occupy $1408\,\mathrm{bytes}$ of onboard memory. This results in a memory utilisation rate of $1760\,\mathrm{bits/s}$.

Stellenbosch University https://scholar.sun.ac.za

Table 8.2: Packet format for raw data (11 bytes).

Bytes	1	2	3	4	5	6	7	8	9	10	11
Description	Packet ID	Time stamp				x-axis		y-axis		z-axis	

8.4.2 Configuration 2: Sample and transmit raw accelerometer data

At a sample rate of 20 Hz, 128 tri-axial accelerometer measurements are captured in the space of 6.4 s. During this period the average current consumption is 1.69 mA, shown in Figure 8.2. The data is subsequently transmitted to a receiver station for further processing. From Figure 8.2 it is clear that, in addition to the 6.4 s required to sample and store the raw data, a further 5.22 s is required for the transmission of this raw information. During transmission the current consumption is 32.56 mA.

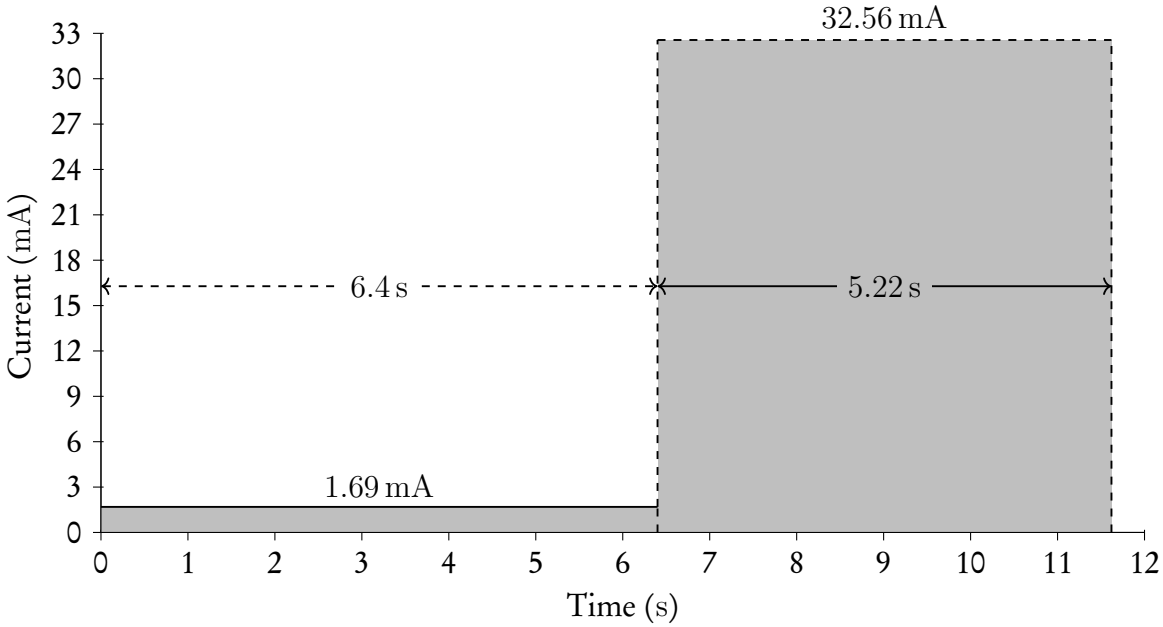


Figure 8.2: Current consumption as a function of time for Configuration 2 (sample and transmit raw data).

8.4.3 Configuration 3: Sample accelerometer data, classify behaviour and store the classification result

As before, 128 tri-axial accelerometer measurements are sampled and temporarily stored. In Configuration 3, this sequence of acceleration measurements is subsequently processed by the onboard classifier to yield a classification result, encoded as a single byte. In the case of rhinoceros, Algorithm 5 was implemented. Figure 8.3 illustrates that, apart from the 6.4 s required to acquire the raw data, the microcontroller spends 7.94 ms to load the data, extract

Stellenbosch University https://scholar.sun.ac.za

the features, classify the behaviour and store the result, while consuming 2.08 mA. A classification result is thus available every 6.41 s. For each 6.41 s cycle, a single 6 byte data packet, as described in Table 8.3, is stored. This results in a memory utilisation rate of 7.49 bits/s.

Table 8.3: Packet format for classified data (6 bytes).

Bytes	1	2	3	4	5	6
Description	Packet ID	Time stamp				Behavioural class

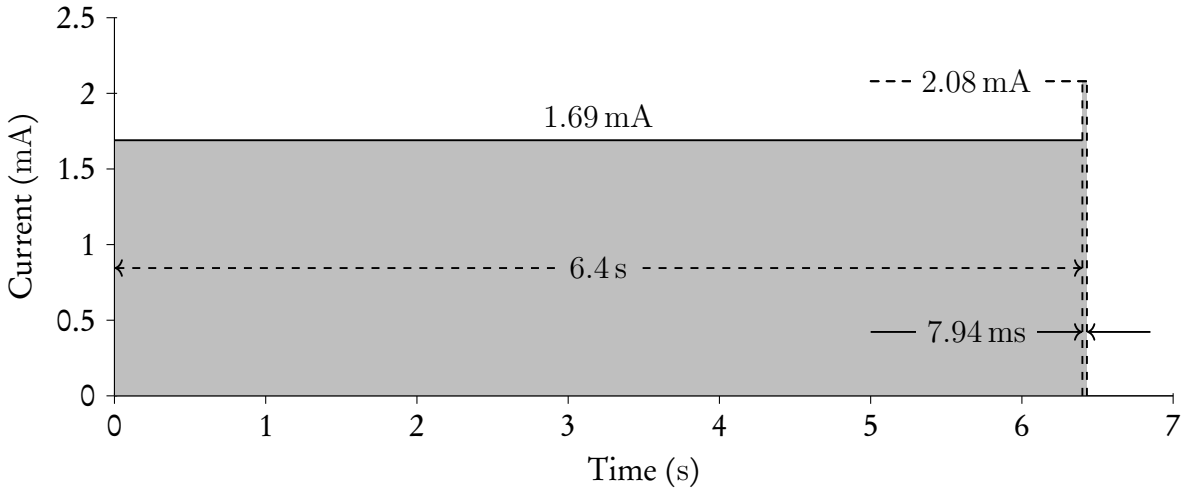


Figure 8.3: Current consumption as a function of time for Configuration 3 (sample, classify and store decision) for the rhinoceros dataset.

In the case of sheep, Algorithm 6 was implemented. From Figure 8.4 it can be seen that, apart from the 6.4 s required to acquire the raw data, the microcontroller spends 89.46 ms to load the data, extract the features, classify the behaviour and store the result, while consuming 2.08 mA. A classification result is thus available every 6.49 s. For each 6.49 s cycle, a single 6 byte data packet, as described in Table 8.3, is stored. This results in a memory utilisation rate of 7.40 bits/s.

8.4.4 Configuration 4: Sample accelerometer data, classify behaviour and transmit the classification result

In this configuration, the same procedure is followed as in Configuration 3. This results in the same power consumption and timing requirements described in Configuration 3 for both the sheep and rhinoceros classifiers. However, in addition, the classification result is subsequently transmitted. This requires an additional 55.4 ms, as shown in Figures 8.5 and 8.6 for rhinoceros and sheep, respectively. As before, the current consumption during transmission is 32.56 mA. Behavioural updates are therefore available every 6.47 s and 6.55 s for rhinoceros

Stellenbosch University https://scholar.sun.ac.za

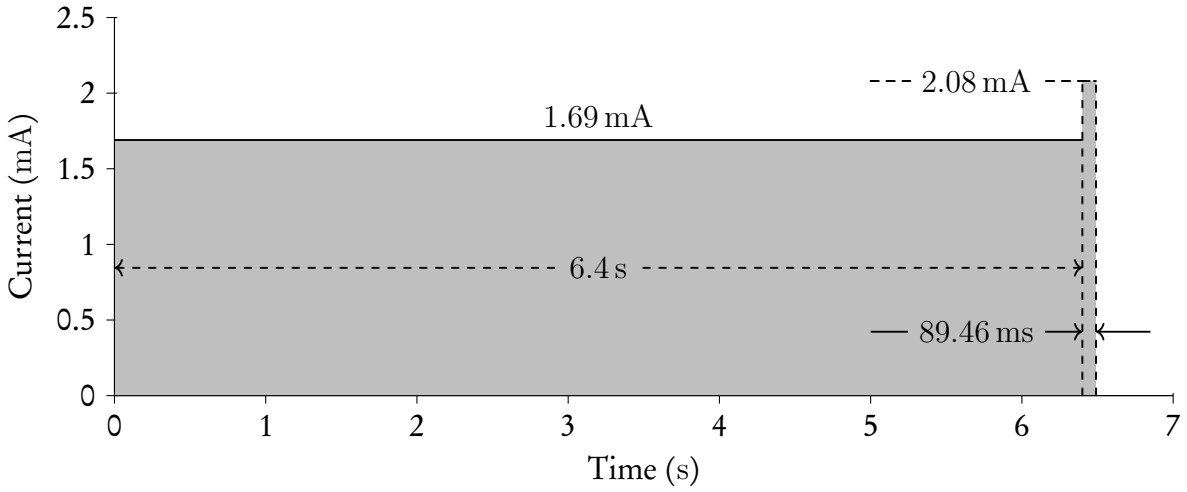


Figure 8.4: Current consumption as a function of time for Configuration 3 (sample, classify and store decision) for the sheep dataset.

and sheep respectively. Furthermore, since the transmission period of the classified data is shorter than in the case of the raw data, as per Configuration 2, the periods during which no data are recorded are shortened from 5.22 s to 63.34 ms for rhinoceros and 144.86 ms for sheep, respectively.

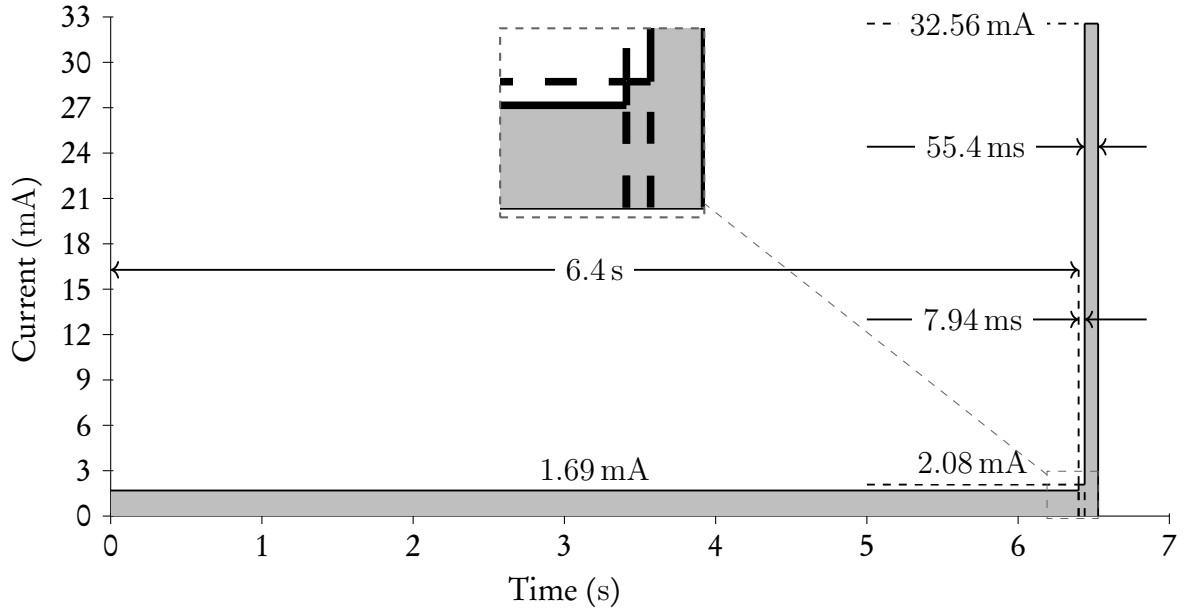


Figure 8.5: Current consumption as a function of time for Configuration 4 (sample, classify, temporarily store and transmit decision) for the rhinoceros dataset.

Stellenbosch University https://scholar.sun.ac.za

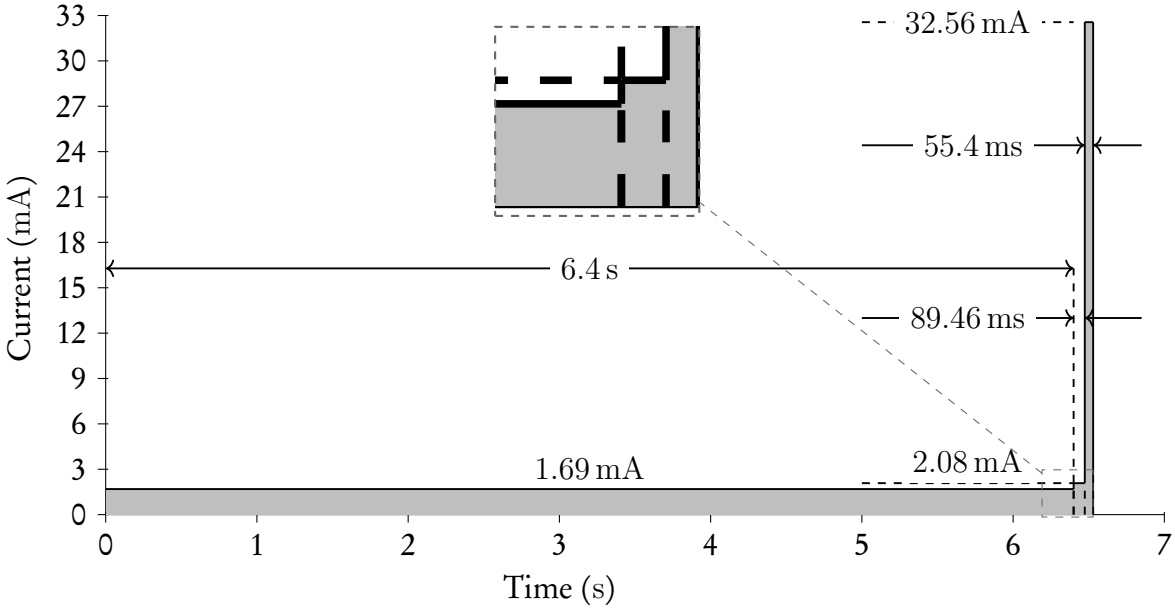


Figure 8.6: Current consumption as a function of time for Configuration 4 (sample, classify, temporarily store and transmit decision) for the sheep dataset.

8.5 Discussion

Tables 8.4 and 8.5 summarises the measurements obtained from the four experimental configurations for the rhinoceros and sheep, respectively. The tables indicate the average current consumption required to process each frame of 128 successive acceleration measurements, the energy (in Joules) required per frame, the number of bytes stored per frame, and the time required to process one frame.

Table 8.4: Summary of the per frame energy consumption and memory usage for each configuration of the rhinoceros dataset.

	Average current (mA)	**Joules per frame** (mJ)	**Memory usage per frame (bytes)**	**Time per frame (s)**
Configuration 1	1.690	35.693	1408	6.40
Configuration 2	15.557	596.568	0	11.62
Configuration 3	1.691	35.747	6	6.41
Configuration 4	1.955	41.67	0	6.46

8.5.1 Analysis of data storage options

Tables 8.4 and 8.5 show that performing behaviour classification onboard and storing the result takes approximately 10 ms longer for rhinoceros and 90 ms longer for sheep, than simply saving the raw data. Performing the classification has a very small effect on the power

Stellenbosch University https://scholar.sun.ac.za

Table 8.5: Summary of the per frame energy consumption and memory usage for each configuration of the sheep dataset.

	Average current (mA)	Joules per frame (mJ)	Memory usage per frame (bytes)	Time per frame (s)
Configuration 1	1.690	35.693	1408	6.40
Configuration 2	15.557	596.568	0	11.62
Configuration 3	1.695	36.304	6	6.49
Configuration 4	1.957	42.260	0	6.54

consumption, which increases by $0.15\,\%$ and $1.71\,\%$ for rhinoceros and sheep, respectively. However, storing the classification result rather than the raw acceleration measurements significantly reduces memory usage and thereby increasing the operational life of the device. For both rhinoceros and sheep, we calculate that $\frac{1408}{6} \approx 234$ times less memory is occupied per frame.

8.5.2 Analysis of data transmission options

Tables 8.4 and 8.5 clearly illustrates that performing behaviour classification onboard before transmission requires $\frac{597}{42} \approx 14$ times less power per frame than transmission of the raw measurements. This is especially advantageous in applications where the transmission of data cannot be avoided when, for example, real-time updates are essential. To illustrate the benefits of onboard behaviour classification, we recall the study presented in [31], where raw accelerometer measurements were sampled and transmitted at $50\,\mathrm{Hz}$ to a receiver station, and subsequently classified. The battery lifetime of this configuration was reported to be approximately two days. The presented system achieves a similar battery lifetime when raw accelerometer measurements are sampled at $20\,\mathrm{Hz}$ and transmitted to a receiver station. Specifically, from Table 8.4, the battery lifetime of Configuration 2 can be calculated as $\frac{1800mAh}{15.557mA*24h} = 4.8$ days. However, onboard behaviour classification extends the battery life to $\frac{1800mAh}{1.956mA*24h} = 38.34$ days with an $\frac{15.56\,\mathrm{s}}{6.5\,\mathrm{s}} \approx 2.4$ times higher update interval. The reduced energy consumption afforded by onboard classification will, therefore, substantially increase the lifetime of the biotelemetry tag.

8.5.3 Reduced update frequency

The data storage and transmission techniques (described in Configurations 1 to 4) continuously store and/or transmit raw or classified information. For example, Configuration 4 will continuously transmit behavioural updates every $6.54\,\mathrm{s}$. However, in many applications behavioural updates are required less frequently. In this case, the lifespan of the biotelemetry

Stellenbosch University https://scholar.sun.ac.za

tags can be further increased by utilising the tag's low-power sleep mode (in this case consuming $6.065\,\mu A$) between updates. By entering the low-power mode between updates, the user can balance data resolution and power consumption for a specific application. The average current consumption i_{avg} for a specific update interval is given by Equation 8.1, where, t_{tot} is the desired update interval, t_{active} is the period required to process each frame provided by Tables 8.4 and 8.5, i_{active} is the average current consumed during t_{active}, also contained in the tables, and i_{sleep} is the sleep mode current consumption of the device.

$$i_{avg} = \frac{t_{active}}{t_{tot}} * i_{active} + \frac{t_{tot} - t_{active}}{t_{tot}} * i_{sleep} \tag{8.1}$$

For example, when live behavioural update are transmitted every five minutes using Configuration 4 for sheep, the tags consume an average current of $48.59\,\mu A$:

$$i_{avg} = \frac{6.54\,s}{300\,s} * 1.9566\,mA + \frac{300\,s - 6.54\,s}{300\,s} * 6.065\,\mu A = 48.59\,\mu A$$

Similarly, transmitting raw information every five minutes using Configuration 2 results in an average current consumption of $608.40\,\mu A$:

$$i_{avg} = \frac{11.62\,s}{300\,s} * 15.557\,mA + \frac{300\,s - 11.62\,s}{300\,s} * 6.065\,\mu A = 608.40\,\mu A$$

Using these values the battery life can be estimated as approximately four months when raw data is transmitted, as opposed to approximately four years when onboard behaviour classification is performed.

8.6 Informed power management

Embedded behaviour classification provides the current behavioural state of the animal in real-time, which is advantageous since these insights can be used to develop new informed power management techniques. The latter will enable the effective management of power hungry components such as GPS modules and radio frequency transmitters. Depending on the application, various data acquisition strategies can be followed to obtain optimal data resolution and deployment times. For example, the system can be configured to transmit behavioural

Stellenbosch University https://scholar.sun.ac.za

updates only when the behavioural state changes. Alternatively, it can alter the data transmission rate between different behavioural states. The latter approach will allow longer update intervals when the animal is immobile and shorter update intervals when the animal is moving. It is also possible to only broadcast information about a particular behaviour. This has the potential to enable long term studies of specific behaviours such as resting or migratory patterns.

8.7 Conclusion

This chapter has considered the effect on power and memory requirements of performing statistical classification of tri-axial accelerometer measurements into behavioural classes directly on a biotelemetry tag, as opposed to processing at a receiver station after wireless transmission of the raw measurements. Statistical classifiers for both rhinoceros and sheep behaviour implemented on low-power biotelemetry tags, achieved a 14-fold reduction in energy consumption when compared with a system in which the raw acceleration data was transmitted. Furthermore, when data is stored, real-time embedded classification led to a 234-fold reduction in the memory utilised. It can be concluded that real-time embedded behaviour classification is highly advantageous from a power consumption and memory usage viewpoint. This applies especially to applications where the transmission of data cannot be avoided when, for example, real-time updates are essential.

Stellenbosch University https://scholar.sun.ac.za

Chapter 9

Conclusion

This dissertation presented a complete design and optimisation technique for a real-time embedded animal behaviour classification system. The proposed technique offers significant advantages over conventional techniques by increasing data immediacy while simultaneously reducing both the energy consumption and memory requirements of the system. This enables long-term real-time behavioural studies, which are advantageous in fields such as nature conservation, behavioural ecology and precision farming.

A novel embedded automatic behaviour classification system which captured and automatically classified three-dimensional accelerometer data in real-time was first presented. All computations occurred on specially designed biotelemetry tags while still attached to the animal. This allows the probable behaviour to be transmitted continuously, thereby providing an enhanced level of detailed real-time information. Applying the principle to rhinoceros behaviour monitoring, a linear support vector machine with 11 features achieves an accuracy of $99.61\,\%$ among three behavioural classes (standing, walking and lying down). In the case of sheep, logistic regression with 34 features achieves a classification accuracy of $89.59\,\%$ among five behavioural classes (standing, walking, grazing, running and lying down). The animal behaviour was estimated approximately every $6.5\,\mathrm{s}$ for both rhinoceros and sheep and transmitted to a receiver station.

Secondly, a novel energy-aware feature and model selection technique was presented. A greedy sequential feature selection algorithm was utilised to minimise a cost function. This function incorporates a linear weighting of the energy expense of adding specific features and the change in classification error afforded by the added features. The energy expense of different classification algorithms was incorporated to select the optimal model configuration. The technique, therefore, favours classifiers and features which in combination are less energy expensive to compute during runtime. For the rhinoceros dataset, a random forest

Stellenbosch University https://scholar.sun.ac.za

classifier with two features is selected as optimal and achieved an overall classification accuracy of $99.33\,\%$. Extracting the features and performing classification consumes 363 times less energy, while only sacrificing $0.28\,\%$ in accuracy when compared to the $99.61\,\%$ achieved with the unconstrained system. For the sheep dataset, a linear support vector machine with 9 features achieves an $88.40\,\%$ classification accuracy. Extracting the features and performing classification consumes 6.8 times less energy, at a cost of $1.19\,\%$ in accuracy against the $89.59\,\%$ achieved with the unconstrained system.

Finally, the project considered the reduced energy requirements and memory usage benefits of the real-time embedded behaviour classification system as a whole. Experiments using the biotelemetry tags demonstrated a 14-fold reduction in energy consumption and a 234-fold reduction in memory usage when classification was performed on the tag, relative to raw data transmission with subsequent processing. It is concluded that real-time behavioural updates can be achieved by means of embedded behaviour classification. The technique furthermore significantly reduces the total energy consumption and memory requirements of the device. This enables long-term behavioural studies in applications such as the conservation of critically endangered species, such as rhinoceros. The technique can also be applied in precision farming applications as well as general embedded machine learning applications implemented on mobile platforms such as smart phones or smart watches or in sensors within the internet of things.

9.1 Project outcomes and contributions

The primary contributions of the work as set out in this dissertation are:

- A novel real-time embedded behaviour classification system for both rhinoceros and sheep as previously presented in [2].
- A novel energy-aware feature and classification model selection technique that utilises a greedy sequential feature selection algorithm to minimise a cost function. This function incorporates a linear weighting of the energy expense of adding specific features and the change in classification error afforded by the added features. In addition, the energy expense of different classification algorithms was considered in selecting the optimal models, which is currently neglected in the literature. The presented technique, therefore, favours classifiers and features which in combination are less energy expensive to compute at runtime.
- The significant reduction of energy and memory requirements of biotelemetry devices by means of real-time embedded behaviour classification, as previously set out in [3].

Stellenbosch University https://scholar.sun.ac.za

This is generally a very significant obstacle in most electronic devices utilised for on-animal behavioural studies.

The secondary contributions of the work are:

- A grouped nested and repeated cross-validation algorithm developed to perform model optimisation, validation and evaluation. Although cross-validation is common practice, details regarding accurate implementation are hard to find. The complete algorithm presented in this dissertation ensures model reproducibility for machine learning applications in general and is not limited to animal behaviour classification.
- An automatic behaviour classification systems for both rhinoceros and sheep which achieve high classification performance among common behaviours such as standing, grazing, walking, running and lying down. Furthermore, datasets of rhinoceros and sheep acceleration measurements are compiled in order to perform experimental evaluation.

9.2 Future research

Possible future research and expansions on this work include:

- A new hardware design revision of the biotelemetry tags. With the increased industrial demand for low-power wireless sensor applications, rapid advances have been made in terms of low-power hardware solutions. Researchers must continue to redesign biotelemetry tags and include the latest low-power integrated circuits. Furthermore, the size of the biotelemetry tags can be reduced to ensure comfort when attached to animals.
- The design of the collar used to enclose and attach the biotelemetry tags to animals. The collar design should provide maximum comfort for the animal while ensuring durability in long-term deployments. Furthermore, an automatic-release mechanism should be employed to detach the collar from the animal when the battery is almost depleted.
- The integration of energy harvesting techniques that utilise potential energy sources such as kinetic energy or solar energy to further extend the lifetime of these devices.
- The design, optimisation and evaluation of a low-power energy-aware wireless communication routing protocol that ensures robust data delivery while minimising the energy requirements of the tags.
- The design, optimisation and evaluation of repeater stations for receiving and relaying information to a control room.

Stellenbosch University https://scholar.sun.ac.za

Bibliography

[1] Poaching statistics. https://www.savetherhino.org/. Save the Rhino International. Accessed: 2018-08-03.

[2] S. P. le Roux, J. Marias, R. Wolhuter, and T. Niesler, "Animal-borne behaviour classification for sheep (dohne merino) and rhinoceros (ceratotherium simum and diceros bicornis)," *Animal Biotelemetry*, vol. 5, no. 25, Nov 2017. [Online]. Available: https://doi.org/10.1186/s40317-017-0140-0

[3] S. P. le Roux, R. Wolhuter, N. Stevens, and T. Niesler, "Reduced energy and memory requirements by on-board behavior classification for animal-borne sensor applications," *IEEE Sensors Journal*, vol. 18, no. 10, pp. 4261–4268, May 2018.

[4] H. Edelhoff, J. Signer, and N. Balkenhol, "Path segmentation for beginners: an overview of current methods for detecting changes in animal movement patterns," *Movement Ecology*, vol. 4, no. 1, p. 21, Sep 2016. [Online]. Available: https://doi.org/10.1186/s40462-016-0086-5

[5] R. Kays, M. C. Crofoot, W. Jetz, and M. Wikelski, "Terrestrial animal tracking as an eye on life and planet," *Science*, vol. 348, no. 6240, 2015.

[6] O. R. Bidder, J. S. Walker, M. W. Jones, M. D. Holton, P. Urge, D. M. Scantlebury, N. J. Marks, E. A. Magowan, I. E. Maguire, and R. P. Wilson, "Step by step: reconstruction of terrestrial animal movement paths by dead-reckoning," *Movement Ecology*, vol. 3, no. 1, p. 23, Sep 2015. [Online]. Available: https://doi.org/10.1186/s40462-015-0055-4

[7] A. Kumar and G. P. Hancke, "A zigbee-based animal health monitoring system," *IEEE Sensors Journal*, vol. 15, no. 1, pp. 610–617, Jan 2015.

[8] H. J. Williams, M. D. Holton, E. L. C. Shepard, N. Largey, B. Norman, P. G. Ryan, O. Duriez, M. Scantlebury, F. Quintana, E. A. Magowan, N. J. Marks, A. N. Alagaili, N. C. Bennett, and R. P. Wilson, "Identification of animal movement patterns using tri-axial magnetometry," *Movement Ecology*, vol. 5, no. 1, p. 6, Mar 2017. [Online]. Available: https://doi.org/10.1186/s40462-017-0097-x

Stellenbosch University https://scholar.sun.ac.za

[9] H. Fourati, N. Manamanni, L. Afilal, and Y. Handrich, "Posture and body acceleration tracking by inertial and magnetic sensing: Application in behavioral analysis of free-ranging animals," *Biomedical Signal Processing and Control*, vol. 6, no. 1, pp. 94 – 104, 2011, biomedical Signal Processing (Extended selected papers from the 7th IFAC Symposium on Modelling and Control in Biomedical Systems(MCBMS'09)).

[10] J. Still, "Use of animal products in traditional chinese medicine: environmental impact and health hazards," *Complementary Therapies in Medicine*, vol. 11, no. 2, pp. 118 – 122, 2003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0965229903000554

[11] E. B. Martin, "Rhinos and daggers: a major conservation problem," *Oryx*, vol. 19, no. 4, pp. 198–201, 1985.

[12] D. Biggs, F. Courchamp, R. Martin, and H. P. Possingham, "Legal trade of africa's rhino horns," *Science*, vol. 339, no. 6123, pp. 1038–1039, 2013.

[13] A. K. Emslie RH, "A conservation assessment of diceros bicornis." 2016, in Child MF, Roxburgh L, Do Linh San E, Raimondo D, Davies-Mostert HT, editors. The Red List of Mammals of South Africa, Swaziland and Lesotho. South African National Biodiversity Institute and Endangered Wildlife Trust, South Africa.

[14] A. K. Emslie R, "A conservation assessment of ceratotherium simum simum." 2016, in Child MF, Roxburgh L, Do Linh San E, Raimondo D, Davies-Mostert HT, editors. The Red List of Mammals of South Africa, Swaziland and Lesotho. South African National Biodiversity Institute and Endangered Wildlife Trust, South Africa.

[15] B. Büscher and M. Ramutsindela, "Green violence: Rhino poaching and the war to save southern africa's peace parks," *African Affairs*, vol. 115, no. 458, pp. 1–22, 2016.

[16] H. Theron, "Rhino genetics should be conserved," *Stockfarm*, vol. 7, no. 4, pp. 38 – 41, 2017.

[17] W. Clack, "Livestock theft: Lies, damn lies and statistics," Khaya Ibhubesi Conference Centre Parys, Aug. 2016, national Red Meat Producers Congress August 2016.

[18] http://www.farmersweekly.co.za, "Saps reports stock theft increase for latest reported period," http://www.farmersweekly.co.za/agri-news/south-africa/saps-reports-stock-theft-increase-for-latest-reported-period/, 2017, accessed: 2017-07-18.

Stellenbosch University https://scholar.sun.ac.za

[19] T. M. Caro, "Behavioural indicators of exploitation," *Ethology Ecology & Evolution*, vol. 17, no. 2, pp. 189–194, 2005. [Online]. Available: https://doi.org/10.1080/08927014.2005.9522608

[20] G. G. Miguel-Pacheco, H. J. Thomas, J. Kaler, J. Craigon, and J. N. Huxley, "Effects of lameness treatment for claw horn lesions on lying behaviour in dairy cows," *Applied Animal Behaviour Science*, vol. 179, pp. 11 – 16, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0168159116300909

[21] M. Pastell, J. Tiusanen, M. Hakojärvi, and L. Hänninen, "A wireless accelerometer system with wavelet analysis for assessing lameness in cattle," *Biosystems Engineering*, vol. 104, no. 4, pp. 545 – 551, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1537511009002748

[22] J. O. Chelotti, S. R. Vanrell, J. R. Galli, L. L. Giovanini, and H. L. Rufiner, "A pattern recognition approach for detecting and classifying jaw movements in grazing cattle," *Computers and Electronics in Agriculture*, vol. 145, pp. 83 – 91, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0168169917302752

[23] D. D. Brown, R. Kays, M. Wikelski, R. Wilson, and A. P. Klimley, "Observing the unwatchable through acceleration logging of animal behaviour," *Animal Biotelemetry*, vol. 1, no. 1, p. 20, 2013.

[24] E. L. Shepard, R. P. Wilson, F. Quintana, A. G. Laich, N. Liebsch, D. A. Albareda, L. G. Halsey, A. Gleiss, D. T. Morgan, A. E. Myers *et al.*, "Identification of animal movement patterns using tri-axial accelerometry," *Endangered Species Research*, vol. 10, pp. 47–60, 2008.

[25] D. W. McClune, N. J. Marks, R. P. Wilson, J. D. Houghton, I. W. Montgomery, N. E. McGowan, E. Gormley, and M. Scantlebury, "Tri-axial accelerometers quantify behaviour in the Eurasian badger (Meles meles): towards an automated interpretation of field data," *Animal Biotelemetry*, vol. 2, no. 1, p. 5, 2014.

[26] R. Nathan, O. Spiegel, S. Fortmann-Roe, R. Harel, M. Wikelski, and W. M. Getz, "Using tri-axial acceleration data to identify behavioural modes of free-ranging animals: general concepts and tools illustrated for griffon vultures," *Journal of Experimental Biology*, vol. 215, no. 6, pp. 986–996, 2012.

[27] A. Rahman, D. Smith, J. Hills, G. Bishop-Hurley, D. Henry, and R. Rawnsley, "A comparison of autoencoder and statistical features for cattle behaviour classification," in *2016 International Joint Conference on Neural Networks (IJCNN)*, July 2016, pp. 2954–2960.

Stellenbosch University https://scholar.sun.ac.za

[28] L. R. Brewster, J. J. Dale, T. L. Guttridge, S. H. Gruber, A. C. Hansell, M. Elliott, I. G. Cowx, N. M. Whitney, and A. C. Gleiss, "Development and application of a machine learning algorithm for classification of elasmobranch behaviour from accelerometry data," *Marine Biology*, vol. 165, no. 4, p. 62, Mar 2018. [Online]. Available: https://doi.org/10.1007/s00227-018-3318-y

[29] K. T. Kasfi, A. Hellicar, and A. Rahman, "Convolutional neural network for time series cattle behaviour classification," in *Proceedings of the Workshop on Time Series Analytics and Applications*, ser. TSAA '16. New York, NY, USA: ACM, 2016, pp. 8–12. [Online]. Available: http://doi.acm.org/10.1145/3014340.3014342

[30] D. W. McClune, N. J. Marks, R. J. Delahay, W. I. Montgomery, and D. M. Scantlebury, "Behaviour-time budget and functional habitat use of a free-ranging european badger(meles meles)," *Animal Biotelemetry*, vol. 3, no. 1, p. 7, 2015. [Online]. Available: http://dx.doi.org/10.1186/s40317-015-0025-z

[31] J. A. Vázquez Diosdado, Z. E. Barker, H. R. Hodges, J. R. Amory, D. P. Croft, N. J. Bell, and E. A. Codling, "Classification of behaviour in housed dairy cows using an accelerometer-based activity monitoring system," *Animal Biotelemetry*, vol. 3, no. 1, p. 15, 2015. [Online]. Available: http://dx.doi.org/10.1186/s40317-015-0045-8

[32] B. Robert, B. White, D. Renter, and R. Larson, "Evaluation of three-dimensional accelerometers to monitor and classify behaviour patterns in cattle," *Computers and Electronics in Agriculture*, vol. 67, no. 1, pp. 80–84, 2009.

[33] J. Shamoun-Baranes, R. Bom, E. E. van Loon, B. J. Ens, K. Oosterbeek, and W. Bouten, "From sensor data to animal behaviour: an oystercatcher example," *PLoS one*, vol. 7, no. 5, p. e37997, 2012.

[34] P. C. P. Khanh, N. D. Chinh, T. T. Cham, P. T. Vui, and T. D. Tan, "Classification of cow behavior using 3-dof accelerometer and decision tree algorithm," in *2016 International Conference on Biomedical Engineering (BME-HUST)*, Oct 2016, pp. 45–50.

[35] P. C. P. Khanh, T. T. Long, N. D. Chinh, and T. Duc-Tan, "Performance evaluation of a multi-stage classification for cow behavior," in *2018 2nd International Conference on Recent Advances in Signal Processing, Telecommunications Computing (SigTelCom)*, Jan 2018, pp. 121–125.

[36] F. Alvarenga, I. Borges, L. Palkovič, J. Rodina, V. Oddy, and R. Dobos, "Using a three-axis accelerometer to identify and classify sheep behaviour at pasture," *Applied Animal Behaviour Science*, vol. 181, pp. 91 – 99, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0168159116301733

Stellenbosch University https://scholar.sun.ac.za

[37] J. Soltis, R. P. Wilson, I. Douglas-Hamilton, F. Vollrath, L. E. King, and A. Savage, "Accelerometers in collars identify behavioural states in captive african elephants Loxodonta africana," *Endangered Species Research*, vol. 18, no. 3, pp. 255–263, 2012.

[38] S. Grünewälder, F. Broekhuis, D. W. Macdonald, A. M. Wilson, J. W. McNutt, J. Shawe-Taylor, and S. Hailes, "Movement activity based classification of animal behaviour with an application to data from cheetah (Acinonyx jubatus)," *PLoS One*, vol. 7, no. 11, p. e49120, 2012.

[39] T. T. Hammond, D. Springthorpe, R. E. Walsh, and T. Berg-Kirkpatrick, "Using accelerometers to remotely and automatically characterize behavior in small animals," *Journal of Experimental Biology*, vol. 219, no. 11, pp. 1618–1624, 2016. [Online]. Available: http://jeb.biologists.org/content/219/11/1618

[40] M. S. Painter, J. A. Blanco, E. P. Malkemper, C. Anderson, D. C. Sweeney, C. W. Hewgley, J. Červený, V. Hart, V. Topinka, E. Belotti, H. Burda, and J. B. Phillips, "Use of bio-loggers to characterize red fox behavior with implications for studies of magnetic alignment responses in free-roaming animals," *Animal Biotelemetry*, vol. 4, no. 1, p. 20, 2016. [Online]. Available: http://dx.doi.org/10.1186/s40317-016-0113-8

[41] M. Sur, T. Suffredini, S. M. Wessells, P. H. Bloom, M. Lanzone, S. Blackshire, S. Sridhar, and T. Katzner, "Improved supervised classification of accelerometry data to distinguish behaviors of soaring birds," *PLOS ONE*, vol. 12, no. 4, pp. 1–19, 04 2017. [Online]. Available: https://doi.org/10.1371/journal.pone.0174785

[42] S. Benaissa, F. A. Tuyttens, D. Plets, T. de Pessemier, J. Trogh, E. Tanghe, L. Martens, L. Vandaele, A. V. Nuffel, W. Joseph, and B. Sonck, "On the use of on-cow accelerometers for the classification of behaviours in dairy barns," *Research in Veterinary Science*, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S003452881730423X

[43] M. A. Ladds, A. P. Thompson, J.-P. Kadar, D. J Slip, D. P Hocking, and R. G Harcourt, "Super machine learning: improving accuracy and reducing variance of behaviour classification from accelerometry," *Animal Biotelemetry*, vol. 5, no. 1, p. 8, Mar 2017. [Online]. Available: https://doi.org/10.1186/s40317-017-0123-1

[44] M. A. Ladds, A. P. Thompson, D. J. Slip, D. P. Hocking, and R. G. Harcourt, "Seeing it all: Evaluating supervised machine learning methods for the classification of diverse otariid behaviours," *PLOS ONE*, vol. 11, no. 12, pp. 1–17, 12 2016. [Online]. Available: https://doi.org/10.1371/journal.pone.0166898

Stellenbosch University https://scholar.sun.ac.za

[45] L. Guo, M. Welch, R. Dobos, P. Kwan, and W. Wang, "Comparison of grazing behaviour of sheep on pasture with different sward surface heights using an inertial measurement unit sensor," *Computers and Electronics in Agriculture*, vol. 150, pp. 394 – 401, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0168169917311936

[46] 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*, vol. 119, no. 3, pp. 158–170, 2009.

[47] D. Smith, A. Rahman, G. J. Bishop-Hurley, J. Hills, S. Shahriar, D. Henry, and R. Rawnsley, "Behavior classification of cows fitted with motion collars: Decomposing multi-class classification into a set of binary problems," *Computers and Electronics in Agriculture*, vol. 131, pp. 40 – 50, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0168169916303180

[48] N. Watanabe, S. Sakanoue, K. Kawamura, and T. Kozakai, "Development of an automatic classification system for eating, ruminating and resting behaviour of cattle using an accelerometer," *Grassland Science*, vol. 54, no. 4, pp. 231–237, 2008.

[49] G. Fehlmann, M. J. O'Riain, P. W. Hopkins, J. O'Sullivan, M. D. Holton, E. L. C. Shepard, and A. J. King, "Identification of behaviours from accelerometer data in a wild social primate," *Animal Biotelemetry*, vol. 5, no. 1, p. 6, 2017. [Online]. Available: http://dx.doi.org/10.1186/s40317-017-0121-3

[50] A. Rahman, D. Smith, B. Little, A. Ingham, P. Greenwood, and G. Bishop-Hurley, "Cattle behaviour classification from collar, halter, and ear tag sensors," *Information Processing in Agriculture*, vol. 5, no. 1, pp. 124 – 133, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2214317317301099

[51] L. Jeantet, F. Dell'Amico, M. A. Forin-Wiart, M. Coutant, M. Bonola, D. Etienne, J. Gresser, S. Regis, N. Lecerf, F. Lefebvre, B. de Thoisy, Y. Le Maho, M. Brucker, N. Châtelain, R. Laesser, F. Crenner, Y. Handrich, R. Wilson, and D. Chevallier, "Combined use of two supervised learning algorithms to model sea turtle behaviours from tri-axial acceleration data," *Journal of Experimental Biology*, 2018. [Online]. Available: http://jeb.biologists.org/content/early/2018/04/13/jeb.177378

[52] E. Walton, C. Casey, J. Mitsch, J. A. Vázquez-Diosdado, J. Yan, T. Dottorini, K. A. Ellis, A. Winterlich, and J. Kaler, "Evaluation of sampling frequency, window size and sensor position for classification of sheep behaviour," *Royal*

Stellenbosch University https://scholar.sun.ac.za

Society open science, vol. 5, no. 2, p. 171442, February 2018. [Online]. Available: http://europepmc.org/articles/PMC5830751

[53] L. Lush, S. Ellwood, A. Markham, A. I. Ward, and P. Wheeler, "Use of tri-axial accelerometers to assess terrestrial mammal behaviour in the wild," *Journal of Zoology*, vol. 298, no. 4, pp. 257–265, 2015. [Online]. Available: https://zslpublications.onlinelibrary.wiley.com/doi/abs/10.1111/jzo.12308

[54] A. M. Pagano, K. D. Rode, A. Cutting, M. Owen, S. Jensen, J. Ware, C. Robbins, G. M. Durner, T. C. Atwood, M. Obbard, K. Middel, G. Thiemann, and T. Williams, "Using tri-axial accelerometers to identify wild polar bear behaviors," *Endangered Species Research*, vol. 32, pp. 19–33, 2017.

[55] L. Lush, R. P. Wilson, M. D. Holton, P. Hopkins, K. A. Marsden, D. R. Chadwick, and A. J. King, "Classification of sheep urination events using accelerometers to aid improved measurements of livestock contributions to nitrous oxide emissions," *Computers and Electronics in Agriculture*, vol. 150, pp. 170 – 177, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0168169917313017

[56] P. Martiskainen, M. Järvinen, J.-P. Skön, J. Tiirikainen, M. Kolehmainen, and J. Mononen, "Cow behaviour pattern recognition using a three-dimensional accelerometer and support vector machines," *Applied Animal Behaviour Science*, vol. 119, no. 1, pp. 32–38, 2009.

[57] L. R. Williams, G. J. Bishop-Hurley, A. E. Anderson, and D. L. Swain, "Application of accelerometers to record drinking behaviour of beef cattle," *Animal Production Science*, 2017. [Online]. Available: https://doi.org/10.1071/AN17052

[58] H. Kour, K. P. Patison, N. J. Corbet, and D. L. Swain, "Validation of accelerometer use to measure suckling behaviour in northern australian beef calves," *Applied Animal Behaviour Science*, vol. 202, pp. 1 – 6, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0168159118300327

[59] S. Reiter, G. Sattlecker, L. Lidauer, F. Kickinger, M. Öhlschuster, W. Auer, V. Schweinzer, D. Klein-Jöbstl, M. Drillich, and M. Iwersen, "Evaluation of an ear-tag-based accelerometer for monitoring rumination in dairy cows," *Journal of Dairy Science*, vol. 101, no. 4, pp. 3398 – 3411, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0022030218300419

[60] 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

Stellenbosch University https://scholar.sun.ac.za

monitoring of eight behavioural states in dogs," *PLOS ONE*, vol. 12, no. 11, pp. 1–13, 11 2017. [Online]. Available: https://doi.org/10.1371/journal.pone.0188481

[61] V. Giovanetti, M. Decandia, G. Molle, M. Acciaro, M. Mameli, A. Cabiddu, R. Cossu, M. Serra, C. Manca, S. Rassu, and C. Dimauro, "Automatic classification system for grazing, ruminating and resting behaviour of dairy sheep using a tri-axial accelerometer," *Livestock Science*, vol. 196, pp. 42 – 48, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1871141316302906

[62] L. A. González, G. J. Bishop-Hurley, R. N. Handcock, and C. Crossman, "Behavioral classification of data from collars containing motion sensors in grazing cattle," *Computers and Electronics in Agriculture*, vol. 110, pp. 91 – 102, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0168169914002798

[63] E. S. Fogarty, D. L. Swain, G. Cronin, and M. Trotter, "Autonomous on-animal sensors in sheep research: A systematic review," *Computers and Electronics in Agriculture*, vol. 150, pp. 245 – 256, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0168169918300450

[64] J. Wall, G. Wittemyer, B. Klinkenberg, and I. Douglas-Hamilton, "Novel opportunities for wildlife conservation and research with real-time monitoring," *Ecological Applications*, vol. 24, no. 4, pp. 593–601, 2014.

[65] Ã. Cangar, T. Leroy, M. Guarino, E. Vranken, R. Fallon, J. Lenehan, J. Mee, and D. Berckmans, "Automatic real-time monitoring of locomotion and posture behaviour of pregnant cows prior to calving using online image analysis," *Computers and Electronics in Agriculture*, vol. 64, no. 1, pp. 53 – 60, 2008, smart Sensors in precision livestock farming. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0168169908001373

[66] F. Cagnacci, L. Boitani, R. A. Powell, and M. S. Boyce, "Animal ecology meets gps-based radiotelemetry: a perfect storm of opportunities and challenges," *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, vol. 365, no. 1550, pp. 2157–2162, 2010. [Online]. Available: http://rstb.royalsocietypublishing.org/content/365/1550/2157

[67] D. Curone, G. M. Bertolotti, A. Cristiani, E. L. Secco, and G. Magenes, "A real-time and self-calibrating algorithm based on triaxial accelerometer signals for the detection of human posture and activity," *IEEE Transactions on Information Technology in Biomedicine*, vol. 14, no. 4, pp. 1098–1105, July 2010.

Stellenbosch University https://scholar.sun.ac.za

[68] A. Purwar, D. U. Jeong, and W. Y. Chung, "Activity monitoring from real-time triaxial accelerometer data using sensor network," in *2007 International Conference on Control, Automation and Systems*, Oct 2007, pp. 2402–2406.

[69] S. Saponas, J. Lester, J. E. Froehlich, J. A. Fogarty, and J. A. Landay, "ilearn on the iphone: Real-time human activity classification on commodity mobile phones," University of Washington, Tech. Rep., August 2002.

[70] D. M. Karantonis, M. R. Narayanan, M. Mathie, N. H. Lovell, and B. G. Celler, "Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring," *IEEE Transactions on Information Technology in Biomedicine*, vol. 10, no. 1, pp. 156–167, Jan 2006.

[71] J. He, H. Li, and J. Tan, "Real-time daily activity classification with wireless sensor networks using hidden markov model," in *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Aug 2007, pp. 3192–3195.

[72] E. M. Tapia, S. S. Intille, W. Haskell, K. Larson, J. Wright, A. King, and R. Friedman, "Real-time recognition of physical activities and their intensities using wireless accelerometers and a heart rate monitor," in *2007 11th IEEE International Symposium on Wearable Computers*, Oct 2007, pp. 37–40.

[73] P. Kumari, L. Mathew, and P. Syal, "Increasing trend of wearables and multimodal interface for human activity monitoring: A review," *Biosensors and Bioelectronics*, vol. 90, pp. 298 – 307, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0956566316312295

[74] L. Bracciale, A. Catini, G. Gentile, and P. Loreti, *Delay Tolerant Wireless Sensor Network for Animal Monitoring: The Pink Iguana Case*, 2018, pp. 18–26.

[75] H. Ghasemzadeh, N. Amini, R. Saeedi, and M. Sarrafzadeh, "Power-aware computing in wearable sensor networks: An optimal feature selection," *IEEE Transactions on Mobile Computing*, vol. 14, no. 4, pp. 800–812, April 2015.

[76] S. Lauwereins, K. Badami, W. Meert, and M. Verhelst, "Optimal resource usage in ultra-low-power sensor interfaces through context- and resource-cost-aware machine learning," *Neurocomputing*, vol. 169, pp. 236 – 245, 2015, learning for Visual Semantic Understanding in Big Data ESANN 2014 Industrial Data Processing and Analysis. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S092523121500363X

Stellenbosch University https://scholar.sun.ac.za

[77] J. Korpela, K. Takase, T. Hirashima, T. Maekawa, J. Eberle, D. Chakraborty, and K. Aberer, "An energy-aware method for the joint recognition of activities and gestures using wearable sensors," in *Proceedings of the 2015 ACM International Symposium on Wearable Computers*, ser. ISWC '15. New York, NY, USA: ACM, 2015, pp. 101–108. [Online]. Available: http://doi.acm.org/10.1145/2802083.2808400

[78] J. Cui and B. Xu, "Cost-effective activity recognition on mobile devices," in *Proceedings of the 8th International Conference on Body Area Networks*, ser. BodyNets '13. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2013, pp. 90–96. [Online]. Available: http://dx.doi.org/10.4108/icst.bodynets.2013.253656

[79] R. Saeedi, B. Schimert, and H. Ghasemzadeh, "Cost-sensitive feature selection for on-body sensor localization," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, ser. UbiComp '14 Adjunct. New York, NY, USA: ACM, 2014, pp. 833–842. [Online]. Available: http://doi.acm.org/10.1145/2638728.2641313

[80] M. Padmanabhan, S. Murali, F. Rincón, and D. Atienza, "Energy-aware embedded classifier design for real-time emotion analysis," in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Aug 2015, pp. 2275–2278.

[81] C. Wang, S. J. Redmond, W. Lu, M. C. Stevens, S. R. Lord, and N. H. Lovell, "Selecting power-efficient signal features for a low-power fall detector," *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 11, pp. 2729–2736, Nov 2017.

[82] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*, 2nd ed. John Wiley & Sons, 2012.

[83] S. Rogers, *A first course in machine learning*, second edition. ed., ser. Chapman & Hall/CRC machine learning & pattern recognition series. CRC Press, 2017.

[84] K. P. Murphy, *Machine learning : a probabilistic perspective*, ser. Adaptive computation and machine learning. Cambridge, Mass.: MIT Press, 2012.

[85] S. B. Kotsiantis, "Supervised machine learning: A review of classification techniques." *Informatica*, vol. 31, no. 3, 2007.

[86] T. Fawcett, "An introduction to roc analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861 – 874, 2006, rOC Analysis in Pattern Recognition. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S016786550500303X

Stellenbosch University https://scholar.sun.ac.za

[87] S. P. le Roux, "A prototype animal borne behaviour monitoring system," Master's thesis, University of Stellenbosch, 2016.

[88] J. Doak, "Cse-92-18 - an evaluation of feature selection methods and their application to computer security," UC Davis: College of Engineering, p. 17, 1992. [Online]. Available: https://escholarship.org/uc/item/2jf918dh

[89] D. Krstajic, L. J. Buturovic, D. E. Leahy, and S. Thomas, "Cross-validation pitfalls when selecting and assessing regression and classification models," *Journal of Cheminformatics*, vol. 6, no. 1, p. 10, Mar 2014.

[90] S. P. le Roux, R. Wolhuter, T. Niesler, and N. Stevens, "Energy benefits of on-board behaviour classification for animal-borne sensor applications," in *2017 IEEE SENSORS*, 2017.

[91] K. Dudacek and V. Vavricka, "Experimental evaluation of the msp430 microcontroller power requirements," in *EUROCON 2007 - The International Conference on Computer as a Tool*, Sept 2007, pp. 400–404.

[92] *INA118 Precision, Low Power Instrumentation Amplifier*, Burr-Brown, Tucson, Arizona, US, Jan 2016. [Online]. Available: http://www.ti.com/lit/ds/symlink/ina118.pdf